S-MAPS: Scalable Mutual Authentication Protocol for Dynamic UAV Swarms

Gaurang Bansal, *Member, IEEE*, Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Unmanned Aerial Vehicles (UAVs) domain has seen rapid developments in recent years. UAVs have been deployed for many applications and missions like data transmission, cellular service provisioning, and computational offloading tasks etc. Yet, UAV deployment is still limited, partially owing to the security challenges it poses. UAVs are particularly vulnerable to physical capture, cloning attacks, eavesdropping, and man in the middle attacks. To address some of these security problems, this paper develops an authentication protocol for use in UAV swarms. To ensure physical security and rapid authentication, the proposed protocol uses Physical Unclonable Functions (PUFs). The protocol achieves high scalability compared to the state of the art by authenticating multiple devices at once. The proposed protocol supports dynamic topologies and multi-hop communication by using spanning tree-based traversal. It is also resistant to mobility, device failure, etc., and its improvements are achieved at significantly lower communication and communication cost as compared to state-of-the-art protocols.**

***Index Terms*—UAVs, physical security, authentication, dynamic topology, privacy, PUFs.**

I. INTRODUCTION

Unmanned Aerial Vehicle (UAVs) are small aerial devices that have become increasingly popular and have a wide range of applications. UAVs have been used for applications such as medical surveillance in natural disasters, traffic monitoring, military operations, delivery services, task offloading, etc. [1]. Although there has been rapid development of UAV based technologies and applications, their deployment has not achieved its full potential due to a number of challenges [2]. For example, bringing UAVs closer to the users makes it possible to provide better services to consumers. However, it also results in increased threats and vulnerabilities to security. Secondly, UAV communication's reliance on wireless channels makes UAVs prone to many active attacks such as replay attacks, man in the middle attacks, and node tampering attacks. These attacks can have drastic effects, which can lead to high levels of commercial and non-commercial losses. Attackers may also aim to exploit these UAV devices to access sensitive information, disrupt the normal operation, corrupt the data, or cause malicious interference [3].

One of the key security requirements for UAV deployments is developing authentication techniques through which a network entity (e.g., base station) or trusted source can ensure that only legitimate devices participate in the communication and the application [4]. In UAVs, it is essential to authenticate the devices frequently because of the dynamic nature of the environment. As UAVs move during their operation, their state (e.g., the state of the links, the base station serving them, etc.) is likely to change with time. Continuous authentication of the devices is necessary to ensure that a malicious adversary cannot access the resources and information related to the UAV application or affect its normal operation [5].

While various authentication protocols have been proposed in the literature for UAV environments (Section II presents a review of related work), they only consider scenarios with single UAV authentication. Many UAV applications involve UAV swarms where a group of UAVs participate in a mission together, for example, in monitoring, surveillance, and disaster management. In such scenarios, multi-hop inter-UAV communication is commonplace [6, 7]. Therefore, the authentication protocol must consider that the authenticating entity may have to rely on other UAVs for its communication with the UAV being authenticated. In this paper, we present an authentication protocol for communication between UAVs and base stations that has been designed for use in environments with UAV swarms. The proposed protocol is scalable for use with a large number of nodes.

The main contributions of this paper are as follows:

1) We propose a Physical Unclonable Function (PUF)-based authentication protocol capable of authenticating a UAV to the base station.
2) The protocol is highly scalable with running time in order of $\mathcal{O}(n)$, where n is the number of UAVs.
3) We use a run-time spanning tree algorithm to take any dynamic topology, UAV arrangement, and mobility into account.
4) Our protocol achieves confidentiality, protects against DoS attack, authentication, physical security, and ensures protection against replay, man in the middle attack (MITM), impersonation, and node-tampering attacks.

The organization of the rest of the paper is as follows. Section II discusses the previous related works in the area of UAV authentication protocols. Next, we present an overview of PUFs, a description of our system model, and an adversarial model in Sections II, III, and IV, respectively. The protocol description and the creation of a spanning tree are presented in Section V. Section VI discusses the security analysis, and Section VII discusses the computational cost. We provide a comparison of the above features in Section VIII and finally conclude in Section IX.

Gaurang Bansal and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, Singapore (e-mail: e0622339@u.nus.edu, bsikdar@nus.edu.sg)

II. Related Works

UAVs have quite different characteristic properties in comparison to other distributed network systems such as Mobile Ad-Hoc Networks (MANETs), Vehicular Ad-Hoc Networks (VANETs), and Wireless Sensor Networks (WSN) in terms of topology, mobility, service provided to the consumer, degree of availability, complexity, etc. The traditional security provisioning applicable to distributed networks fails to give similar results for UAVs [8, 9]. As highlighted in the Introduction, many security challenges have hindered the large scale deployment of UAVs [10, 11].

In recent years there has been considerable research work in developing lightweight security provisioning for UAVs [12–14]. In the authentication area, Hooper presented a framework [15] for attack resistance, which was improved by Blazy et al. in [16]. However, their protocol did not take into account physical security. In [17], the authors proposed the use of a secure channel to provide continuous authentication by using an array of random numbers. During the execution of the protocol, this array is used as a challenge by the BS to authenticate the UAVs. An authentication technique that uses a RFID-based architecture is presented in [18]. This work provides privacy and device uniqueness using cryptographic identities. However, the major drawback of the work is that there is no mutual authentication.

The work in [19] proposed the first distributed key authentication mechanism using a Certification Authority (CA). The major contribution of the work involved multi-party key management in a wireless mesh network. Each of the participating entities is provided with a unique identifier or a serial number that is used to generate public and private key by applying cryptographic functions. After each authentication round, the CA periodically updates the unique identifier and generates new private and public keys for authentication. The major drawback of this approach is its reliance on centralized trusted entities and high key computation costs. The authors of [3, 20] presented authentication protocols based on bilinear pairing and elliptical curve cryptography (ECC). Although they increased the security levels, their techniques are far from being scalable. In [21], the authors considered the problem of authentication in edge-assisted UAV scenarios. The proposed system considers third-party communication and allows mobile edge computing service providers to authenticate the UAVs. All the protocols above considered authentication as objective but failed to provide a scalable solution for dynamic UAV networks.

A protocol for anonymous mutual authentication has been proposed in [22]. This was the first work on combined authentication and anonymity for UAV networks. However, this solution is based on Trusted Platform Modules (TPMs), which are specialized and expensive security co-processors that need to be integrated into the system, leading to higher costs. Moreover, the authors failed to discuss the resistance against UAV node tampering and physical attacks that can result in an adversary extracting information and launching attacks quickly.

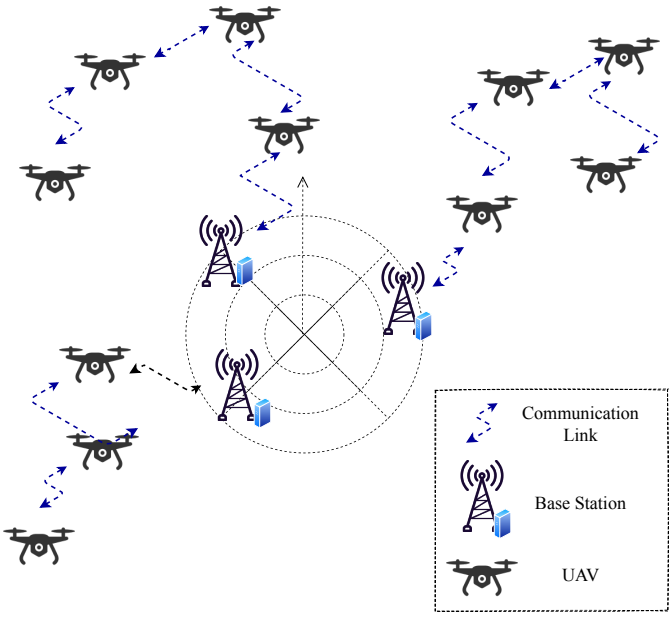


Fig. 1: System model.

Among scalable protocols, SEDA [23] proposed by Asokan et al. developed the first spanning tree-based protocol for distributed networks. However, subsequent research found its lack of resilience with the improvement suggested by Ibrahim et al. [24] in their DARPA protocol. Kohnhäuser et al. [25] introduced a new technique called cluster election mechanism to support more dynamic networks in recent work. Later, Ibrahim et al. [24], and Ambrosin et al. [26] solved the issue of mobility but used the idea of self-verification and reaching consensus. While their models are scalable, they do not guarantee a 100% coverage of the devices for authentication. Works like [27, 28] introduced physical security using PUF. Their models dealt with problems related to one-to-one authentication but failed to provide solutions for dynamic and large scale networks. To resolve all these issues, we present our proposed model and protocol in the following sections.

III. System Model

Figure 1 describes the system model used in this paper. There exist two types of entities in our system: base stations (BSs) and Unmanned Aerial Vehicle (UAVs). The proposed protocol applies to scenarios with multiple UAVs and multiple base stations. However, for easy understanding, the protocol is described in terms of one base station authenticating multiple UAVs simultaneously. The base stations are stationary and assumed to be trusted. UAV devices are deployed for operations and are vulnerable to physical and other security threats.

This paper does not make any assumptions on the type of UAV. The protocol is applicable even in scenarios where UAVs are heterogeneous and have different storage, memory, processor, etc. The UAV's are free to move, and it is assumed that the network remains connected during

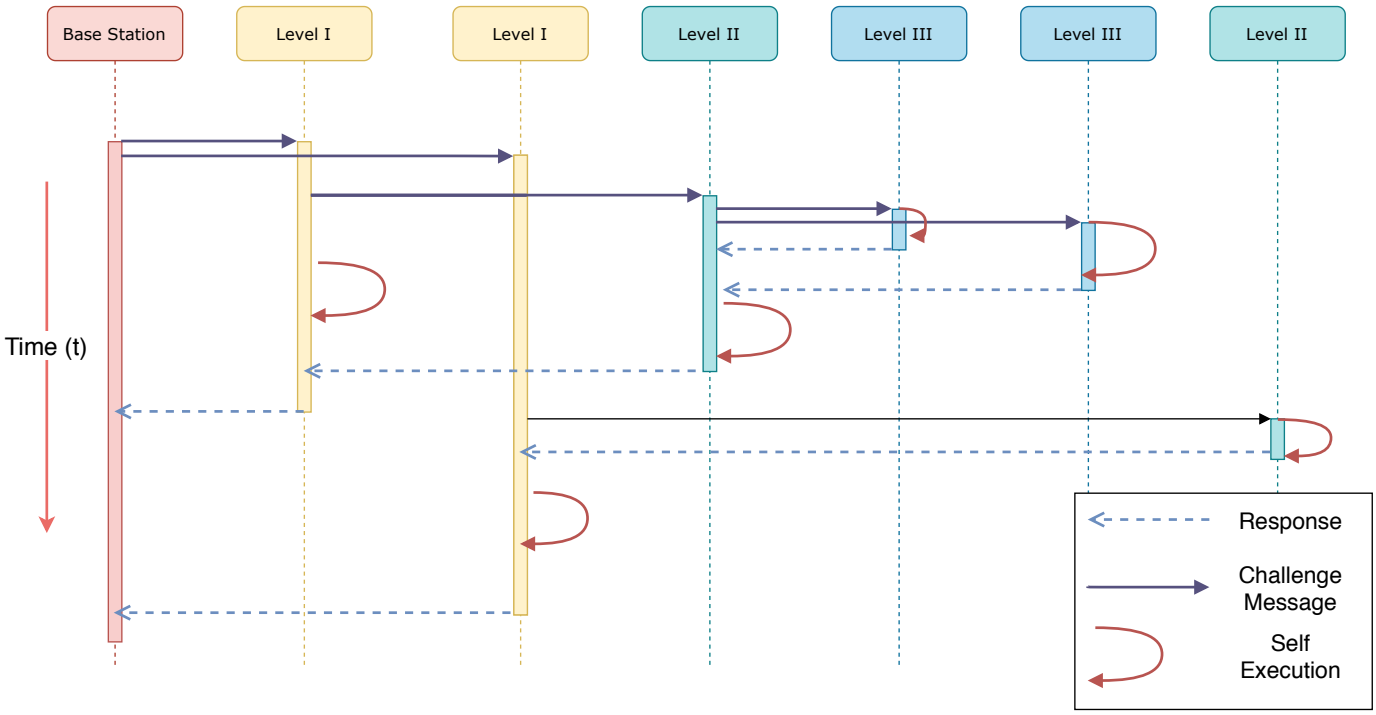


Fig. 2: Execution cycle of the proposed protocol. The levels in this figure denote the number of hops between the device and the base station. Level I consists of nodes that are a direct child of the base station in the spanning tree. Level II consists of children of level I nodes. The base station sends the message to the level I nodes, level I sends the message to Level II, and so on. Parent nodes wait for a reply from all its children and finally forward the reply to the base station.

the operation of the protocol. Each UAV is equipped with a Physical Unclonable Function. A description of PUFs is given in the next section. Base stations periodically verify the authenticity of the UAVs using the proposed protocol.

The proposed protocol can detect device tampering attacks, e.g., when the attacker changes/interrupts the device's normal functioning. In one iteration of the protocol, the proposed mechanism is resistant to only a single active attacker who disables future connections or tampers with the PUF. However, as the iterations of protocol increases, security against multiple active attackers can be achieved.

A. Attack Model

We assume an adversary who is granted complete control over the entire network (Dolev-Yao model) [29]. An attacker can hear all the unencrypted communication between the receiver and the sender. An adversary has the ability to masquerade as a legitimate UAV or tamper with the ongoing message exchanges. The tampering of messages is not limited to man-in-the-middle attacks. An attacker may try to eavesdrop on the transmitted messages, modify these messages, or replay them in the network. The attacker can also capture any UAV, disrupt any communication, and use brute force computation to decrypt any secret information.

B. Design Goals

This section highlights the primary security design goals for the proposed mutual authentication protocol.

i. An UAV and the base station should be able to authenticate each other successfully mutual. Also, the base station must be able to identify if the conversation is happening with an uncompromised legitimate UAV or not.
ii. If the communicated messages are tampered with, the receiver (either base station or UAV) must be able to detect the tampering and abort the authentication process. In other words, the integrity of messages must not be compromised.
iii. The protocol must be secure against security threats like replay attacks, masquerade attacks, and man-in-the-middle attacks.
iv. A unique session key must be generated for each authentication session. There must not exist any other way to generate this session key. Moreover, there should not be any correlation among session keys generated for different sessions.
v. As the attacker can physically capture or damage the UAV, the protocol must be safe against cloning attacks as well as physical attacks.

C. Assumptions

The assumptions made in this paper are as follows:

i. The communication between a device and its PUF is secure and tamper-proof [30].
ii. Attackers can physically capture or damage the UAV. An attacker can disable the communication of a captured UAV with other devices. If a UAV is captured, any

attempt to tamper with the PUF will render the PUF unusable.

iii. An iteration of the protocol refers to the actual data communication session between UAVs and the base station and UAV to UAV until the whole message is communicated.

iv. The base station is considered as a trusted authority and has sufficient resources.

v. Every new UAV must be first registered with the base station before it can be successfully authenticated. Upon registration, its identity (ID) D_{ij} and an initial challenge-response pair (C, R) generated from its PUF are saved in the base station's database.

D. Background of Physical Unclonable Functions

Physical unclonable functions can be considered as digital fingerprints of integral circuits. PUFs exploits the inherent randomness that is unique to a device and cannot be cloned or forged. This intrinsic randomness is generated during the fabrication of the chip. A PUF can be modeled as R = PUF(C), where the PUF uses its internal characteristics to map a challenge C to response R. A challenge C and its corresponding response R are called a challenge-response pair (CRP). CRPs are unique to a device, i.e., the same challenge gives a different response when applied on a different device.

IV. Proposed Protocol

A. Overview

In this section, we present the proposed protocol. The protocol consists of three phases:

1) **Registration Phase:** This step occurs before the system is deployed. The trusted BS initializes each device once and stores the challenge-response pair for each device in its directory.
2) **Message Communication Phase:** The devices receive the authentication request message from the base station in hop by hop manner. An UAV receives the authentication challenge from its parent and forwards the challenge to its descendants. Each UAV sends its response message to the base station through its parent, where the response also includes the aggregate of responses from its descendants.
3) **Authentication / Key Establishment:** In this phase, the BS identifies compromised entities and develops a secure session key for communication.

B. Registration Phase

Let n be number of devices connected at a given instance. Then we define the device set as:

$$D_S = \{D_1, D_2, D_3, \cdots, D_n\} \tag{1}$$

$$= \bigcup_{j=1}^{n} D_j, \tag{2}$$

$$n = ||D_S||. \tag{3}$$

When a new UAV needs to be deployed during the registration phase, the base station stores the new device's ID along with a challenge and response pair generated by the device's PUF. This CRP acts as the identification of the device. The set (C_S, R_S) represents the stored CRP in the memory of base station where C_i and R_i represent the CRP for i^{th} device:

$$C_S = \{C_1, C_2, C_3, \cdots, C_n\}, \tag{4}$$

$$= \bigcup_{j=1}^{n} C_j, \tag{5}$$

$$R_S = \{R_1, R_2, R_3, \cdots, R_n\}, \tag{6}$$

$$= \bigcup_{j=1}^{n} R_j. \tag{7}$$

C. Communication Phase

In our scenario, the base station periodically checks the authenticity of each device. The base station first identifies the devices in its vicinity. The base station and all UAVs near the base station form the vertices of a spatial graph. We consider that all the vertices are connected, forming a fully connected graph. Then base station invokes a spanning tree to identify the flow of authentication request messages. We consider the cost of connection or communication for each link to be the same. A spanning tree is commonly referred to as a subset of a graph containing all the vertices with a minimum number of edges. Since each vertex is covered only once, the spanning tree's inherent property is that it has no cycles.

Algorithm 1 Generating Spanning Tree

Input: **E:** List of edges "(u, v)", where 'u' and 'v' are UAV or BS

Output: j**:** list of edges in spanning tree

```
/* Initialisation */
j ← ϕ
/* Component[i]: Set of all vertices connected */
/* by a path to i */
while (| T |< (| N | −1)) do
    (u,v) = E.next()
    if (Component[u] != Component[v]) then
        j = j ∪ (u, v);
        Component[u] = Component[u] ∪ Component[v];
        Component[v] = Component[u] ∪ Component[v];
    end
end
```

The algorithm for tree construction is presented in Algorithm I. This algorithm takes the list of edges as input and returns a list containing the edges in the spanning tree. The output of the spanning tree protocol provides a list of routes containing the set of paths starting from the base station covering all the devices. The notations used in this paper are presented in Table 1. Let ρ be the number of routes in the spanning tree. A route is defined as a finite sequence

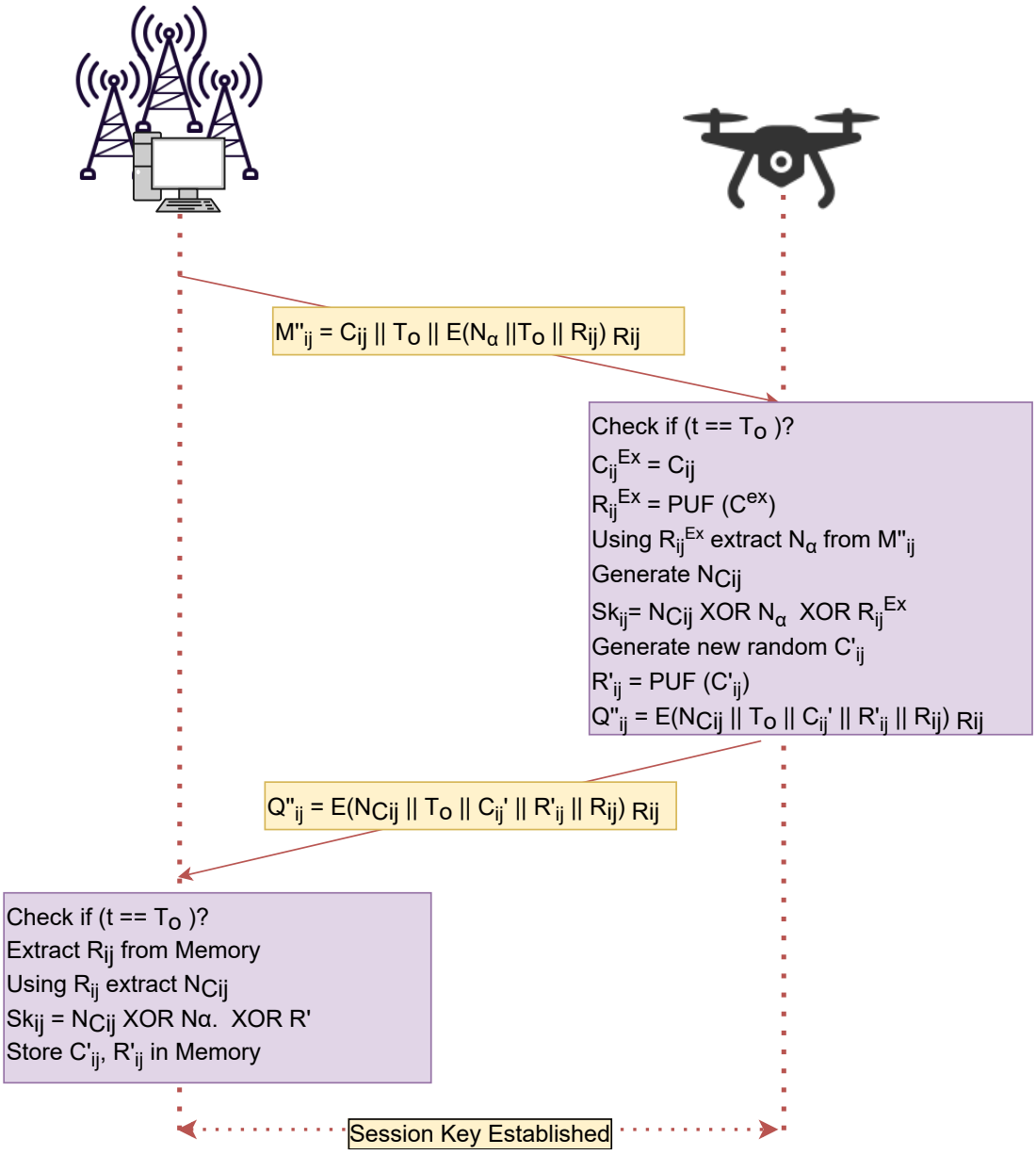


Fig. 3: Proposed Protocol

Notation	Meaning of Variables
UAV_{ij}	i^{th} UAV in j^{th} Path
(C_{ij},R_{ij})	Challenge Response Pair for UAV_{ij}
ρ	Total Number of Paths
P_j	j^{th} Path
k	Maximum number of UAVs in a path
T_o	Current Timestamp of BS
Encrypt(M)	Encryption of Message M
EM_{ij}	Encrypted Message for UAV_{ij}
$M"_{ij}$	Authentication Message for UAV_{ij}
$M"_j$	Aggregated Authentication Message along j^th path
R_{ij}^{Ex}	Extracted response from PUF
T^{Ex}	Extracted Current Time stamp of Device
N_α	Nonce Generated by BS
N_{Cij}	Nonce Generated by UAV_{ij}
Q_{ij}	Reply by UAV_{ij}
Sk_{ij}	Session Key generate between UAV_{ij} and BS

TABLE I: Table of notations.

of edges which joins a sequence of devices and terminates at node (with degree 1). The set of routes S is given by:

$$S = \{P_1, P_2, ..., P_\rho\}, \tag{8}$$

$$= \bigcup_{i=1}^{\rho} P_i, \tag{9}$$

$$\rho = ||S||. \tag{10}$$

The message flow chain of the protocol is designed using the edges in the spanning tree. The messages flow from parent to child during transmission, originating from the base station. On the return path, the children send their response to the parent, ultimately terminating at the base station.

Consider an instant $t = T_o$, where the base station initiates the authentication protocol. The base station generates

a pseudo random number N_α and uses the set of challenge-response pairs C and R for the entire iteration:

$$C = \begin{bmatrix} C_{11} & C_{21} & \cdots & \cdots & C_{k1} \\ C_{12} & C_{22} & \cdots & \cdots & C_{k2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ C_{1\rho} & C_{2\rho} & \cdots & \cdots & C_{k\rho} \end{bmatrix}$$

$$R = \begin{bmatrix} R_{11} & R_{21} & \cdots & \cdots & R_{k1} \\ R_{12} & R_{22} & \cdots & \cdots & R_{k2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ R_{1\rho} & R_{2\rho} & \cdots & \cdots & R_{k\rho} \end{bmatrix}$$

where (C_{ij} and R_{ij}) denote the challenge and response for i^{th} device in the j^{th} path, respectively.

Let Υ be a set which contains the total number of devices in path j. K is maximum value in the Υ. Then,

$$\Upsilon = \{k_1, k_2, \cdots, k_\rho\}, \tag{11}$$

$$= \bigcup_{i=1}^{\rho} k_i, \tag{12}$$

$$||\Upsilon|| = \rho, \tag{13}$$

$$k = \max(\Upsilon). \tag{14}$$

Recall that for the i^{th} device in the j^{th} path, where $j \in S$ and $i \leq k_j$, the CRP pair is given by (C_{ij}, R_{ij}). Here

$$C_{ij}, R_{ij} = 0 \iff (i > |k_j|) \vee (j > \rho). \tag{15}$$

Using the challenge response pair C and R, the base station generates the message M″, whose element M''_{ij} contains the message for i^{th} device in path j and is represented by:

$$M'' = \begin{bmatrix} & & & \\ C_{11}, T_o, 1, EM_{11} & \cdots & \cdots & C_{k1}, T_o, 1, EM_{k1} \\ C_{12}, T_o, 2, EM_{12} & \cdots & \cdots & C_{k2}, T_o, 2, EM_{k2} \\ \vdots & \ddots & \ddots & \vdots \\ C_{1\rho}, T_o, \rho, EM_{1\rho} & \cdots & \cdots & C_{k\rho}, T_o, \rho, EM_{k\rho} \end{bmatrix}.$$

The variable EM_{ij} denotes the encrypted message for i^{th} device in path j. EM_{ij} includes N_α, time stamp t, and secret key R_{ij} which is encrypted using R_{ij}:

$$EM_{ij} = \text{Encrypt}[N_\alpha || T_o || R_{ij}]_{R_{ij}} \tag{16}$$

The base station sends a message M''_j along the j^{th} path, where M''_j is the concatenated message of all M''_{ij} in the j^{th} path:

$$M''_j = M''_{1j} || M''_{2j} || \ldots || M''_{k_j j}. \tag{17}$$

For message transmission along the path, the parent removes itself from the list and sends it along with the child's message. The message also contains an expiry time. This expiry time can be used by the device to discard outdated messages. For example, it is possible that because of communication latency or loops in networks, the previously communicated messages are again re-transmitted to nodes. The expiry time is included in the message to avoid such a situation. Next, we describe the methodology for mutual authentication between the i^{th} device in path j and the base station.

1) When device D_{ij} (i^{th} device in path j) receives message M''_j from the base station, it Uses the path list in the message to find its position in the path and extract the message M''_{ij}:
$$M''_{ij} = C_{ij} || T_o || P_j || E[N_\alpha || T_o || R_{ij})_{R_{ij}}. \tag{18}$$
2) On extracting the message M''_{ij}, the device forwards the message to its children and starts the timer based on the number of its decedents. The timer value is calculated as: $2 \times RTT \times N$ where RTT is the round trip time with its immediate descendent and N is the number of remaining nodes in the path. The factor of 2 takes into account the computation time and other factors affecting the network latency. Device D_{ij} waits until the timer expires or all its descendants have communicated back before communicating its reply to its parent.
3) On receiving the BS's message, a device checks if the current timestamp is same as the expected time stamp T_o (timestamp refers to a time for an authentication period to occur). If not, the authentication request is rejected to avoid any replay attacks.
4) Using its PUF, device D_{ij} generates R_{ij} by giving C_{ij} as the input:
$$R_{ij} = \text{PUF}(C_{ij}). \tag{19}$$
Using R_{ij} as the key, it then decrypts the message to extract N_α. The second and third parameters in the message are included to ensure data integrity. The device then compares the value of the received timestamp with the expected T_o. To avoid confusion, we refer to the extracted parameters as ${T_o}^{Ex}$ and ${R_{ij}}^{Ex}$ (time extracted and response extracted). Thus, the device checks
$$N_{BS} = N_{(\alpha)} \iff ({R_{ij}}^{Ex} == R_{ij}) \text{ and } ({T_o}^{Ex} == T_o). \tag{20}$$
5) If the device fails to verify these security measures then the authentication process is terminated. Else, it generates a nonce N_{Cij} and also generates a new random challenge-response pair (C'_{ij}, R'_{ij}) using its PUF.
6) Having verified the base station, the device generates Sk_{ij} as
$$Sk_{ij} = R_{ij} \oplus N_\alpha \oplus N_{Cij}. \tag{21}$$
7) The device then sends creates its response T''_{ij} to the base station as:
$$T''_{ij} = E(N_{Cij} || T_o || C'_{ij} || R'_{ij} || R_{ij})_{R_{ij}}. \tag{22}$$
8) The response from the device is sent to the base station through its parent in the spanning tree. Before sending the response to parent, the device checks the path to see if it is a leaf node or not. In case it is a leaf node, it extracts its parent node using the spanning-tree and sends the response as its reply to its parent. The reply

from device D_{ij} is denoted by Q''_{ij} and for leaf nodes it is given by

$$Q''_{ij} = T''_{ij}. \tag{23}$$

In case the device is not a leaf node, it waits for responses from its descendants. Once it receives the replies from all its children, it aggregates the response by concatenating all the replies.

$$Q''_{ij} = T''_{ij}||Q''_{(i+1)j}||\cdots||Q''_{k_ij}. \tag{24}$$

If a descendent does not send a response message and timeout occurs, then the device assumes that its decedent is unavailable or compromised. The reply of such a descendent is set to 0.

9) On receiving the message from the devices in the path, the base station calculates the received N_{Cij}. The new challenge-response pairs (C'_{ij}, R'_{ij}) are stored in its memory. Then, it calculates the session key for device D_{ij} as:

$$Sk_{ij} = R_{ij} \oplus N_\alpha \oplus N_{Cij}. \tag{25}$$

10) With the establishment of a session key between D_{ij} and the BS, the mutual authentication between D_{ij} and the BS is complete.

V. Illustration of Execution Scenarios

In this section we present example scenarios for the execution of the protocol in order to highlight the features of the protocol and its operation under different conditions. Figure 6 shows the scenarios and the operation of the proposed mechanism.

1) **Case I:** No attacker (message communication from base station to devices).
   Base station creates the spanning tree $[(0 \rightarrow 1), (1 \rightarrow 2), (1 \rightarrow 3), (2 \rightarrow 4), (4 \rightarrow 5), (5 \rightarrow 6)]$ for the network as shown in Fig. 5 (State 1) and the authentication message (shown in yellow) is communicated from parent to child.
2) **Case II:** No attacker (message communication from device to base station).
   On receiving the authentication request message from base station, the UAVs decrypt the message and send the reply back (shown in red) to their parents starting from the leaf node $[(3 \rightarrow 1), (6 \rightarrow 5), (5 \rightarrow 4)]$ as shown in Fig. 5 (State 2).
3) **Case III:** Compromised UAV (Compromised UAV fails to participate in communication).
   In network state 3 of Fig. 5, we consider UAV 4 to be malicious or attacked by an adversary. As a result, UAV 4 does not participate in communication. Thus, its parent (UAV 2) will not receive any communication from UAV 4 before a timeout and understand that UAV 4 is compromised. This information is further communicated to the base station in the reply from UAV 2.
4) **Case IV:** Compromised UAV (Compromised UAV participates in communication).
   In state 4 of Fig. 5, UAV 4 becomes malicious or is attacked by an adversary. In this scenario, the UAV participates in the communication. Since any malicious entity cannot forge the PUF, the message is encrypted with a random string, rather than the correct response (shown in blue). The base station's expected response from UAV 4 will thus not match with what the malicious entity would have sent. Hence, the base station can detect that UAV 4 is compromised.

VI. Formal Security Analysis

This section provides a formal security analysis of our protocol, by modelling the communication in the protocol using Mao-Boyd logic [31]. The notations for symbols as used by Mao-Boyd logic are:

1) $D_{ij} \models$ BS: D_{ij} believes BS.
2) $D_{ij} \overset{K_{ij}}{\sim}$ M: D_{ij} encrypted M using the key K_{ij}.
3) $D_{ij} \overset{K_{ij}}{\triangleleft}$ M: D_{ij} extracts M using key K_{ij}.
4) $D_{ij} \overset{Sk_{ij}}{\leftrightarrow}$ BS: Sk_{ij} is a valid shared key.
5) #(N_α): Nonce N_α is unique and not used before.
6) sup(BS): BS is assumed to be secure and trustworthy.
7) $D_{ij} \triangleleft \| M$: D_{ij} cannot get the message M.

We show that D_{ij} knows that N_α is a valid shared and secure message between D_{ij} and BS. All other proofs can be derived in a similar way.

Proof. We assume that a PUF is secure and R_{ij} is known only to the base station and the corresponding device D_{ij}. Also, we assume that the base station is trusted and cannot be compromised. Using the communication sequence presented in Fig. 2, we now describe the proof for the proposed authentication mechanism.

In the Initialization phase, the CRP of each UAV U_j was stored in the BS. Hence, U_j knows that R_j is a shared secret between U_j and BS (i). In the communication phase, U_j is able to obtain N_U using R_j (ii). The Mao Boyd logic equivalents of these statements are shown below:

$$U_j \models U_j \overset{R_j}{\leftrightarrow} \text{BS}, \tag{i}$$

$$U_j \overset{R_j}{\triangleleft} N_U. \tag{ii}$$

Using the authentication rule, (the Mao Boyd rules are provided as part of Appendix), We can combine (i) and (ii) to get (iii) which states that the U_j knows BS encrypted N_U using the key R_j

$$U_j \models \text{BS} \overset{R_j}{\sim} N_U. \tag{iii}$$

BS is the super principal with respect to N_U. The nonce N_U generated by BS must be fresh and unused.

$$U_j \models sup(\text{BS}). \tag{iv}$$

$$U_j \models \#(N_U). \tag{v}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{BS\models D_{ij}\models D_{ij}\overset{R_{ij}}{\leftrightarrow}BS}\wedge BS\models D_{ij}\models\{BS\}^{c}\lhd\|N_{Cij}\wedge\dfrac{\dfrac{}{BS\models D_{ij}\overset{R_{ij}}{\leftrightarrow}BS}\wedge BS\overset{R_{ij}}{\lhd}N_{Cij}}{BS\models D_{ij}\overset{R_{ij}}{|\sim}N_{Cij}}}{BS\models D_{ij}\models\{D_{ij},BS\}^{c}\lhd\|N_{Cij}}\wedge BS\models sup(D_{ij})}{BS\models\{D_{ij},BS\}^{c}\lhd\|N_{Cij}}\wedge BS\models sup(D_{ij})\wedge BS\models\#(N_{Cij})}{BS\models D_{ij}\overset{N_{Cij}}{\leftrightarrow}BS}$$

(a) Proof for: "**BS** is convinced that N_{Cij} is a valid shared key between $\mathbf{D_{ij}}$ and **BS**".

$$\dfrac{\dfrac{\dfrac{}{BS\models D_{ij}\overset{R_{ij}}{\leftrightarrow}BS}\wedge BS\models D_{ij}^{c}\lhd\|N_{\alpha}\wedge BS\overset{R_{ij}}{|\sim}N_{\alpha}}{BS\models\{D_{ij},BS\}^{c}\lhd\|N_{\alpha}}\wedge BS\models\#(N_{\alpha})}{BS\models D_{ij}\overset{N_{\alpha}}{\leftrightarrow}BS}$$

(b) Proof of "**BS** knows N_{α} is a secure message key between D_{ij} and **BS**".

$$\dfrac{\dfrac{\dfrac{}{D_{ij}\models D_{ij}\overset{R_{ij}}{\leftrightarrow}BS}\wedge D_{ij}\models BS^{c}\lhd\|N_{Cij}\wedge D_{ij}\overset{R_{ij}}{|\sim}N_{Cij}}{D_{ij}\models\{D_{ij},BS\}^{c}\lhd\|N_{Cij}}\wedge D_{ij}\models\#(N_{Cij})}{D_{ij}\models D_{ij}\overset{N_{Cij}}{\leftrightarrow}BS}$$

(c) Proof of "$\mathbf{D_{ij}}$ knows $N_{C_{ij}}$ is a secure message between $\mathbf{D_{ij}}$ and **BS**".

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{D_{ij}\models BS\models D_{ij}\overset{R_{ij}}{\leftrightarrow}BS}\wedge D_{ij}\models BS\models\{D_{ij}\}^{c}\lhd\|N_{\alpha}\wedge\dfrac{\dfrac{}{D_{ij}\models D_{ij}\overset{R_{ij}}{\leftrightarrow}BS}\wedge D_{ij}\overset{R_{ij}}{\lhd}N_{\alpha}}{D_{ij}\models BS\overset{R_{ij}}{|\sim}N_{\alpha}}}{D_{ij}\models BS\models\{D_{ij},BS\}^{c}\lhd\|N_{\alpha}}\wedge D_{ij}\models sup(BS)}{D_{ij}\models\{D_{ij},BS\}^{c}\lhd\|N_{\alpha}}\wedge D_{ij}\models sup(BS)\wedge D_{ij}\models\#(N_{\alpha})}{D_{ij}\models D_{ij}\overset{N_{\alpha}}{\leftrightarrow}BS}$$

(d) Proof of "$\mathbf{D_{ij}}$ knows N_{α} is a secure message between $\mathbf{D_{ij}}$ and **BS**".

U_j is aware that BS believes that R_j is a valid shared key between U_j and BS (CRP exchanged during initialization phase), so it can be written as:

$$U_j\models BS\models U_j\overset{R_j}{\leftrightarrow}BS. \tag{vi}$$

As per the protocol description, U_j generates nonce N_U which is only communicated in message Re_j in encrypted format. Hence, U_j is aware that BS is aware that no one other than U_j knows N_U.

$$U_j\models BS\models\{U_j\}^{c}\lhd\|N_U. \tag{vii}$$

Applying the confidentiality rule using (iii), (vi) and (vii) U_j is convinced that no one else except itself and base station knows the secret nonce N_U.

$$U_j\models BS\models\{U_j,BS\}^{c}\lhd\|N_U. \tag{viii}$$

Applying the super principle rule, we can reduce (viii) to.

$$U_j\models\{U_j,BS\}^{c}\lhd\|N_U. \tag{ix}$$

Finally, applying the good-key rule to (iv), and (ix) we have,

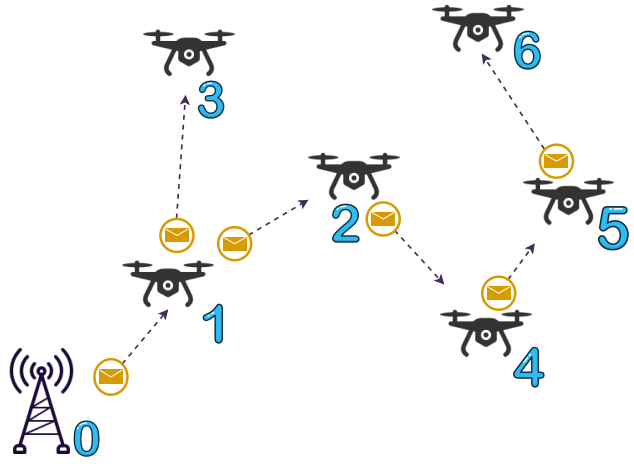
$$U_j\models U_j\overset{N_U}{\leftrightarrow}BS. \tag{x}$$

Hence, it is proved that U_j is convinced of the shared secret N_U between U_j and BS. Hence, a secure session key Sk can be generated for its communication with the BS. The formal proof using Mao-Boyd logic is presented in Fig. 4. □

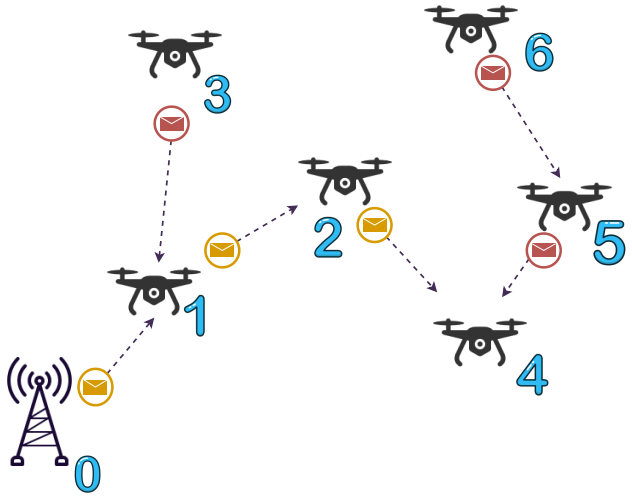
A. Security Goal Analysis

This section discusses the security features of the proposed protocol. Informally, the security properties of the proposed protocol are based on the following.

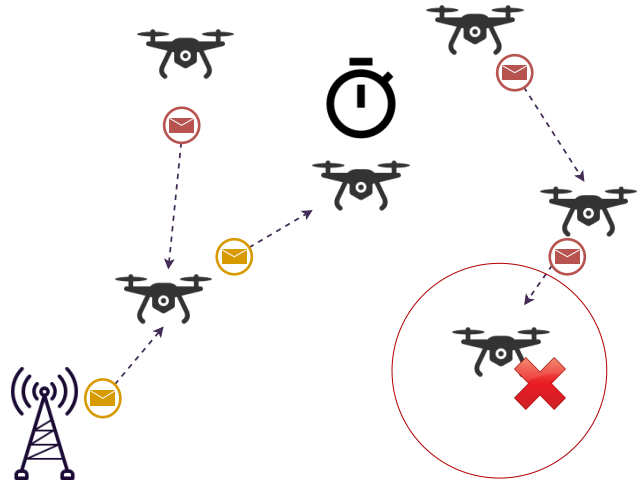
- All the messages sent by an UAV are encrypted with the response from its PUF. PUFs provide digital fingerprints to devices and ensure physical security. Any manipulation or tampering of the device makes the PUF unusable, thereby making it resistant to device tampering attacks.
- The PUF response helps in authentication of the UAVs, and encryption makes it resistant to eavesdropping. A PUF is unique to the device and cannot be cloned. Thus, the PUF registered during the initialization period facilitates secure authentication of the device.
- The proposed protocol uses multiple iterations of the protocol and allows for the UAV network's topology to be dynamic. An UAV may change its connection before the start of the protocol. In such a case, the spanning tree is altered accordingly.
- Since the spanning tree is generated with a random initial point, the possibility of a compromised UAV



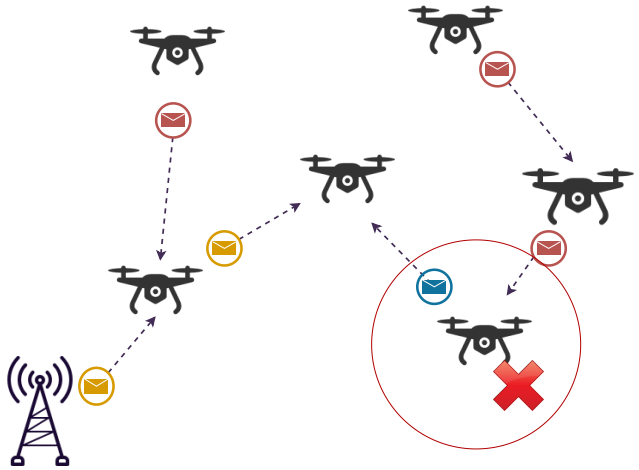
(a) **Network State 1:** Communication from base station to UAVs in hop to hop manner $[(0 \rightarrow 1), (1 \rightarrow 2), (1 \rightarrow 3), (2 \rightarrow 4), (4 \rightarrow 5), (5 \rightarrow 6)]$.



(b) **Network State 2:** Replies (shown in red) from children are returned to parents starting from leaf nodes $[(3 \rightarrow 1), (6 \rightarrow 5), (5 \rightarrow 4)]$.



(c) **Network State 3:** UAV 4 is compromised. UAV 4 does not participate in protocol.



(d) **Network State 4:** Compromised UAV 4 sends a message encrypted by a random string instead of actual PUF response.

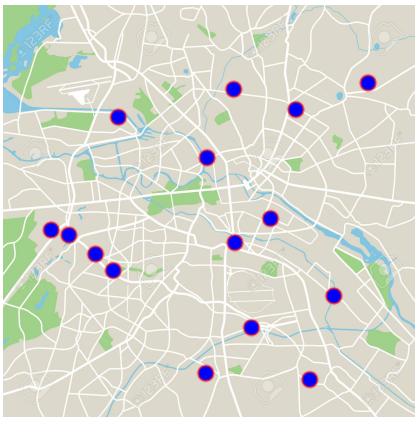
Fig. 5: Different scenarios of execution of proposed protocol

blocking the entire communication with other UAVs is very low. As a result, the protocol is resistant to static DoS attacks.

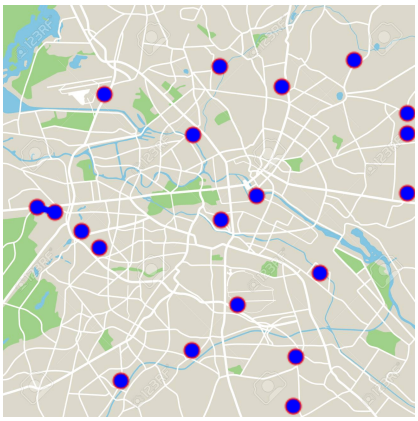
- We use timestamps in the communication to avoid replay attacks. Since the timestamp is secured in the message, an attacker cannot change the timestamp. Thus, in case the attacker sends the same message at a later point in time, it can be identified by the UAVs and discarded by comparing the current timestamp with the timestamp sent by the attacker.
- An attacker may use a proxy UAV device to generate a correct checksum by generating messages from the compromised UAV relay to the proxy device and vice versa. However, our model is resistant to such an attack. The proxy device would need the same hardware to generate the corresponding challenge-response pair, which is not possible. Also, it would mean that there is a need for additional time if the proxy server used brute force, resulting in a timeout.

With this overview of the security features of proposed protocol and how they are achieved, we now present a comparison of these features to the current start of the art. The features of interest are: (i) authentication, (ii) mobility, (iii) dynamic topology, (iv) parallel execution, (v) Resistance to eavesdropping, (vi) DoS attack resistance, (vii) proxy attack resistance, (viii) man in the middle attack resistance, (ix) replay attack resistance, and (x) physical security.

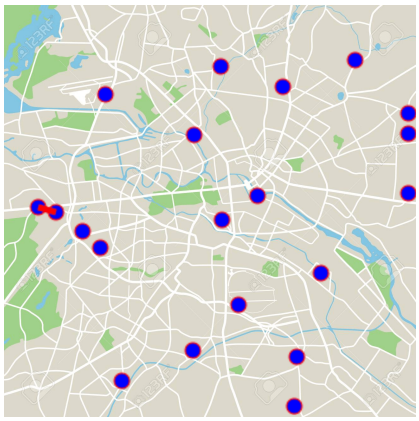
We compare our protocol with [20] ,[32], [33], [34] and [35]. The comparison of security features is presented in Table I. All the protocols provide either formal or informal analysis to show the security of protocol. However, only [35] has considered physical threats and provides protection



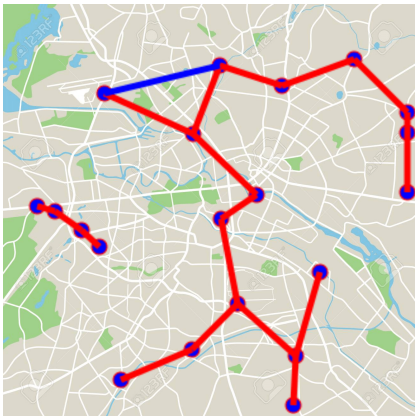
(a) Iteration 0: Initial deployment of UAVs.



(b) Iteration 1: Unconfirmed edge (shown in blue).



(c) Iteration 2: Edge confirmed (shown in red). Now edge is part of spanning tree.



(d) Iteration 63: Edge unconfirmed (blue line).

(e) Iteration 64: Previous blue edge (Fig. (d)) unaccepted, new edge unconfirmed (blue line).

(f) Iteration 65: Edge confirmed. All nodes covered. Spanning tree creation complete.

Fig. 6: Different iterations of the execution of the spanning tree algorithm. Blue dots are UAVs that are deployed in a region. Red line show the links that are finalised and are part of spanning tree. Blue lines represent probable spanning tree edges which can be rejected or accepted to spanning tree based on Algorithm 1.

TABLE II: Comparison of Security Features

	Security Features									
Protocols	**SF1**	**SF2**	**SF3**	**SF4**	**SF5**	**SF6**	**SF7**	**SF8**	**SF9**	**SF10**
[20]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
[32]	✓	✗	✗	✗	✓	✓	✗	✓	✓	✗
[33]	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗
[34]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗
[35]	✓	✓	✗	✗	✓	✗	✗	✓	✓	✓
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

SF1: Authentication, SF2: Mobility, SF3: Dynamic Topology
SF4: Parallel Execution , SF5: Eavesdropping Attack, SF6: DoS Attack
SF7: Proxy Attack, SF8: Man In The Middle Attack
SF9: Replay Attack, SF10: Physical Attack

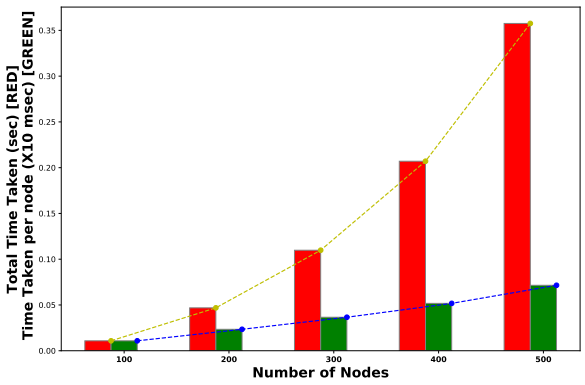


Fig. 7: Scalability Evaluation: Time taken (sec) in communication vs number of nodes [Red], Time taken (x10 msec) per node [green]

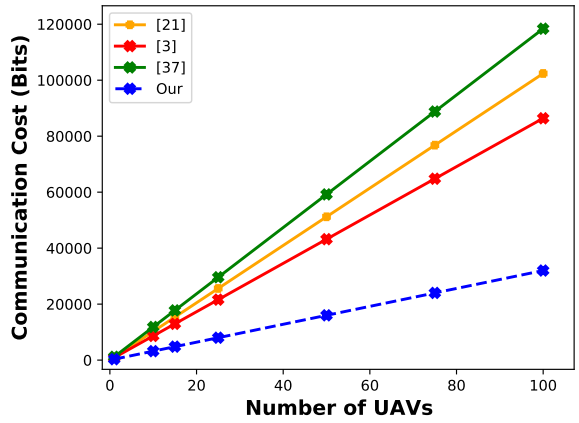


Fig. 8: Comparison of total communication cost (bits) vs number of devices for different techniques

using PUFs. Also, only [33] has implemented resistance to proxy attacks. The common drawback in all the existing works is the lack of support for handling dynamic network topologies. Most of these works are based on one-to-one authentication and are not scalable to UAV swarms. Using a spanning tree, the proposed protocol offers dynamic execution, scalability, and parallel performance. It also avoids DoS attacks by choosing a new initial point in each iteration. Finally, PUF based response generation in the proposed protocol provides protection against node tampering attacks.

VII. Simulation and Computational Performance

This section compares the proposed technique with the state of the art techniques for authentication in UAV networks. To evaluate the various techniques, the operations of UAVs were performed on a Raspberry Pi 3B device. The base station operations were evaluated on Mac OS (1.8 GHz Dual-Core Intel Core i5, 8 GB 1600 MHz DDR3) device. For our simulations, 20 UAVs were deployed in a region and served by a base station, as shown in Fig. 6 and the code was implemented using the Python programming language.

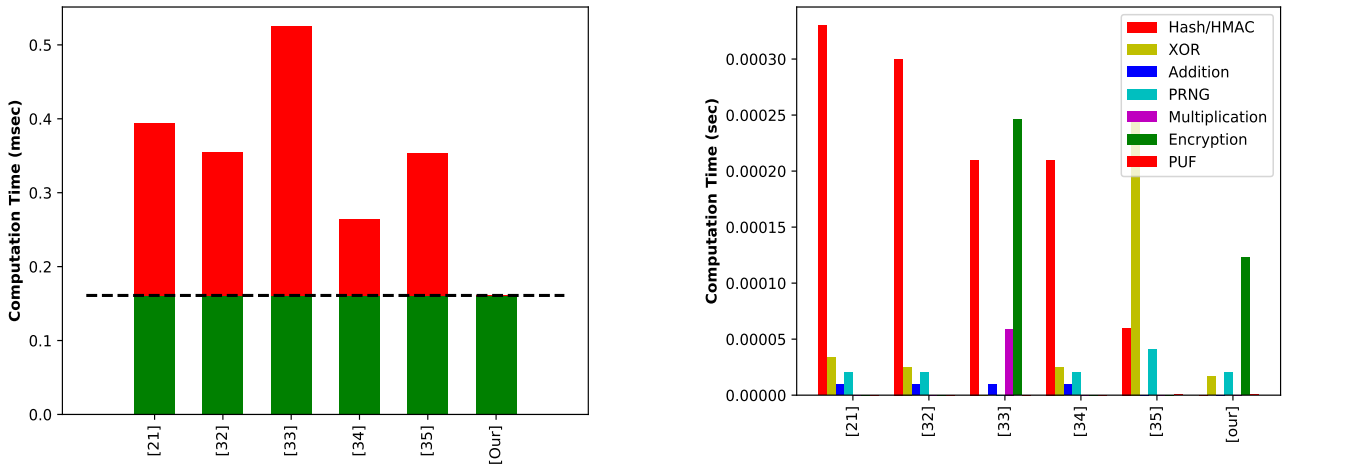
The protocol works with the base station initiating the protocol and construction of the spanning tree. Figure 6(a) depicts the initial stage, when the system is deployed, but protocol execution has not started. In Fig. 6(b), the base station randomly selects two UAVs, connected through an edge (highlighted in blue) and checks if both UAVs are part of the same spanning tree component or not. The component of an UAV is the set of all the vertices connected by a path to that UAV. If there is an element in the disjoint sets of both UAV components, the edge is finalised as shown in Fig. 6 (c). Figure 6 (d) presents a situation when most of the spanning tree edges are finalized. Figure 6 (e) describes the case when the blue line or (trial edge) fails to be confirmed since the trial edge's vertices or UAVs are already part of the same component. So the blue edge is dropped. The base station then chooses the next random edge (shown in Fig. 6 (f)), which gets confirmed in Fig. 6 (g), marking the confirmation of edge (shown in red between the vertices) and completion of spanning tree protocol.

To evaluate the scalability of the proposed protocol, Fig. 7 shows the total time taken as the number of nodes increases. The time taken for communication takes around 0.014 sec for 100 nodes, which increases to 0.11 sec when the number of nodes is tripled (300) and to 0.37 sec when the number of UAVs is five-fold (500). We can observe that time taken per UAV for protocol creation is approximately linear (0.00014 sec (100 UAVs) to 0.00074 sec (500 UAVs)), i.e., order of communication time is $\mathcal{O}(n)$. Thus, our protocol is scalable with increasing number of devices.

We use a Raspberry Pi 3B device to simulate UAVs in our system model, and to run commonly used mathematical and cryptographic operations such as XOR, pseudo-random number generation (PRNG), hash (SHA-1), HMAC (SHA-1), and concatenations. We consider a recent PUF proposed in [36] to be deployed in the UAVs for our protocol. The PUF generates a response of 320-bit and the PUF operation time is 0.4 μs.

We first compare the proposed protocol with other protocols in terms of communication cost. Standard sizes for the different fields communicated across the entities are chosen similar to [20] and [3], where the timestamp is 32 bits long, nonce is 128 bits long, and hash, encryption, MAC, and device ID are each 160 bits long. As shown in Fig. 8, based on these sizes, the communication overhead in [20] [3], and [37] is 1024 bits, 864 bits, and 1184 bits, respectively, while in our protocol, the communication overhead is just 320 bits per UAV. As the number of UAVs increases, the total communication overhead increases. Thus, we justify the scalability of our protocol both in terms of execution time and communication overhead.

Figure 9 (a) illustrates the comparison of the total time taken for the execution of our proposed protocol with the protocols in [20, 32–35]. The dotted line shows the time taken by our protocol, and the red region in the graph is the difference in the execution time between our method



(a) Time taken (in sec) for 10 iterations of protocols.

(b) Computation time in different operations for different protocols

Fig. 9: Comparison for time taken for different protocols. In left figure: Red region is amount of extra time spent in techniques compared to proposed model

and other state of the art protocols. While [20], [32], [33], [34] and [35] have computation costs of 394µs, 355µs, 525µs and 265µs and 354µs, respectively, our protocol has a cost of only 161µs. It is also important to note that that none of these protocols are capable of authenticating multiple UAVs at a single execution instance. In our study, when works [35] and [34] were extended to multihop UAV scenarios, their execution time increased to 200% higher than our protocol.

Figure 9(b) gives an insight into the time consumed by different operations by various protocols. We have not shown PUF operation as it consumes 0.4µs, which is very small compared to the time consumed by other operations. Apart from our proposed method, PUF operations are also used in [35]. Hash operations take the major portion of time computation of the protocols, constituting 83.75%, 72.6% and 65% of the time in [20], [32] and [34], respectively. In contrast, the proposed protocol does not employ any hash operation. In our protocol, the major computation operation is encryption, which is necessary to ensure the resilience of the protocol against a multitude of attacks and cannot be eliminated for the sake of scalability. We used the AES encryption scheme (taking 61.6µs for one execution). Also, we used a pseudo random function for generating nonce (128 bits) to provide freshness, rather than message exchanges, resulting in speeding up of the protocol by 20.3µs and 40.6µs from [20] and [35], respectively. Only [33] uses multiplication operations (60µs) instead of XOR or PRNG.

VIII. Conclusion

In this paper, we presented a scalable protocol for mutual authentication in UAV swarm networks. The proposed approach is based on the use of a spanning tree to allow the protocol to function even in dynamic topologies and where UAVs are mobile. The proposed protocol ensures physical security using Physical Unclonable Functions and is also resistant to man in the middle attacks, replay attacks, DoS attacks, etc. We show that the proposed protocol performs better in terms of computation cost and performance compared to other state of the art protocols, while also being the only one that provides authentication to UAV swarms.

IX. Appendix

The rules in the Mao-Boyd logic are as follows:

1) Authentication rule :

$$\frac{\mathcal{X} \models \mathcal{X} \overset{K}{\leftrightarrow} \mathcal{Y} \wedge \mathcal{X} \overset{K}{\lhd} M}{\mathcal{X} \models \mathcal{Y} \overset{K}{\vdash} M}.$$

2) Nonce-verification rule :

$$\frac{\mathcal{X} \models \#(M) \wedge \mathcal{X} \models \mathcal{Y} \overset{K}{\vdash} M}{\mathcal{X} \models \mathcal{Y} \models \mathcal{X} \overset{K}{\leftrightarrow} \mathcal{Y}}.$$

3) Confidentiality rule :

$$\frac{\mathcal{X} \models \mathcal{X} \overset{K}{\leftrightarrow} \mathcal{Y} \wedge \mathcal{X} \models S^c \lhd \| M \wedge \mathcal{X} \overset{K}{\vdash} M}{\mathcal{X} \models (S \cup \{\mathcal{Y}\})^c \lhd \| M}.$$

4) Super-principal rule :

$$\frac{\mathcal{X} \models \mathcal{Y} \models X \wedge \mathcal{X} \models sup(\mathcal{Y})}{\mathcal{X} \models X}.$$

5) Intuitive rule :

$$\frac{\mathcal{X} \overset{K}{\lhd} M}{\mathcal{X} \lhd M}.$$

6) Good key rule :

$$\frac{\mathcal{X} \models \{\mathcal{X}, \mathcal{Y}\}^c \lhd \| K \wedge \mathcal{X} \models \#(K)}{\mathcal{X} \models \mathcal{X} \overset{K}{\leftrightarrow} \mathcal{Y}}.$$

7) Fresh rule :

$$\frac{\mathscr{X} \models \#(\mathrm{M}) \wedge \mathscr{X} \lhd \mathrm{N}\mathbf{R}\mathrm{M}}{\mathscr{X} \models \#(\mathrm{N})}.$$

REFERENCES

[1] N. Hossein Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[3] S. Jangirala, A. K. Das, N. Kumar, and J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, 2019.

[4] D. Wang, C.-g. Ma, P. Wang, and Z. Chen, "Robust smart card based password authentication scheme against smart card security breach," *Cryptol. ePrint Archive*, vol. 2012, no. 439, pp. 1–35, 2012.

[5] D. Fang, Y. Qian, and R. Q. Hu, "Security for 5g mobile wireless networks," *IEEE Access*, vol. 6, pp. 4850–4874, 2017.

[6] L. Hong, H. Guo, J. Liu, and Y. Zhang, "Toward swarm coordination: Topology-aware inter-uav routing optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 177–10 187, 2020.

[7] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[8] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[9] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[10] C. Zhong, J. Yao, and J. Xu, "Secure uav communication with cooperative jamming and trajectory control," *IEEE Communications Letters*, vol. 23, no. 2, pp. 286–289, 2018.

[11] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[12] A. Birk, B. Wiggerich, H. Bülow, M. Pfingsthorn, and S. Schwertfeger, "Safety, security, and rescue missions with an unmanned aerial vehicle (uav)," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 1, pp. 57–76, 2011.

[13] X. Sun, D. W. K. Ng, Z. Ding, Y. Xu, and Z. Zhong, "Physical layer security in uav systems: Challenges and opportunities," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 40–47, 2019.

[14] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[15] M. Hooper, Y. Tian, R. Zhou, B. Cao, A. P. Lauf, L. Watkins, W. H. Robinson, and W. Alexis, "Securing commercial wifi-based uavs from common security attacks," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 1213–1218.

[16] O. Blazy, P.-F. Bonnefoi, E. Conchon, D. Sauveron, R. N. Akram, K. Markantonakis, K. Mayes, and S. Chaumette, "An efficient protocol for uas security," in *2017 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2017, pp. 1–21.

[17] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, and M. Robinson, "Security authentication system using encrypted channel on uav network," in *2017 First IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2017, pp. 393–398.

[18] S. Benzarti, B. Triki, and O. Korbaa, "Privacy preservation and drone authentication using id-based signcryption." in *SoMeT*, 2018, pp. 226–239.

[19] H. Nicanfar, P. Jokar, and V. C. Leung, "Smart grid authentication and key management for unicast and multicast communications," in *2011 IEEE PES Innovative Smart Grid Technologies*. IEEE, 2011, pp. 1–8.

[20] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3572–3584, 2018.

[21] P. Gope and 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. 13 621–13 630, 2020.

[22] L. Chen, S. Qian, M. Lim, and S. Wang, "An enhanced direct anonymous attestation scheme with mutual authentication for network-connected UAV communication systems," *China Communications*, vol. 15, no. 5, pp. 61–76, 2018.

[23] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "Seda: Scalable embedded device attestation," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 964–975.

[24] A. Ibrahim, A.-R. Sadeghi, G. Tsudik, and S. Zeitouni, "Darpa: Device attestation resilient to physical attacks," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2016, pp. 171–182.

[25] F. Kohnhäuser, N. Büscher, S. Gabmeyer, and S. Katzenbeisser, "Scapi: a scalable attestation protocol to detect software and physical attacks," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 75–86.

[26] M. Ambrosin, M. Conti, R. Lazzeretti, M. M. Rabbani, and S. Ranise, "Pads: practical attestation for highly dynamic swarm topologies," in *2018 International Workshop on Secure Internet of Things (SIoT)*. IEEE, 2018, pp. 18–27.

[27] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for v2g using puf," *IEEE Transactions on Vehicular Technology*, 2020.

[28] G. Bansal, N. Naren, and V. Chamola, "Rama: Real-time automobile mutual authentication protocol using puf," in *Proceedings of IEEE International Conference on Information Networking (ICOIN), Barcelona, Spain*. IEEE, 2020.

[29] I. Cervesato, "The dolev-yao intruder is the most powerful attacker," in *16th Annual Symposium on Logic in Computer Science—LICS*, vol. 1, 2001.

[30] T. Alladi, N. Naren, G. Bansal, V. Chamola, and M. Guizani, "Secauthuav: A novel authentication scheme for uav-base station scenario," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[31] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *[1993] Proceedings Computer Security Foundations Workshop VI*. IEEE Comput. Soc. Press, pp. 147–158. [Online]. Available: http://ieeexplore.ieee.org/document/246631/

[32] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6903–6916, 2019.

[33] G. K. Verma, B. Singh, N. Kumar, and D. He, "Cb-ps: An efficient short-certificate-based proxy signature scheme for uavs," *IEEE Systems Journal*, vol. 14, no. 1, pp. 621–632, 2019.

[34] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: a lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43 711–43 724, 2020.

[35] T. Alladi, V. Chamola, N. Kumar *et al.*, "Parth: A two-stage lightweight mutual authentication protocol for uav surveillance networks," *Computer Communications*, 2020.

[36] X. Zhao, Q. Zhao, Y. Liu, and F. Zhang, "An ultracompact switching-voltage-based fully reconfigurable rram puf with low native instability," *IEEE Transactions on Electron Devices*, vol. 67, no. 7, pp. 3010–3013, 2020.

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