Scalable Authenticated Communication in Drone Swarm Environment

Kyusuk Han[1*], *Eiman Al Nuaimi*[1], *Shamma Al Blooshi*[1], *Rafail Psiakis*[1], *Chan Yeob Yeun*[2]

[1] *Secure Systems Research Center, Technology Innovation Institute, UAE*
[2] *C2PS, EECS Dept. Khalifa University, UAE*
{kyusuk.han, eiman.alnuaimi, shamma.alblooshi, rafail.psiakis}@tii.ae, chan.yeun@ku.ac.ae

Abstract

The drone swarm is a preferable way to deploy many drones for large-scale missions. Establishing secure communication among drones in drone swarms is essential as the fog drone controls all other edge drones in the swarm. Although many researchers proposed methods of authenticating drones, most of them are unsuitable for use in swarm environments as they require either the ground station during the authentication or expensive PKI-based crypto operations with limited flexibility and scalability. In this work, we propose an efficient and scalable authentication protocol for drone swarm environments, enabling mutual authentication between fog and edge drones without involving the ground station. Moreover, we show that the protocol enables the verification of the sender in group communication. Protocol evaluations show security requirements satisfaction while achieving 14 - 20 times less computation overhead as compared to PKI-based models.

Keywords: UAV, Fog-Edge, Authentication, Drone swarm, Group communication

1 Introduction

Unmanned Aerial vehicles (UAV) and more specifically drones are emerging in our modern cyber-physical environment, being used in various domains such as public safety, surveillance, and monitoring of industrial, agricultural, infrastructural facilities, telecommunications, and others [1]. Missions of large-scale demand multiple drones operating together, forming groups. Controlling those groups of UAVs requires efficient handling, thus ‘swarming’ is considered, as the management of each drone by the ground station would be complicated with the number of drones in the same mission.

Through ‘swarming’, the ground station only needs to control the leader, which is called the ‘Fog drone’, letting all other drones be controlled by the fog drone, those are called the ‘Edge drone’. Since edge drones only communicate with other edge drones and the fog drone within the swarm, an edge drone could consist of a lighter-weight environment than the fog drone. For example, the capability of long-range communication such as cellular communication doesn’t necessarily need to be included in edge drones as they only communicate over Wireless Mesh Networks [2].

As security concerns rise in UAV environments, providing secure communication among UAVs in the drone swarm is one of the critical requirements. For establishing secure communication in a swarm, mutual authentication among drones is required, to avoid several threats including a man-in-the-middle attack. Also, as the fog drone is controlling all other edge drones, once it is compromised, the impact could be critical.

Several research works exist for the fog-IoT environment, however, many of them require the involvement of a ground station to perform the authentication process [3-6], which could be a problem in certain environments where connectivity to the ground station may not be guaranteed. Employing PKI-based approaches [7-10], brings substantial computation overhead, especially when a fog drone needs to authenticate hundreds of edge drones within swarms. Some techniques proposed lightweight authentication for resource-constrained drone platforms [11] using a reputation model for immediate authentication decisions. However, since the protocol uses shared secret values to generate keys, once the key is exfiltrated, there is a risk that any adversaries can generate the shared key between certain targeted entities.

The protocol in [12] provides efficient mutual authentication in fog-edge drone swarm environments, without the involvement of the ground station during the authentication process and aims for optimal efficiency by minimizing the use of public key-based cryptography. Moreover, it gives resiliency against key compromises.

In this paper, our main motivation is to extend [12], enabling authenticated group communication within the drone swarm once the initial authentication process is performed. While we still preserve the design fundamentals, the extended design includes group key distribution during the initial authentication process with minimum overheads. With the participant of the fog drone as an observer, edge drones verify the validity of the sender. The scalability and flexibility of the design and the resiliency against key compromises are still preserved.

Our contributions are finally summarized as follows:

- Our protocol provides mutual authentication in fog-edge drone swarm environments, efficiently preventing man-in-the-middle attacks and replay attacks on both sides establishing the secure channels between the fog drone and each edge drone.

[*] Corresponding Author: Kyusuk Han; E-mail: kyusuk.han@tii.ae

DOI: 10.53106/160792642024032502008

- It effectively manages the case against a compromise of a drone, providing forward and backward secrecy on the swarm management by limiting the impact only to the compromised drone or swarm.
- It doesn't require the involvement of the ground station to establish the authenticated communication between the fog drone and each edge drone.
- Our protocol enables efficient key management as keys are automatically revoked and invalid when the drone swarm mission is over.
- Our protocol supports group communication using the group key shared with the fog drone and edge drones, enabling the verification of the authenticity of the message sender within the drone swarm.
- In evaluating the protocol, 14-20 times better computational efficiency than the PKI-based design was observed for the initial authentication, as well as the minimizing of the storage size requirement for the certificate management in the group authentication.

The remainder of this paper is organized as follows. In the beginning, we discuss the drone swarm characteristics in Section 2. After that, we present the novel ideas of efficient drone-to-drone authentication in the swarm environment in Section 3. Next, we present the security analysis of the proposed protocol in Section 4 and the performance evaluation in Section 5. We briefly discuss the related work in Section 6. We conclude this paper in Section 7.

2 Characteristics of Drone Swarm Environment

In this section, we discuss the security issues in the Fog-Edge drone swarm environment.

2.1 Drone Swarm Environments

For a large-scale mission, many drones could be deployed into a certain region. In this case, controlling individual drones, in the same way, controlling a single drone could be complicated, grouping those multiple drones for the same mission into a drone 'swarm', and managing the swarm by setting a leading drone and delegating the management of all other drones in the swarm to the leading drone as Fog-IoT models is considerable.

Like the Fog-IoT model, let us call the leading drone *Fog Drone* and the other drones in the swarm *Edge Drones*. As depicted in Figure 1, the ground station has a direct channel only to the fog drone, while other edge drones are interconnected over the mesh network channels [2] within the swarm and communicate to the ground station only through the fog drone. Also, as depicted in Figure 2(a) fog drones could broadcast to edge drones and Figure 2(b) edge drones in the swarm could broadcast to other drones.

Note that the fog drone's roles are more for the management of the swarm by controlling the multiple individual edge drones, in contrast to edge drones, which perform only specific operations and mostly only communicate with the fog drone or other nearby edge drones.

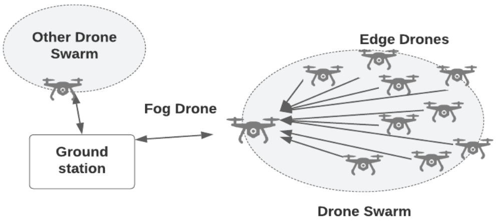
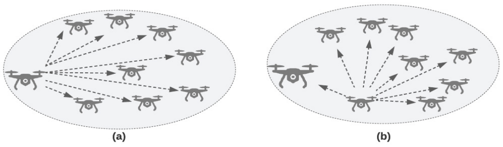


Figure 1. The ground station only communicates to Fog drone in the drone swarm while Edge drones only communicate to the Fog drone

2.2 Security Issues in Fog-Edge Drone Swarm Environments

When drones are sent to the mission field, they first establish communication in the operation phases [13]. Here, an attacker around the field could try to attack the drones, for example, he/she tries to impersonate either the fog drone or edge drones to intercept the information or compromise the swarm. To prevent such attacks, all drones establish secure communication channels.



(a) Fog drone broadcasts to edge drones

(b) Edge drone broadcasts to other drones (including Fog drone)

Figure 2. Group communication scenarios

In group communication scenarios as depicted in Figure 2, a malicious insider could send the message impersonating the valid sender, thus verifying the message sender is required.

To avoid the man-in-the-middle attack, the mutual authentication processes between fog drones and edge drones need to be done. Since the communication and computation overheads become a burden in fog drones than edge drones, achieving the efficiency of the authentication process is critical. As the attack can eavesdrop on the communication over the wireless channel, preventing a replay attack is also required.

Moreover, since the drone is flying out of the reach of the ground station, the attacker may physically capture and even disclose all information about the drone. In such a case, although the disclosure of the information in the current mission would be inevitable, however, attacker shall not know the previous secrets or future secrets from the exposure.

Some situations (e.g., desert areas) may not guarantee communication between the ground station and the fog drone. In such an environment, the authentication process should be able to be performed only between the fog drone and the edge drones.

Also, a drone swarm consists of hundreds of drones, and the changing of the drone groups is very frequent. As this is a huge burden to the key management, an efficient way of the key revocation is required.

Finally, as a fog drone could communicate with hundreds of edge drones, reducing the overhead in the authentication process for the fog drone is required.

2.3 Design Requirements

We define the following security requirements for the fog-edge drone swarm environment.

- **Man-in-the-middle-attack prevention:** Fog drone and edge drone shall authenticate each other.
- **Replay attack prevention:** Any previous data transmitted shall not be reusable.
- **Forward and Backward Secrecy:** The impact of any exfiltrated secrets from the compromised drone, and the impact on the swarm shall be limited to the current session.
- **Offline authentication:** Each drone should authenticate the other without the involvement of the ground station.
- **Efficient key revocation:** The protocol should provide an efficient way to revoke the key.
- **Authentication of the sender:** In broadcast communication, the sender of the message should be identifiable in the swarm.

Also, the overhead to the fog drone should be minimized. Note that the jamming of the communication is not in the scope of this work.

3 Proposed Protocol

In this section, extending [12], we propose scalable and efficient mutual authentication and key agreement among fog and edges. We also establish a group key for the drone swarm to enable secure group communication within the drone swarm.

3.1 Protocol Overview

We have the following entities for the drone swarm authentication scenarios.

- **Ground station** *GS*, manages one or multiple drone swarms for missions.
- **Fog drone** *fd*, communicates with *GS* on behalf of all drones in the swarm.
- **Edge drone** *ed*, only communicates to the fog drone and other edge drones in the swarm.
- a **swarm**, *S*, is a temporal group consisting of a *fd* and *ed*s for a mission. It is disbanded when the mission is over, either completed or aborted.

Our drone swarm authentication protocol consists of the following two phases.

- **Preparation Phase** (**PP**): In this stage, *GS* collects edge drones and a fog drone to build a swarm before the mission starts. During the preparation phase, we assume the communications between the ground station and the drones are protected. We describe the details in Section 3.2.
- **Initial Authentication Phase** (**AP**): In this stage, edge drones and the fog drone in the swarm are set to fly and mutually authenticate to establish secure channels. We describe the detail in Section 3.3. Note that *GS* does not involve in this phase.
- **Group Communication Phase (GP)**: In this stage, drones broadcast to other drones in the swarm. We describe the details in Section 3.4.

3.2 Preparation Phase

Let a ground station *GS* build an *i*-th swarm, S_i for a large-scale mission. *GS* first collects *n* number of edge drones $ed_{i,j}$, where $1 \leq j \leq n$ for S_i . *GS* also collects a fog drone fd_i, to manage S_i . Once drones are collected, *GS* performs the following:

PP.1 *GS* selects a random challenge C_i for S_i .

PP.2 *GS* then sends C_i and S_i to each edge drone $ed_{i,j}$, $1 \leq j \leq n$.

Once $ed_{i,j}$, for $1 \leq j \leq n$, receives C_i , it performs the following:

PP.3 Each edge drone $ed_{i,j}$ uses C_i as the input and generates the output $R_{i,j}$, where $R_{i,j} = F_{i,j}(C_i)$. $F_{i,j}(\mathrm{X})$ denotes the function that generates an output upon the input *X* as defined in [12].

PP.4 Each drone returns $R_{i,j}$ to *GS*, and stores S_i.

Then, *GS* performs the following:

PP.5 *GS* generates $dk_{i,j}$ per drone $ed_{i,j}$, where $dk_{i,j} = KDF(S_i \| fd_i \| ed_{i,j} \| R_{i,j})$. $KDF(X)$ denotes the key generation function with input *X*, and $X \| Y$ denotes the concatenation of *X* and *Y*.

PP.6 *GS* then generates the drone list DL_i , where $DL_i = \{fd_i, ed_{i,j}, dk_{i,j} | 1 \leq j \leq n\}$.

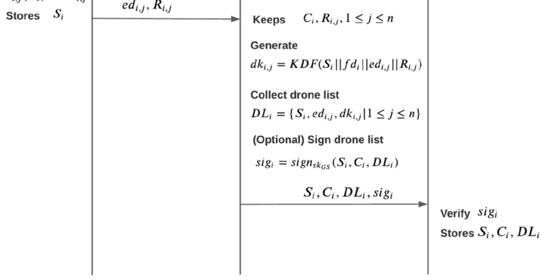
PP.7 ***GS*** deploys S_i , C_i and DL_i to the fog drone.

We assume the communication between *GS* and drones is done in a protected environment. The overall sequences are depicted in Figure 3.

3.2.1 Provisioning Public key

For the hybrid authentication case and group communication, fd_i generates the public key pair (pk_{fd_i}, sk_{fd_i}), where pk_{fd_i} denotes the public key and sk_{fd_i} denotes the private key. fd_i provides pk_{fd_i} to *GS*. Then *GS* deploys pk_{fd_i} , certified by *GS*, to each edge drone $ed_{i,j}$. In this scenario, we assume that edge drones are capable of public-key cryptography operations.

The hybrid authentication case is described as Case 2 in Section 3.3.



$(1 \leq j \leq n)$
Edge Drone
Ground Station
Fog Drone
$S_i, C_i, (pk_{fd_i})$
Select S_i, C_i
$F_{i,j}(C_i) \rightarrow R_{i,j}$
Stores S_i
$ed_{i,j}, R_{i,j}$
Keeps $C_i, R_{i,j}, 1 \leq j \leq n$
Generate
$dk_{i,j} = KDF(S_i||fd_i||ed_{i,j}||R_{i,j})$
Collect drone list
$DL_i = \{S_i, ed_{i,j}, dk_{i,j}|1 \leq j \leq n\}$
(Optional) Sign drone list
$sig_i = sign_{sk_{GS}}(S_i, C_i, DL_i)$
S_i, C_i, DL_i, sig_i
Verify sig_i
Stores S_i, C_i, DL_i

Figure 3. Swarm secret establishment in Preparation Phase

3.3 Initial Authentication Phase

When drones are deployed into the mission, they immediately start the establishment of the swarm S_i as the 'Ingress' stage [13]. In this stage, the fog drone and edge drones exchange the challenge and responses to authenticate each other through the temporally established mesh network mutually. We present two cases: *using only symmetric cryptographic operations* (Case 1) and *a hybrid approach that uses digital signatures together* (Case 2).

Improving [12], we include the group key distribution step in these processes. **AP.7-G** and **AP.11-G** are defined exclusively for the group key distribution, while all steps between **AP.7** to **AP.11** are revised as well.

3.3.1 Case 1: Using Only Symmetric Cryptographic Operation

Let a fog drone fd_i initiate the establishment of S_i. The fog drone fd_i performs the following:

AP.1 fd_i first randomly selects a nonce N_1.

AP.2 fd_i then broadcasts S_i, fd_i, and C_i with N_1 over the mesh network (line 2 in Figure 1).

Once the edge drone $ed_{i,j}$ in the field, for $1 \le j \le n$, receives S_i, C_i and N_1 from fd_i, it performs following:

AP.3 $ed_{i,j}$ generates a mission secret $rk_{i,j}$, where $rk_{i,j} = KDF(S_i \| fd_i \| ed_{i,j} \| R_{i,j})$.

$R_{i,j}$ is obtained by using $F_{i,j}$, $R_{i,j} = F_{i,j}(C_i)$ as same as **PP.3**.

AP.4 $ed_{i,j}$ randomly selects nonce N_2 and generates $auth^1_{i,j}$, where $auth^1_{i,j} = MAC(rk_{i,j}, N_1 \| N_2)$. $MAC(K, M)$ denotes the message authentication code or keyed hash function of the message M using the key K.

AP.5 $ed_{i,j}$ responds it's ID $ed_{i,j}$, N_2 and $auth^1_{i,j}$ to Fog Drone.

Whenever the fog drone fd_i receives the response from each $ed_{i,j}$, it performs following:

AP.6 fd_i finds $dk_{i,j}$ associated to $ed_{i,j}$ from DL_i, and generates $auth^{*}_{i,j}$, where $auth^{*}_{i,j} = MAC(dk_{i,j}, N_1 \| N_2)$, then compare $auth_{i,j}$ to $auth^{*}_{i,j}$. If both are equivalent, fd_i authenticates $ed_{i,j}$. Note that $sk_{i,j} \equiv rk_{i,j}$.

AP.7 fd_i then generates a session key $sek^{fd}_{i,j}$, $sek^{fd}_{i,j} = KDF(sk_{i,j}, N_2 \| N_1)$, where $KDF(X)$ denotes the key derivation function using the input X. Then fd_i derives the encryption key $sek^{enc}_{i,j}$ and the authentication key $sek^{mac}_{i,j}$, where $sek^{enc}_{i,j} = KDF(sek^{fd}_{i,j}, 0)$, and $sek^{mac}_{i,j} = KDF(sek^{fd}_{i,j}, 1)$ respectfully. Note that 0, 1 as input is used considering the Avalanche effect.

AP.7-G fd_i randomly selects the group key GK_i, and generates $egk_{i,j}$, where $egk_{i,j} = enc(sek^{enc}_{i,j}, GK_i)$, which denotes encryption of GK_i with $sek^{enc}_{i,j}$.

AP.8 fd_i generates a ACK and $auth^2_{i,j}$, where $auth^2_{i,j} = MAC(sek^{mac}_{i,j}, fd_i, ACK \| egk_{i,j} \| N_1 \| N_2)$.

AP.9 fd_i sends ACK, $egk_{i,j}$ and $auth^2_{i,j}$ to the edge drone $ed_{i,j}$.

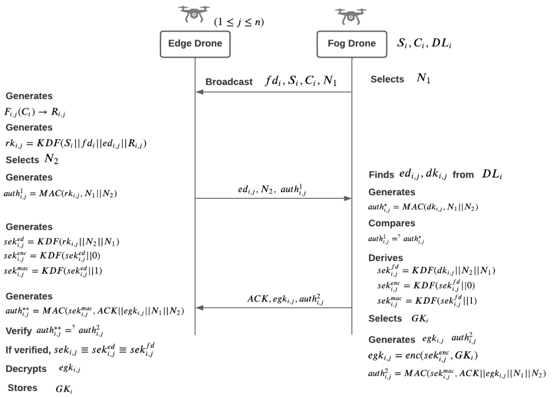
In receiving ACK, $egk_{i,j}$ and $auth^2_{i,j}$, each edge drone $ed_{i,j}$ performs **AP.10** and following:

AP.10 $ed_{i,j}$ generates $sek_{i,j}^{ed}$ using N_1 and N_2, where $sek^{ed}_{i,j} = KDF(rk_{i,j} \| N_2 \| N_1)$, then generates $sek^{enc}_{i,j}$ and $sek^{mac}_{i,j}$, where $sek^{enc}_{i,j} = KDF(sek^{ed}_{i,j}, 0)$, and $sek^{mac}_{i,j} = KDF(sek^{ed}_{i,j}, 1)$ respectfully.

AP.11 $ed_{i,j}$ generates $auth^{**}_{i,j}$ using $sek_{i,j}^{mac}$, where $auth^{**}_{i,j} = MAC(sek_{i,j}^{mac}, ACK \| egk_{i,j} \| N_1 \| N_2)$. If $auth^2_{i,j}$ and $auth^{**}_{i,j}$ are equivalent, $ed_{i,j}$ authenticates fd_i.

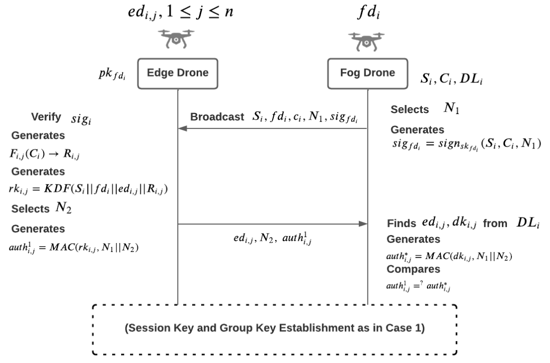
AP.11-G $ed_{i,j}$ decrypts $egk_{i,j}$ with $sek^{enc}_{i,j}$ and retrieves GK_i.

For all edge drone $ed_{i,j}$ in the swarm S_i, $1 \le j \le n$ completed **AP.11**, each individual edge drone and fog drone are mutually authenticated and established authenticated session key $sek_{i,j}$ between each $ed_{i,j}$ and fd_i, where $sek_{i,j} \equiv sek^{fd}_{i,j} \equiv sek^{ed}_{i,j}$. Thus, $sek^{enc}_{i,j}$ and $sek^{mac}_{i,j}$ are also shared.



$(1 \le j \le n)$
Edge Drone
Fog Drone
S_i, C_i, DL_i
Broadcast fd_i, S_i, C_i, N_1
Selects N_1
Generates
$F_{i,j}(C_i) \rightarrow R_{i,j}$
Generates
$rk_{i,j} = KDF(S_i || fd_i || ed_{i,j} || R_{i,j})$
Selects N_2
Generates
$auth^1_{i,j} = MAC(rk_{i,j}, N_1 || N_2)$
$ed_{i,j}, N_2, auth^1_{i,j}$
Finds $ed_{i,j}, dk_{i,j}$ from DL_i
Generates
$auth^*_{i,j} = MAC(dk_{i,j}, N_1 || N_2)$
Compares
$auth^1_{i,j} =^? auth^*_{i,j}$
Generates
$sek^{ed}_{i,j} = KDF(rk_{i,j} || N_2 || N_1)$
$sek^{enc}_{i,j} = KDF(sek^{ed}_{i,j} || 0)$
$sek^{mac}_{i,j} = KDF(sek^{ed}_{i,j} || 1)$
Derives
$sek^{fd}_{i,j} = KDF(dk_{i,j} || N_2 || N_1)$
$sek^{enc}_{i,j} = KDF(sek^{fd}_{i,j} || 0)$
$sek^{mac}_{i,j} = KDF(sek^{fd}_{i,j} || 1)$
Generates
$auth^{**}_{i,j} = MAC(sek^{mac}_{i,j}, ACK || egk_{i,j} || N_1 || N_2)$
$ACK, egk_{i,j}, auth^2_{i,j}$
Selects GK_i
Verify $auth^{**}_{i,j} =^? auth^2_{i,j}$
Generates $egk_{i,j}$ $auth^2_{i,j}$
If verified, $sek_{i,j} \equiv sek^{ed}_{i,j} \equiv sek^{fd}_{i,j}$
$egk_{i,j} = enc(sek^{enc}_{i,j}, GK_i)$
Decrypts $egk_{i,j}$
$auth^2_{i,j} = MAC(sek^{mac}_{i,j}, ACK || egk_{i,j} || N_1 || N_2)$
Stores GK_i

Figure 4. Authentication Phase (Case 1)
(Fog Drone and Edge Drones perform mutual authentication only with symmetric cryptographic operations.)



$ed_{i,j}, 1 \le j \le n$
fd_i
pk_{fd_i}
Edge Drone
Fog Drone
S_i, C_i, DL_i
Selects N_1
Verify sig_i
Broadcast $S_i, fd_i, c_i, N_1, sig_{fd_i}$
Generates
$sig_{fd_i} = sign_{sk_{fd_i}}(S_i, C_i, N_1)$
Generates
$F_{i,j}(C_i) \rightarrow R_{i,j}$
Generates
$rk_{i,j} = KDF(S_i || fd_i || ed_{i,j} || R_{i,j})$
Selects N_2
Finds $ed_{i,j}, dk_{i,j}$ from DL_i
Generates
Generates
$auth^1_{i,j} = MAC(rk_{i,j}, N_1 || N_2)$
$ed_{i,j}, N_2, auth^1_{i,j}$
$auth^*_{i,j} = MAC(dk_{i,j}, N_1 || N_2)$
Compares
$auth^1_{i,j} =^? auth^*_{i,j}$
(Session Key and Group Key Establishment as in Case 1)

Figure 5. Authentication Phase (Case 2)
(Fog Drone and Edge Drones perform mutual authentication with both asymmetric and symmetric cryptographic operations.)

3.3.2 Case 2: Hybrid Authentication Using Digital Signature

For the case that drones are capable of PKI, we show the design that only the fog drone uses the digital signature in broadcasting the challenge message to edge drones. While most steps are the same as Case 1, Case 2 has the following modifications.

At first, instead of **AP.2**, Fog drone fd_i performs following in sending the challenge:

AP.2-1 fd_i generates Signature sig_i of S_i, C_i and N_1 using the private key sk_{fd_i}, where $sig_i = sign(sk_{fd_i}, S_i \| C_i \| N_1)$. $sign(K, M)$ denotes signing a message M using the private key K.

AP.2-2 fd_i then broadcasts S_i, fd_i, C_i, N_1, with sig_i.

For the edge drone receiving the message from fd_i, it performs the following instead of AP.3:

AP.3-1 $ed_{i,j}$ verifies sig_i using fd_i 's public key pk_{fd_i}.

AP.3-2 For the valid sig_i , $ed_{i,j}$ generates $rk_{i,j}$ as **AP.3**.

Since fd_i is authenticated by $ed_{i,j}$ at these steps, by performing **AP.6**, each edge drone $ed_{i,j}$ and fd_i can be mutually authenticated. Therefore, the remaining steps, from **AP.7** to **AP.11**, are optionally performed to generate the shared session key $sek_{i,j}$ between the edge drone and the fog drone. Figure 4 depicts the overall step sequences of Case 2.

3.4 Group Communication

For group communication, each drone uses the group key GK_i distributed in the initial authentication process. Drones in the swarm have the group encryption key GK^{enc}_i and the authentication key GK^{mac}_i , where $GK^{enc}_i = KDF(GK_i, 0)$ and $GK^{mac}_i = KDF(GK_i, 1)$. Note that 0, 1 as input is used considering the Avalanche effect.

Let a drone $ed_{i,j}$ in the drone swarm i is broadcasting message M using GK^{enc}_i and GK^{mac}_i , as below.

GP.1 $ed_{i,j}$ generates the message authentication code $auth^m_{i,j}$, where $auth^m_{i,j} = MAC(sek^{mac}_{i,j}, M||ctr_k)$. ctr_k denotes k-th counter, which is set to 0 for the first time, and increments whenever the group communication happens.

GP.2 $ed_{i,j}$ generates the encryption $enc^g_{i,j}$, where $enc^g_{i,j} = enc(GK^{enc}_i , M||ctr_k||auth^m_{i,j})$. ctr_k denotes the counter of the message, where k is an incremental sequence, starting from 0 and $ctr_k = ctr_{k-1} + 1$.

GP.3 $ed_{i,j}$ generates a message authentication code $auth^g_{i,j}$ where $auth^g_{i,j} = mac(enc^g_{i,j}||ctr_k)$.

GP.4 $ed_{i,j}$ broadcasts $enc^g_{i,j}$ and $auth^g_{i,j}$ to drones in the swarm.

Let the recipients be the edge drones $ed_{*,j}$ and fd_i . In receiving $enc^g_{i,j}$ and $auth^g_{i,j}$, each drone performs the following.

GP.5 $ed_{*,j}$ and fd_i verify $auth^g_{i,j}$ using GK^{mac}_i .

GP.6 If $auth^g_{i,j}$ is valid, $ed_{*,j}$ and fd_i decrypt $enc^g_{i,j}$ using GK^{enc}_i and retrieves M, ctr_k and $auth^m_{i,j}$.

fd_i performs following while other edge drones $ed_{*,j}$ wait.

GP.7, fd_i verifies $auth^m_{i,j}$ by comparing to $auth^*$, where $auth^* = MAC(sek^{mac}_{i,j}, M||ctr_k)$.

GP.8 For valid $auth^m_{i,j}$, fd_i generates sig_{ack} , where $sig_{ack} = sign(sk_{fd_i}, h(ACK||M||ctr_k))$, and broadcasts ACK and sig_{ack} to drone swarm.

GP.9 $ed_{*,j}$ verifies sig_{ack} using pk_{fd_i} and ctr_k . When valid, all drones increase ctr_k to ctr_{k+1}.

If the result is invalid, fd_i broadcast the error message to the swarm. Ctr_K remains the same.

When the group message sender is fd_i , **GP.1** is not necessary. Instead, fd_i generates $enc^{g,f}_i$ and sig_i where $enc^{g,f}_{i,j} = enc(GK^{enc}_i , M \| ctr_k)$ in **GP.2** and $sig_i = sign(sk_{fd_i}, h(enc^{g,f}_i \| ctr_k))$ in **GP.3**. Since $ed_{*,j}$ verify the message in **GP.5,** steps from **GP.7** to **GP.9** are not required in this case.

3.5 Authentication in Scalable Scenarios

We show that our protocols are capable of being used for the following scalable scenarios.

3.5.1 Authentication with Multiple Fog Drones in a Swarm

When only a single fog drone is used to communicate with the ground station, the failure of the fog drone may result in the failure of the connection of the channel between the swarm and the ground station. Thus, deploying multiple fog drones in the swarm could be considered to increase the robustness against such a situation.

Let GS generate the swarm secrets for θ the number of fog drones in Swarm S_i . For fd^k_i , where $1 \leq k \leq \theta$, instead of **PP.5**, GS performs following:

PP.5-F For $1 \leq k \leq \theta$, GS generates $dk^k_{i,j}$, where $dk^k_{i,j} = KDF(S_i||fd^k_i||ed_{i,j}||R_{i,j})$.

i and j denote the swarm's ID and edge drone's ID respectively, as in Section. 3.2. Figure 6 depicts an example scenario that two fog drones fd_i^1 and fd_i^2 are leading the swarm S_i . In this case, GS generates two secrets $dk^1_{i,j}$ for fd_i^1 and $dk^1_{i,j}$ for fd_i^2, where $dk^1_{i,j} = KDF(S_i||fd^1_i||ed_{i,j}||R_{i,j})$, and $dk^2_{i,j}=KDF(S_i||fd^2_i||ed_{i,j}||R_{i,j})$. Note that the same $R_{i,j}$ is used to generate the difference.

GS also performs the following substituted steps instead of **PP.6** and **PP.7**.

PP.6-F GS generates the drone list DL^k_i , where $DL^k_i = \{fd^k_i, ed_{i,j}, dk_{i,j} | 1 \leq j \leq n, 1 \leq k \leq \theta\}$. θ denotes the number of fog drones in the same swarm.

PP.7-F GS deploys S_i , C_i and DL^k_i to the fog drone fd^k_i.

The authentication phase (**AP**) can be done without modification.

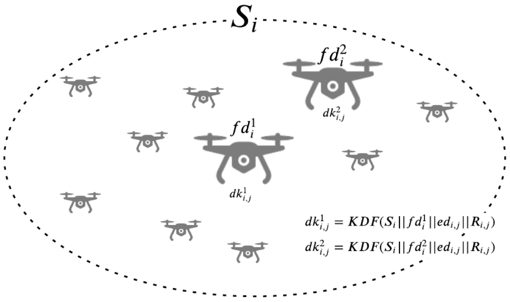

S_i
fd_i^2
$dk^2_{i,j}$
fd_i^1
$dk^1_{i,j}$
$dk^1_{i,j} = KDF(S_i||fd_i^1||ed_{i,j}||R_{i,j})$
$dk^2_{i,j} = KDF(S_i||fd_i^2||ed_{i,j}||R_{i,j})$

Figure 6. Multiple fog drones in the swarm

3.5.2 Authentication of Drones in Multiple Swarms

Upon the size of the mission and area, even deploying multiple swarms could be considered. In this scenario, several edge drones may need to join multiple swarms adaptively, as depicted in Figure 7.

Let GS set the edge drones to participate in multiple swarms. Instead of **PP.1** and **PP.2**, GS performs the following:

PP.1-S GS selects a random challenge C for S_i , where $1 \leq i \leq z$. z denotes the number of swarms. Note that the same challenge C can be used for multiple swarms.

PP.2-S GS then sends C and S_i to each drone ed_j , $1 \leq j \leq n$ and $1 \leq i \leq z$. n denotes the number of the edge drone, and z denotes the number of the swarm. Note that we omitted i in the challenge and edge drone's ID.

Each edge drone keeps assigned S_i , $1 \leq i \leq z$ and uses them during the authentication phase. Once the mission is over, the edge drone removes all S_i . Note that C is never stored in the edge drone.

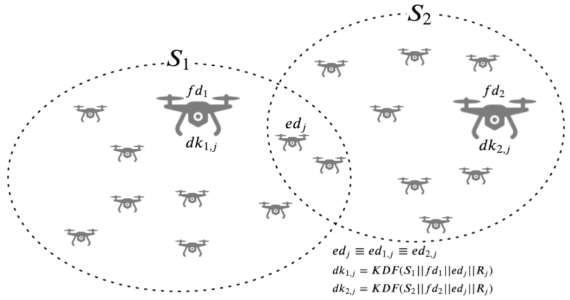

S_1
S_2
fd_1
$dk_{1,j}$
ed_j
fd_2
$dk_{2,j}$
$ed_j \equiv ed_{1,j} \equiv ed_{2,j}$
$dk_{1,j} = KDF(S_1||fd_1||ed_j||R_j)$
$dk_{2,j} = KDF(S_2||fd_2||ed_j||R_j)$

Figure 7. Edge Drones join multiple swarms

4 Security Analysis

In this section we show the security analysis of our proposed protocols.

4.1 Man-in-the-middle Attack Prevention

Let a third-party attacker *Adv* attempt to impersonate an edge drone $ed_{i,j}$ in the swarm S_i .

In **AP.5,** *Adv* generates a fake authentication message $auth^{Adv}{}_{i,j}$, to make the recipient accept $auth^{Adv}{}_{i,j} \equiv auth^1{}_{i,j}$, without the knowledge of $rk_{i,j}$. The probability that *Adv* generates a valid $auth^{Adv}{}_{i,j}$ without the knowledge of $rk_{i,j}$, is not higher than the probability that the *Adv* randomly choose one.

In **AP.9**, *Adv* attempts to impersonate the fog drone by generating $auth^{Adv}{}_{i,j}$ without knowledge of $sek_{i,j}$, where the recipient accepts $auth^{Adv}{}_{i,j} \equiv auth^2{}_{i,j}$. In this case, the probability that *Adv* generates a valid $auth^{Adv}{}_{i,j}$, is not higher than the probability that the *Adv* randomly chooses one.

In case of Case 2, in **AP.2-1,** the probability that *Adv* generates the fake signature sig^{Adv}, $sig^{Adv} \equiv sig_i$ without knowledge of the private key sk_{fd_i} is the same as the probability that *Adv* breaks the cryptographic primitives. Also *Adv* may impersonate an edge drone by compromising the message authentication code in **AP.9**, which is already shown above.

4.2 Replay Attack Prevention

In **AP.2,** *Adv* attempts to perform the replay attack by reusing the values captured during the previous authentication process.

For Case 1, *Adv* first captures the broadcast message fd_i , S_i , C_i , and N_1. Replaying this message without modification only results in the edge drone regenerating the $rk_{i,j}$, which *Adv* has no knowledge. *Adv* then also captures $ed_{i,j}$, N_2, and $auth^1{}_{i,j}$ from **AP.5** and replay with this information. However, without the knowledge of $rk_{i,j}$, *Adv* has no control of $auth^1{}_{i,j}$, thus the replay fails. The probability of success that *Adv* makes the fog drone fd_i believes $auth^1{}_{i,j}$ with the attacker's fake identity ed^{Adv} is not higher than the probability of success that *Adv* generates the collision of the cryptographic hash function. *Adv* also captures *ACK* and $auth^2{}_{i,j}$, and replay it. However, the probability of success that another edge drone accepts $auth^2{}_{i,j}$ that *Adv* replayed is not higher than the probability of success that *Adv* finds the collision of the cryptographic hash function. Thus, we show our protocol prevents the replay attack.

4.3 Forward Secrecy

Let *Adv* physically capture either an edge drone or a fog drone and completely disclose all secrets inside.

4.3.1 Attacker Captures Edge Drone

Adv uses a captured challenge C_i to generate the response $R_{i,j}$. Then, *Adv* may try to generate $rk_{i+1,j}$ using the swarm ID S_{i+1} and the fog drone's ID fd_{i+1} , however, those parameters are no longer valid in other swarm S_{i+1} , since the other fog drone, fd_{i+1} would not know the equivalent $dk_{i+1,j}$ in DL_{i+1} , *Adv*'s attempt to be authenticated as the same edge drone ID $ed_{i,j}$ fails. Instead, *Adv* may try to impersonate other ID $ed_{i+1,j}$, whose can be caught during **AP.5**. However, in this case using $R_{i,j}$ is meaningless, thus the probability of success that *Adv* is authenticated and establish the session key in another swarm S_{i+1} is not higher than the probability that *Adv* randomly selects a fake rk^{Adv} , where $rk^{Adv} \equiv rk_{i+1,j}$.

4.3.2 Attacker Captures Fog Drone

Let *Adv* capture a fog drone fd_1 of a Swarm S_1 and knows all secrets inside and *Adv* tries to impersonate edge drones in a new swarm S_{i+1}.

However, the probability that *Adv* generates a new valid $auth^2{}_{i+1,j}$ with old $dk_{i,j}$ is the same as the probability that *Adv* randomly selects dk^{Adv} , where $dk^{Adv} \equiv dk_{i+1,j}$.

In contrast, *Adv* may try to impersonate another fog drone $fd^2{}_i$ in the same swarm S_i , with the knowledge of $fd^1{}_i$. Adv needs to replicate $DL^2{}_i$ with $DL^1{}_i$. However, the probability that Adv finds $DL^2{}_i$, where $DL^1{}_i \equiv DL^2{}_i$ is not higher that the probability that *Adv* randomly chooses DL^{Adv}. Therefore, our protocol achieves forward secrecy.

4.4 Backward Secrecy

We also analyze that our protocol provides backward secrecy when either the edge drone or the fog drone is compromised.

For the compromised edge drone $ed_{i,j}$, *Adv* discloses the knowledge of the challenge and the associated response (C_i, $R_{i,j}$). Assuming *Adv* can also capture all previous public information of **AP.2**, and **AP.6**, *Adv* may know all previous secrets. However, knowing all information in the previous mission that $ed_{i,j}$ joined is not easy in practice, and all information is destroyed once the mission is over. For example, nonce N_1 and N_2 are only not stored, even in GS. Although perfect backward secrecy may not be achieved, partial protection is available in the real scenario.

Thus, although *Adv* breaks S_i by compromising the fog drone as a single point of failure, we show the impact is limited only to the current mission, and not influential to either past or future missions.

For compromised Fog Drone, since S_i expires after the mission, all $dk_{i,j}$ become invalid after the mission. Fog drone erases any expired information after the mission, thus any disclosed $dk_{i,j}$ could only reveal the information within the mission of S_i , not from S_{i-1} .

4.5 Offline-Authentication

Once drones are put in the mission and grouping the swarm, the ground station does not involve in the mutual authentication process, as depicted in both Figure 4, and

Figure 5. Also, the session key establishment and the group key establishment are performed without *GS*'s involvement.

4.6 Key Revocations

Since the secret $dk_{i,j}$ and $rk_{i,j}$ use S_i as the input, the secret becomes invalid once the mission is over. The fog drone simply renews the secret $dk_{i,j}$ with drone list DL_{i+1} when before the next mission S_{i+1} . The edge drone also generates $rk_{i+1,j}$ in joining the new mission S_{i+1} .

4.7 Authentication in Group Communication

For the third-party attacker who is not in the swarm, the attacker shall generate a fake encryption and message authentication code without knowing the valid $GK^{enc}{}_{i}$ and $GK^{mac}{}_{i}$, where the security is already analyzed in the above sections.

Let *Adv* be a compromised drone which becomes the malicious insider in the swarm. *Adv* knows the *k*-th counter ctr_k and the group key GK_i , and attempts to broadcast a fraudulent message M^{Adv} , impersonating the other genuine edge drones $ed_{i,j}$.

Adv generates a fake $auth^{m}{}_{adv}$, to be accepted as that $auth^{m}{}_{adv} \equiv auth^{m}{}_{i,j}$ without any knowledge of $sek^{mac}{}_{i,j}$. The *Adv* also generates $enc^{g}{}_{adv}$, where $enc^{g}{}_{adv} \equiv enc(GK^{enc}{}_{i}, \| ctr_k \| auth^{m}{}_{adv})$ in **GP.2 and** $mac^{g}{}_{adv} = mac(GK^{mac}{}_{i}, enc^{g}{}_{adv})$.

Although *Adv* could generate valid $enc^{g}{}_{adv}$ and use valid group keys, *Adv* still needs to generate $auth^{m}{}_{adv}$ which is to be $auth^{m}{}_{adv} \equiv auth^{m}{}_{i,j}$, without $sek^{mac}{}_{i,j}$ in **GP.1**. The probability of generating a valid $auth^{m}{}_{adv}$ is not higher than randomly selecting it.

In **GP.8,** *Adv* may attempt to broadcast the fake acknowledgment. *Adv* needs to generate the fake signature without knowledge of sk_{fd_i} , while fd_i can detect the behavior. Note that interruption of communication among drones in the same drone swarm is not our scope in this paper.

5 Performance Evaluation

We evaluated the performance of our protocol by first analyzing the computation overhead by design, then analyzing it with the implementation.

5.1 Computation Overhead by Design

We analyzed the number of operations to evaluate the computation overhead by design improving the computation overhead in [12].

5.1.1 Overhead in Authentication between Fog and Edge

The most overhead during the authentication occurs when the fog drone is receiving and handling the responses from multiple edge drones, and we evaluate the computation overhead during the following steps:

- Generation of shared key and response at edge drone **(AP.3-AP.4)**
- Verification of the response from an edge drone, at the fog drone (**AP.6**)
- Session key generation at the fog drone (**AP.7)**
- Session key generation at the edge drone **(AP.10)**
- Generation of the confirmation message at the fog drone (**AP.8)**
- Verification of the confirmation message at the edge drone **(AP.11)**

Additionally, the computation overhead of signature generation and verification are also evaluated as follows:

- Generation of signature of the fog drone (**AP.2-1)**
- Verification of signature of the fog drone (**AP.3-1**)
- *Signature Only* case

Table 1. Computation overhead in initial authentication

Case	Step (Fog/Edge)	Fog	Edge
Case 1	AP.6/AP.3 – AP.4	$n \times \boldsymbol{H}$	$2 \times \boldsymbol{H}$
	AP.7 /AP.10	$2n \times \boldsymbol{H}$	$2 \times \boldsymbol{H}$
	AP.8/AP.11	$n \times \boldsymbol{H}$	$1 \times \boldsymbol{H}$
Case 2	AP.2-1/AP.3-1	$1 \times \boldsymbol{S}$	$1 \times \boldsymbol{V}$
	AP.6/AP.3-2 – AP.4	$n \times \boldsymbol{H}$	$2 \times \boldsymbol{H}$
	AP.7/AP.10	$2n \times \boldsymbol{H}$	$(2 \times \boldsymbol{H})$
	AP.8/AP.11	$n \times \boldsymbol{H}$	$(1 \times \boldsymbol{H})$
Signature only	Challenge	$1 \times \boldsymbol{S}$	$1 \times \boldsymbol{V}$
	Response	$n \times \boldsymbol{V}$	$1 \times \boldsymbol{S}$

The result the total computation overheads of Case 1 are $4n{\cdot}H$ for the fog drone and $3{\cdot}H$ for an edge drone, while the ones of Case 2 are $1{\cdot}S + 4n{\cdot}H$ for the fog drone and $1{\cdot}V{+}1{\cdot}H$ for an edge drone when only the overhead of mutual authentication is measured. For Case 2, $3{\cdot}H$ for generating the authenticated session key is optional, which becomes mandatory for group communication. Since it is trivial that the computation overhead of *H* operation is much less than the overhead of *V* operation, both Case 1 and 2 show more huge efficiency than the signature-based design that requires $1S + nV$ for mutual authentication in the swarm. Symbols used in Table 1 denote the operations. *H* denotes the operation of the cryptographic hash functions including KDF and MAC. *S* and *V* denote the operation of signing the signature and the operation of verifying the signature respectfully. Note that asymmetric crypto operations only case is simplified from signature-based models such as [7].

5.1.2 Overhead in Authentication in Group Communication

Considering the group communication, the following additional operations are added for the group key distribution in the initial authentication process.

- Group key generation and encryption (**AP.7-G**)
- Decryption of Group key (**AP.11-G**)
- **AP.10** and **AP.11** to generate the decryption key

The computation overheads in fog and edge are estimated in Table 2. *E* denotes the encryption/decryption.

Table 2. Computation overhead in group key distribution

Case	Step	Fog	Edge
Group communication (in Case 1)	AP.7-G	$(1 \times \boldsymbol{H} + n \times \boldsymbol{E})$	-
	AP.10	-	$2 \times \boldsymbol{H}$
	AP.11	-	$1 \times \boldsymbol{H}$
	AP.11-G	-	$1 \times \boldsymbol{E}$

After the group key distribution is done, authenticated group communication processes are performed with the computation overhead as shown in Table 3.

Table 3. Computation overhead in group communication

Steps	Edge (Sender)	Fog	Edge (Recipient)
Key initialization	$2 \cdot H$	$2 \cdot H$	$2 \cdot H$
GP.1 – GP.3	$2 \cdot H + 1 \cdot E$	-	-
GP.5 – GP6	-	$1 \cdot H + 1 \cdot E$	$1 \cdot H + 1 \cdot E$
GP.7 – GP.8	-	$1 \cdot H + 1 \cdot S$	-
GP.9	-	-	$1 \cdot V$

Key Initialization is only performed at the first time, thus the overheads are $2 \cdot H + 1 \cdot E$ for the sender (edge drone), $2 \cdot H + 1 \cdot E + 1 \cdot S$ for the fog drone, and $1 \cdot H + 1 \cdot E + 1 \cdot V$ for the edge drone.

5.2 Simulation Setup

We then implement the simulation environment to evaluate the protocol in the computation times for the initial authentication.

We first designed the evaluation setup as Figure 6. A fog drone is on a Raspberry PI 4, with Ubuntu 20.04, and the edge drone swarm is on Intel i9 system. Both devices are connected over Wi-Fi.

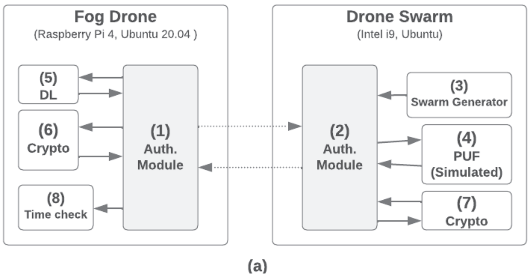


Figure 8. Implementation scenario comprising of a fog drone and edge drones

On each side, (1) and (2) in Figure 8, performs the overall protocol in the Fog drone and edge drone simultaneously. The swarm generator, (3), selects the number of edge drones. **PUF**, (4), provides the pre-computed response assuming the output generated by the function $F_{i,j}$ with the input C_i , since the scope of the evaluation is focused on the fog drone side. The drone list *DL* containing the swarm secrets for fog drones is managed separately as (5) and called when the mutual authentication is in progress. Cryptography functions (6) and (7) were implemented with the Cryptographic Services of the Python standard library and *PyCryptodome*. Once the computation from **AP.6** to **AP.8** are done, the computation time is checked in (8). Note that we assume the initial secrets *DL* are already established during the preparation phase, which is not the scope of this evaluation.

(1) and (2) perform three scenarios as discussed in Section 3.3. In addition to our two models, we implemented a mutual authentication protocol by exchanging the digital signature, which is a simplified model of existing protocols such as [7].

For the cryptographic functions, (6), we deployed HMAC-SHA512 for MAC and KDF, and RSA-4096 for the signing algorithm. We omitted the other asymmetric crypto algorithms such as ECDSA in this evaluation as the performance of the digital signature is only for comparison purposes.

5.3 Evaluation Results

Based on the setup in Section 5.2, we measured the time, (8) in Figure 8, to complete the authentication of all edge drones at the fog drone side. Table 1 shows the computation of three scenarios, and the graphs in Figure 9 and Figure 10 show the comparison among different scenarios with different numbers of edge drones in a swarm for initial authentication process. Note that the experiments only include the essential parts to establish the secure connection between the god drone and multiple edge drones.

For the case of one edge drone in Case 1, the total computation overhead of the symmetric-only scenario is approximately 0.67 milliseconds (*ms*) to complete **AP.6**, **AP.7** and **AP.8**, which is approximately 20 times faster than the signature-only scenario. Note that the latter case doesn't include the time to establish the session key.

The signature generation in fog drone (**AP.2-1**) shows approximately 646.64 *ms*, however, it is generated only once, and at the beginning of the protocol. The RSA key generations in both Case 2 (hybrid case) and asymmetric-only cases are approximately equivalent.

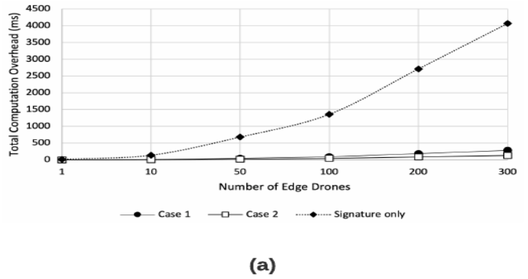


Figure 9. Comparison of computation time for mutual authentication only

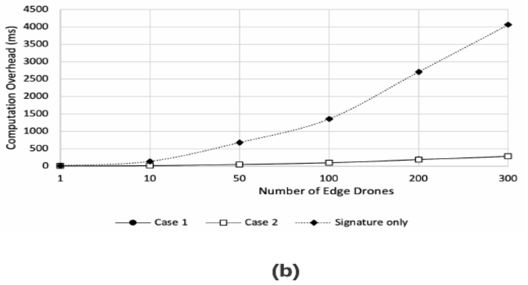


Figure 10. Comparison of computation time for mutual authentication with the session key agreement process

In contrast, the computation time for **AP.6** of Case 2 shows approximately 0.43 milliseconds which is almost equivalent to the symmetric key-only scenario. Moreover, at this point, the fog drone and edge drones are mutually authenticated. The remaining steps including **AP.7** and **AP.8** can be optionally performed to generate the session key and exchange the confirmation message, which is not the authentication purpose and only to confirm that the session key is generated, which shows a similar result as in Figure 10.

As the number of edge drones in a swarm increases, the difference in computation overhead between the proposed models (Case 1 and 2) and using asymmetric cryptography increases. As depicted in Figure 9 and Figure 10, the total overhead of our proposed model shows that it is approximately 14 times faster than the signature-only case.

Although the communication overhead is not explicitly measured, the fog drone needs the *challenge-response-confirmation* steps with the individual edge drone for the initial authentication processes and *the receiving message -sending proof* for each group communication. As the data rate is 150Mbps in Wireless Mesh Networks those interconnect as measured in [2], measuring the time for transmission overhead is reasonably considered marginal as each of essential data to be transmitted is approximately less than 500 bits, thus omitted in this paper. Moreover, we can still see the hybrid design has the least overhead by reducing the 1 step for fog drone authentication, which is required in case 1. We leave the evaluation in the actual environment as future work.

5.4 Comparison

Table 4 compares our proposed design with previous work for initial authentication.

Table 4. Comparison of protocols

Requirements	[7]	[10]	[5]	[1]	Proposed
Offline authentication		x	x	x	x
Mutual authentication		x	x	x	x
Replay attack prevention	x	x	x	x	x
Forward secrecy			x		x
Backward secrecy					Fog
Automated revocation					x
Scalability		x		x	x
Flexibility					x

We also compare the storage requirement to the *signature-only* model.

Using the signature of the edge drone requires the fog drone to store the public key of all edge drones in the drone swarm. Also, members of the drone swarm could be different per mission. Moreover, if edge drones are also communicating with the digital signature, edge drones also need to store multiple public keys of other edge drones. While the sizes of the certificate vary depending on the primitives, the size of an X.509 certificate with an ECDSA key using NIST P-256 is roughly 600-1100 bytes. If the number of drones in the swarm is big, this could be a huge burden to the fog drone and, mostly to the edge drone.

In contrast, our design only requires the fog drone, and the edge drone only stores one certificate when PKI computation is involved. Only the number of symmetric keys could increase, e.g., 128 bits or 256 bits for AES-128 or AES-256, which significantly reduces the burden of storage requirement.

6 Related Work

We explore the related work including the generic Fog-IoT environments and the Fog-Edge drone environments. Several protocols for drone environments such as [3-6] are not suitable for the fog-edge swarm model, as they require communication between the ground station and the individual drone. PKI-based models [14-15] for Fog-IoT and [1, 16] for drone environments could enable direct mutual authentication, however, they require a lot of performance overhead. The shared key-based models [16-17] could enhance efficiency, however, require more complicated key management including key provisioning and revocation. To deal with threats that try compromising shared keys, some researchers focused on hardware-assisted methods such as PUF-based techniques [17-19]. Some research works [1, 20] employed PUFs for the mutual authentication of drones. However, these protocols require the involvement of the authentication service, which owns Challenge-Response Pairs (CRPs), where the communication between the ground station and the drones may not be guaranteed. Our protocol addresses this limitation and enables lightweight mutual authentication between Fog and Edge drones, independent from the server, ensuring seamless authentication without communication with the server. Also, our protocol could leverage hardware-assisted methods [21] including PUF [19] for drone authentication by design as described [12].

7 Conclusion

In this paper, we extended the scalable mutual authentication protocol between fog drones and edge drones in establishing a drone swarm environment [12] to provide sender authentication in group communication. We showed that our designs effectively protect against attacks such as man-in-the-middle attacks and replay attacks. Our designs have resiliency against compromise by providing forward and backward secrecy and an efficient key revocation process, which is adequately suitable for fog-edge drone swarm environments. We also showed that our protocol efficiently performs the mutual authentication and key agreement with hundreds of edge drones, 14-20 times more efficiently than previous models, and can leverage various hardware-assisted security functions including PUFs, to increase the security strength of cryptographic assets used in real-world scenarios as in [12].

References

[1] P. Gope, B. Sikdar, An Efficient Privacy-Preserving Authenticated Key Agreement Scheme for Edge-Assisted Internet of Drones, *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 11, pp. 13621-13630, November, 2020.

[2] G. Raja, S. Anbalagan, A. Ganapathisubramaniyan, M. S. Selvakumar, A. K. Bashir, S. Mumtaz, Efficient and Secured Swarm Pattern Multi-UAV Communication, *IEEE Transactions on Vehicular Technology*, Vol. 70, No. 7, pp. 7050-7058, July, 2021.

[3] G. Cho, J. Cho, S. Hyun, H. Kim, SENTINEL: A Secure and Efficient Authentication Framework for Unmanned Aerial Vehicles, *Applied Sciences*, Vol. 10, No. 9, Article No. 3149, May, 2020.

[4] S. U. Jan, H. U. Khan, Identity and Aggregate Signature-Based Authentication Protocol for IoD Deployment Military Drone, *IEEE Access*, Vol. 9, pp. 130247-130263, 2021.

[5] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, H. Alhakami, LAKE-IoD: Lightweight Authenticated Key Exchange Protocol for the Internet of Drone Environment, *IEEE Access*, Vol. 8, pp. 155645-155659, 2020.

[6] M. Tanveer, A. U. Khan, N. Kumar, M. M. Hassan, RAMP-IoD: A Robust Authenticated Key Management Protocol for the Internet of Drones, *IEEE Internet of Things Journal*, Vol. 9, No. 2, pp. 1339-1353, January, 2022.

[7] M. A. Abdel-Malek, K. Akkaya, A. Bhuyan, A. S. Ibrahim, A Proxy Signature-Based Drone Authentication in 5G D2D Networks, *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Helsinki, Finland, 2021, pp. 1-7.

[8] M. A. Abdel-Malek, K. Akkaya, A. Bhuyan, A. S. Ibrahim, A Proxy Signature-Based Swarm Drone Authentication With Leader Selection in 5G Networks, *IEEE Access*, Vol. 10, pp. 57485-57498, 2022.

[9] Y. Aydin, G. K. Kurt, E. Ozdemir, H. Yanikomeroglu, Authentication and Handover Challenges and Methods for Drone Swarms, *IEEE Journal of Radio Frequency Identification*, Vol. 6, pp. 220-228, 2022.

[10] B. Semal, K. Markantonakis, R. N. Akram, A Certificateless Group Authenticated Key Agreement Protocol for Secure Communication in Untrusted UAV Networks, *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, London, UK, 2018, pp. 1-8

[11] T. D. Khanh, I. Komarov, L. D. Don, R. Iureva, S. Chuprov, TRA: Effective Authentication Mechanism for Swarms of Unmanned Aerial Vehicles, *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Canberra, ACT, Australia, 2020, pp. 1852-1858.

[12] K. Han, E. Al Nuaimi, S. Al Blooshi, R. Psiakis, C. Y. Yeun, A New Scalable Mutual Authentication in Fog-Edge Drone Swarm Environment, in: C. Su, D. Gritzalis, V. Piuri (Eds.), *ISPEC 2022: Information Security Practice and Experience*, Lecture Notes in Computer Science, vol. 13620, Springer, Cham, pp 179-196.

[13] T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis, M. Jones, Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs, *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 1255-1262

[14] A. B. Amor, M. Abid, A. Meddeb, A Privacy-Preserving Authentication Scheme in an Edge-Fog Environment, *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, Hammamet, Tunisia, 2017, pp. 1225-1231.

[15] M. S. Pardeshi, S.-M. Yuan, SMAP Fog/Edge: A Secure Mutual Authentication Protocol for Fog/Edge, *IEEE Access*, Vol. 7, pp. 101327-101335, 2019.

[16] M. H. Ibrahim, Octopus: an edge-fog mutual authentication scheme, *Journal of Network Security*, Vol. 18, No. 6, pp. 1089-1101, November, 2016.

[17] M. Barbareschi, A. De Benedictis, N. Mazzocca, A PUF-based hardware mutual authentication protocol, *Journal of Parallel and Distributed Computing*, Vol. 119, pp. 107-120, September, 2018.

[18] M. Barbareschi, A. De Benedictis, E. La Montagna, A. Mazzeo, N. Mazzocca, PUF-Enabled Authentication-as-a-Service in Fog-IoT Systems, *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Napoli, Italy, 2019, pp. 58-63

[19] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, R. J. Young, A PUF taxonomy, *Applied Physics Reviews*, Vol. 6, No. 1, Article No. 011303, March, 2019.

[20] V. Pal, B. S. Acharya, S. Shrivastav, S. Saha, A. Joglekar, B. Amrutur, PUF Based Secure Framework for Hardware and Software Security of Drones, *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, Kolkata, India, 2020, pp. 1-6.

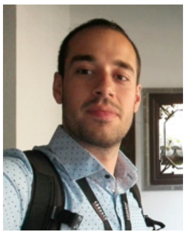
[21] Society of Automotive Engineers International, *Hardware Protected Security for Ground Vehicles*, SAE J3101, February, 2020.

Biographies

Kyusuk Han received M.S. (2004) and Ph.D. degree (2010) in Information and Communication Engineering, and Information Security from the KAIST. He is currently a Principal Researcher at Technology Innovation Institute, UAE. His current research focus is on the security of vehicles and UAVs.

Eiman Al Nuaimi is a cybersecurity expert with a B.S in Computer Engineering from Khalifa University. She is currently Associate Security Researcher at the Technology Innovation Institute.

Shamma Al Blooshi is a cybersecurity expert with a B.S. in Computer Engineering from Khalifa University and a M.S. in Cybersecurity from the University of Edinburgh (2023). She is currently Associate Security Researcher at Technology Innovation Institute, specializing in systems and network security, contributing to cutting-edge research and innovative solutions.



Rafail Psiakis holds Ph.D. degree from University of Rennes, France in 2018, and M.S. & Bachelor joint diploma at University of Patras, Greece in 2015. Currently he is a Lead Researcher at Technology Innovation Institute, UAE. His research interests include embedded systems, fault tolerance, and confidential computing.

Chan Yeob Yeun received the M.Sc. and Ph.D. in information security from the Royal Holloway, University of London, in 1996 and 2000, respectively. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Khalifa University, UAE.