SHOTS: Scalable Secure Authentication-Attestation Protocol Using Optimal Trajectory in UAV swarms

Gaurang Bansal, Naren, Vinay Chamola, *Senior Member, IEEE*, Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Unmanned Aerial Vehicles (UAVs) have enabled a broad spectrum of applications serving social, commercial, and military purposes. However, since UAVs use wireless communication technologies, they are highly vulnerable to security threats. Establishing trust with the base station is the most fundamental security aspect in UAV networks to mitigate these threats. However, due to a UAV's constrained resources, deploying traditional trust establishment schemes in UAV networks becomes challenging. Further, this issue escalates as the number of UAVs increases. To address this issue, we propose an authentication cum attestation protocol for UAV swarms using an optimal communication trajectory, which can establish the required trust in a lightweight manner. Furthermore, our protocol uses Physical Unclonable Functions (PUFs) and thus guarantees physical security as well. We demonstrate that the proposed protocol is feasible, scalable, and secure using a formal Mao Boyd logic approach. Comparative analyses show that the proposed protocol outperforms the state-of-the-art.**

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

I. INTRODUCTION

While the development of UAV-based applications has seen a steep increase in the last decade, the peak is yet to be reached. Many upcoming applications require UAVs to operate in close vicinity to end-users, but this poses significant risks to the UAV's security. UAVs are prone to physical attacks such as capture and tampering and several network attacks such as MITM, replay, and cloning. [1, 2].

Authentication and attestation are two security mechanisms required for proper functioning and ensuring security in UAV communications. Authentication is a mechanism used by communicating parties to establish that each of them is a valid device [3, 4]. Authentication is a mechanism used by communicating parties to validate the identity or source of a message or information flow between them. However, malicious attackers may try to gain control/compromise either/both of the communicating parties at any time in a dynamic network. Hence, authentications are required to be carried out periodically to prevent malicious attackers from gaining access to communications [5].

Attestation is used to verify if the device's memory/firmware is unchanged [6]. A UAV's firmware is at risk of modification due to malicious users who may try to reprogram the UAV either wirelessly or while interacting with it during an application-specific context (say, for example, drone delivery). A modified firmware can result in improper functioning and security loopholes during the UAV's operations, and hence attestation is also a vital security objective in UAV communications. In attestation, a prover is an entity whose firmware is to be attested, and a verifier is an entity that attests the prover [7, 8].

This paper presents a lightweight and scalable authentication cum attestation protocol for communication between UAVs and base stations. We consider the scenario of a swarm of UAVs, which, after deployment, needs to be attested and authenticated at regular intervals. The size of a UAV swarm can vary greatly, and hence, the proposed protocol is designed to be scalable and lightweight. Although several similar works have been published in recent literature (discussed in Section II), they have significant performance, security, and scalability issues.

The major contributions of the paper are as follows:

1) We propose a secure, lightweight, and scalable authentication cum attestation protocol for a swarm of UAVs with a Base Station (BS).
2) A scalability of the order of $\mathcal{O}(n)$ in computational complexity has been achieved.
3) We use a run time christofides algorithm to account for swarm dynamicity and mobility.
4) The proposed protocol achieves message integrity, mutual authentication, attestation, confidentiality, and protection against denial of Service, man-in-the-middle (MITM), replay, impersonation, and cloning attacks. By the use of PUF, our protocol also achieves physical security.

The rest of this paper is organized as follows. Related works are discussed in Section II. In Section III, we discuss the system and adversarial models along with an overview of PUFs. Section IV presents optimal trajectory generation using the christofides algorithm followed by the proposed protocol in Section V. Security analysis of protocol is presented in Section VI. We provide a performance analysis of the proposed protocol in Section VII. Finally, the conclusions of the paper are presented in Section VIII.

II. RELATED WORKS

The network characteristics of a UAV swarm are quite different from other distributed networks mainly due to their mobility which is higher than that of Mobile Ad-Hoc Networks (MANETs) and Vehicular Ad-Hoc Networks (VANETs). Network topology is another factor where UAV swarms display a high dynamic nature. Additionally, UAVs are constrained in both computation and battery storage, and thus it is desirable

Gaurang Bansal, Naren, Vinay Chamola and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, (e-mail: e0622339@u.nus.edu, naren.mysore97@gmail.com, vinay.chamola@u.nus.edu and bsikdar@nus.edu.sg).

to develop lightweight algorithms for implementing security features in their operations. Due to the above-mentioned unique characteristics, the security requirements of a UAV swarm are much different from other Ad-Hoc networks. In the past few years, several works [9, 10, 11] have tried to develop security schemes that are computationally lightweight and fast, yet highly secure. We briefly discuss below some of the authentication and attestation solutions proposed for securing communications in UAV networks.

Jiang et al. have presented an Artificial Intelligence-based UAV identification and authentication mechanism which relies on behavioral data such as the UAV's location collected on a real-time basis [12]. This work is, however, limited to the authentication of a single UAV. Yahuza et al. have presented a secure lightweight proven authenticated key agreement (SLPAKA) [13]. Using Mobile Edge Computing (MEC), they achieve scalability, i.e., multiple UAVs can be dynamically authenticated on the network. This scheme is, however, vulnerable to physical attacks. In [14], the authors present a privacy-preserving authentication protocol for the Internet of Drones - for authenticating a UAV with a UAV service provider through a MEC device without loss of privacy. A point to note about this work is the use of PUFs to ensure physical security. Each UAV has two PUF devices that are used during the authentication. The authors of [1] present a UAV-base station and a UAV-UAV authentication scheme using a single PUF device on each UAV. The authors of [15] present a PUF based authentication scheme for authenticating a two-layered swarm of drones. A larger leader drone controls several mini drones in its vicinity. Their protocol consists of authenticating the leader drone with the base station and authenticating a mini drone with the leader drone. Both [1, 15] provide security against physical attacks by the use of PUFs. However, [1] does not address scalability and [15] is applicable to only a fixed two-level scenario.

In [16], the authors present an authentication protocol for UAV swarms based on spanning-tree topology, but it has been found to lack resiliency [17]. Chen et al. present a direct anonymous attestation for network-connected UAV (NC-UAV) systems [18]. Their scheme utilizes Trusted Platform Modules (TPMs), dedicated microcontroller, or cryptoprocessor for storing secret credentials. These are generally highly expensive, and hence the feasibility of their usage in commercial UAVs is uncertain. In [19], the authors present the Practical Attestation for Highly Dynamic Swarm Topologies (PADS). Their model is scalable and can be employed in unstructured networks. However, it is meant for use in networks of autonomous devices operating without a central controlling entity such as a Base Station.

As discussed above, most of the present works deal with either attestation or authentication, except for [18] which includes both attestation and authentication and uses expensive TPMs. The remaining works which deal with either attestation or authentication fall short in terms of physical security, scalability, and limited usage scenarios. Hence, there is the need for a robust, physically secure, and scalable authentication cum attestation protocol for UAV swarms that can quickly and in a lightweight manner authenticate all the UAVs in a swarm with a central base station for security and privacy-preserving operations. Therefore, in this paper, we present SHOTS: a scalable authentication-attestation protocol using optimal trajectory in UAV swarms. SHOTS first determines an optimal communication trajectory from the BS to the farthest UAV. This path will be used to determine the flow of authentication messages from the BS to all the UAVs in a hop-by-hop manner. It relies on PUFs to achieve physical security and simple cryptographic operations to ensure lightweight computation on the UAVs.

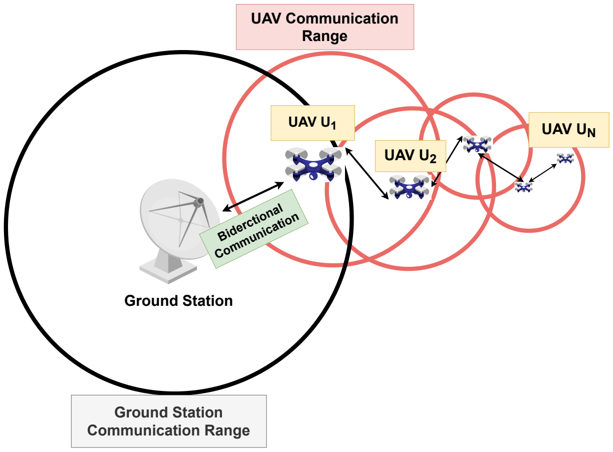

Fig. 1: System Model

III. System Model

The system model used in this paper is presented in Fig. 1. As shown in the figure, several UAVs in a swarm carry out a collective objective, and a central base station coordinates their operation. The BS is assumed to be trusted, while each UAV has to be repeatedly authenticated and attested at regular intervals to detect and prevent malicious entities from gaining control over one or more UAVs of the swarm. It is assumed that at least one UAV remains in the coverage area of the BS. Similarly, it is assumed that the entire UAV swarm is either directly or indirectly (through a hop-by-hop manner) in the communication range of the BS. Each UAV of the swarm is equipped with a single physical unclonable function (PUF) chip that acts as the basis for securely identifying each UAV and ensures protection against device tampering attacks. Details of PUF are mentioned in the following subsection. All UAVs are clock-synchronized with each other and the BS.

It is assumed that the adversary can try to launch various network-related attacks such as Man-In-The-Middle (MITM), replay, impersonation, and message replay attacks. In addition, the adversary may even try to capture the UAV physically and try to tamper with the device to extract its secret credentials.

A. Background of PUF

PUFs are derived from the randomness and process variations in the fabrication of an integrated circuit and serve as unique fingerprints for a particular chip [20, 21]. It is used in a challenge-response mechanism where a PUF device is evaluated with a challenge C to obtain a response R (C and R are called Challenge-Response Pair - CRP). A single PUF device will show different responses to different challenges, and two PUF devices will not show the same response for the same challenge. It is assumed that the construction of the PUF in the device is such that any attempts by the attacker/capturer of the UAV to extract/tamper with the PUF will render it unusable, thus preventing the particular UAV from being able to communicate with the BS [22].

IV. Optimal Trajectory for Protocol

In this section, we consider the Christofides topology for determining an optimal message flow trajectory among the UAVs using the Christofides algorithm [23]. The heuristics used for optimizing the message flow path leverage the triangle inequality. The distance of communication between UAV x and UAV z must be less than the sum of the distance of communication between UAV x and UAV y and UAV y and UAV z.

$$w(x,y) + w(y,z) \leq w(x,z) \quad \forall x,y,z \in \mathcal{V} \tag{1}$$

Once the base station knows each UAV's location (shown in Fig. 2), creating an optimal path can be considered a traveling salesman problem. The UAV network forms a complete graph G. The UAV locations are the vertices of G. The logical connection between two UAVs forms an edge in graph G. The salient steps involved in the Christofides algorithm are explained in the following subsections.

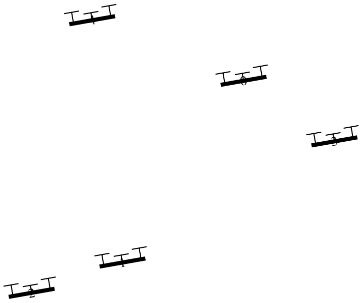

Fig. 2: Initial Positioning of UAVs

A. Creation of Spanning Tree

A spanning tree is a subset of a graph containing all the vertices with a minimum number of edges. We create a spanning tree using Algorithm I. Algorithm I is inspired from the Kruskal algorithm [24] which uses the greedy approach to find the minimum cost spanning tree. A tree connects to another if it has the least cost among all available options and does not violate spanning-tree properties like cycle formation. The resultant spanning tree has edges one less than the number of vertices or nodes (shown in Fig. 3). In the following subsection, we find odd degree vertices. A vertex with an odd number of edges incident on it is called an odd degree vertex.

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 */
$j \leftarrow \phi$
/* 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
        $j = j \cup (u,v)$;
        Component[u] = Component[u] ∪ Component[v];
        Component[v] = Component[u] ∪ Component[v];
    end
end
```

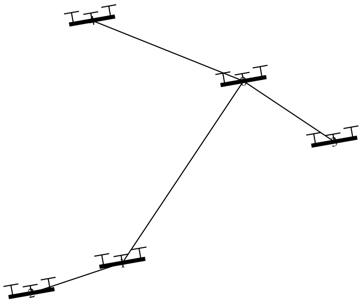

Fig. 3: Creation of Spanning Tree

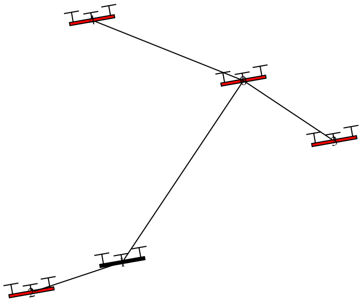

Fig. 4: UAVs with odd degree in Spanning Tree (Marked with Red)

B. Finding Vertices With An Odd Degree

After creating the spanning tree, our resultant graph covers all the vertices and ensures no cycle. To convert a spanning tree into an Eulerian tour, we find vertices with odd degrees in linear search time $O(n)$ as shown in Fig. 4. We convert these odd degree vertices to even degrees by forming the perfect matching in the following subsection. The need to convert the degree of vertices to even is to find an Eulerian path. Carl Hierholzer, in his work [25], proved that all vertices must have even degrees for the Eulerian circuit to exist.

C. Perfect Matching & Handshaking Lemma

Figure 4 shows a minimum spanning tree (let us call it T), where odd degree vertices are marked red and even degree vertices are black. We connect all the odd-degree vertices to obtain a subgraph. A perfect matching [26] is supposed to cover every vertex of the graph. A perfect matching in G=(V, E) is a subset of E such that every vertex in V is adjacent to exactly one edge in the subset as shown in Algorithm 2. While there are many possibilities of perfect matching, we consider one such case in Fig. 5. Let us refer to this perfect matching as M. We then take the union of the spanning tree T and the perfect matching M. According to the Handshaking lemma in graph theory, a finite undirected graph has an even number of odd degree vertices. When we perform perfect matching on a graph formed by odd vertices, each vertex forms part of only one edge. So, when we take the union of T and M, the degree of every odd vertex is increased by one. Thus, all odd degree vertices will eventually become even degree vertices.

Algorithm 2 Greedy Perfect Matching Algorithm

```
Input  : Graph G
         $V_o$: Set of Odd degree vertices
Output: Graph G with even degree
for $v$ in $V_o$ do
    length ← inf
    for $u \in V_o$ do
        if $weight(u, v) < length$ then
            $length \leftarrow weight(u, v)$
            $closest \leftarrow u$
    G ← G $\cup E(closest, v)$
    $V_o \leftarrow V_o$ - v
    $V_o \leftarrow V_o$ - closest
return $G$
```

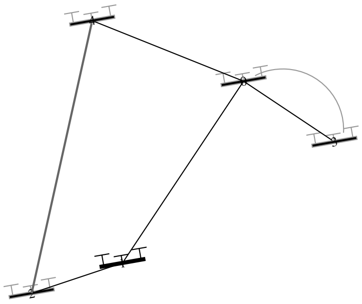

Fig. 5: Find a perfect minimal matching on odd vertices

D. Eulerian Tour

With even degrees at all vertices, we now create an Eulerian circuit or Eulerian cycle. An Eulerian circuit constructs a path from the initial vertex and visits all the edges exactly once using Algorithm 3. In Algorithm 3, we choose the closest UAV to BS as the starting vertex. The starting vertex is added to the path and pushed to a stack. To find the Eulerian tour, we then pop the top of the stack (current vertex) and push its neighbors (directly connected to the current vertex) to the stack. This process is repeated until no neighbor is remaining and the stack is empty. Hence the resultant graph gives us the Eulerian circuit shown in Fig 6.

Algorithm 3 Euler Path Algorithm

```
Input  : G: Graph (Adjacency Matrix)
         V: Set of Vertices
         L: List of Edges
Output: P: Directed Path
$k \leftarrow V[0]$;
Stack s;
V = V - k
s.push(v)
while $s$ != Empty do
    v = k
    length ← inf
    for $u \in V$ do
        if $weight(u, v) < length$ then
            $length \leftarrow weight(u, v)$
            $closest \leftarrow u$
    if  $P = P \cup v$
    $v$ = s.pop() then length == inf
    E← $E$ - (v,closest)
    s.push(closest)
    v = closest
return P
```

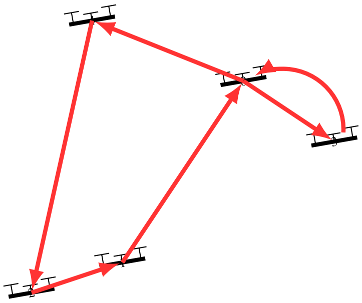

Fig. 6: Euler circuit

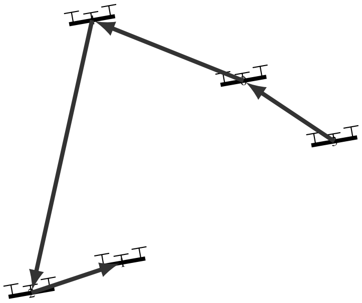

Fig. 7: Final Hamiltonian path by skipping visited nodes

E. Hamiltonian Path

Finally, while moving along the Euler path, we check if the node is visited or not. If it is not visited, it is added to the Hamiltonian path depicted in Fig. 7. Else, we skip the node and move on. This skipping will not increase the length as the graph satisfies the triangle inequality.

F. Optimal Traversal

Once the Hamiltonian path is finalized, the base station generates the path P. It knows the identity of each UAV in the path. Finally, it proceeds with the execution of the protocol described in the following section.

V. Proposed Protocol

In this section, we describe the working of the SHOTS protocol.

A. Initialisation

Before the UAV swarm is deployed for carrying out its task, a trusted BS records a single challenge-response pair (CRP) for each UAV of the swarm in its storage. There are N UAVs $U = \{U_1, U_2, ...U_N\}$ at a given instance. All the UAVs in U are registered with the BS, i.e., a CRP for each UAV in U is stored in the BS's memory. These CRPs are denoted as $CRP = \{CRP_1, CRP_2, ...CRP_N\}$. Further, each CRP_j is composed of a challenge and a response, i.e., $CRP_j = \{C_j, R_j\}$.

B. Communication

The BS verifies from time to time the identity and integrity of the connected UAVs. To verify the authenticity, the BS sends a single composite message M to all the UAVs in an optimal path P as determined by the cristofides algorithm (discussed in Section IV). This optimal path provides a list of nodes in which a message is communicated. For each UAV, BS generates an authentication message M_j or the j^{th} UAV in the path. Having generated all M_j messages, BS sends M. M can be written as $M = \{M_1, M_2, ...M_N\}$ where M_j is the sub-messages meant for the j^{th} UAV in the path, i.e., U_j.

Each $M_j = C_j||Ts||Enc([N_B, Ts], R_j)$ is composed of an unencrypted part - the corresponding challenge and a timestamp $C_j||Ts$, and an encrypted part - $Enc([N_B, Ts], R_j)$. Here, C_j and R_j are the challenge and response of the j^{th} UAV in path P. N_B is a nonce and Ts is a timestamp generated at the time instant when the message was composed by the BS. As shown in Fig 8, the message M travels along the path P, and each UAV U_j extracts its corresponding message M_j based on its index in P. Finally, each UAV forwards this received message to the next UAV in the path.

C. Authentication

As mentioned above, on extracting M_j, the UAV U_j obtains the challenge C_j, timestamp Ts, and the encrypted string $Enc([N_B, Ts], R_j)$. U_j checks whether the difference between its current time stamp T and received time stamp Ts is less than ϕ. UAV also checks if the plaintext T_s matches with the T_s that was decrypted using R_j. This check ensures temporal binding of the plaintext and ciphertext portions of M_j. If yes, U_j will evaluate the response of C_j as $R_j = PUF(C_j)$, else, authentication is stopped and message is discarded. UAV U_j also checks the equality of the received R_j and the computed value PUF(C_j). If authentication proceeds, U_j will decrypt $Enc([N_B, Ts], R_j)$ to get N_B, and Ts using R_j. Now, U_j will generate a new nonce N_U, new challenge response pair (C'_j, R'_j) and encrypt it in the partial return message $Re_j = C_j||T||Enc([N_U, Ts], C'_j, R'_j, csum*_j], R_j)$, where $csum*_j$ is the checksum computed by the UAV. More details about this is explained in the next subsection. U_j also calculates a unique session key Sk_j for its communication with the BS.

The last UAV in the path initiates the reply message with Re_n while subsequent UAVs in the return path concatenate their reply message Re_j with the reply message received from its decedent UAV (Re_{j+1}). In this manner, the first UAV (U_1) in the path forms the entire reply message containing the replies of all UAVs and sends it to the BS.

On receiving the entire reply message, the BS checks if the difference between its current timestamp Ts and received timestamp T is less than ϕ. If yes, the BS will decrypt each reply message Re_j with the corresponding response R_j and obtain the new CRP C'_j, R'_j for each UAV U_j. This is possible if UAV's PUF response R_j matches the response stored in memory of the base station. It will also obtain their respective nonces (N_U) and checksums ($csum_j$) from reply messages from UAV to the base station. Then, it will verify whether the checksums $csum_j$ and $csum*_j$ are equal. If this condition is satisfied, then the UAVs have been successfully attested. It will now generate a unique session key $SK_j = N_U \oplus N_B$ for communicating with each UAV.

D. Attestation

The Base station attests to each UAV by verifying different memory blocks on the UAV. The addresses of the memory blocks to be attested are generated at random during each iteration. Both BS and UAV generate the address of the memory block of the UAV's firmware to be attested using pseudo-random generator function (PRNG). This PRNG takes T_s as input. Since T_s is unique, the output memory location δ is unique for each round. The BS retrieves the memory content π_j corresponding to the memory location δ of the j^{th} UAV. Note that the entire contents of each UAV's firmware were acquired during the registration phase. Then, it uses the SHA Hash function to hash the output to constant string length $H(\pi_j)$. This, we refer to as $csum_j$.

The UAV on receiving message M_j from BS, compares its own timestamp T to that of received message T_s. If the difference is less than ϕ, the protocol proceeds. Each UAV also evaluates δ using the message timestamp T_s received from BS. Then, the UAV U_j uses δ to calculate $csum^*_j$, where

$$csum^*_j \Longleftarrow \mathrm{H}(Mem^*_j(\delta)) \tag{2}$$

U_j encrypts $csum^*_j$ using R_j and sends as part of Re_j. The BS on receiving the complete response message M' checks the value of $csum^*_j$ received to the evaluated ($csum_j$). If they match, the device is successfully attested.

VI. Formal Security Analysis

We provide a formal security analysis of our protocol by modeling the communication in the protocol using Mao Boyd logic [27]. The notations for symbols as used by Mao Boyd logic are presented as follows:

1) $U_j | \equiv BS$: U_j believes BS.

SHOTS Authentication & Attestation Protocol

Base Station		UAV_j
$P \Longleftarrow$ Perform optimal path generation (Section IV)		
$(C_j, R_j) \Longleftarrow$ Extract (C_j, R_j) from memory		
$Ts \Longleftarrow$ Current Timestamp		
$N_B \Longleftarrow$ Generate Nonce		
$M_j \Longleftarrow C_j \vert\vert Ts \vert\vert Enc([N_B, Ts], R_j)$		
$M \Longleftarrow M_1 \vert\vert M_2 \vert\vert ... \vert\vert M_N$		
	$\xrightarrow{\text{Send M,P}}$	
		$j \Longleftarrow$ Identify device position in P using its ID
		$M_j \Longleftarrow$ Extract M_j from M
		Communication with UAV_{j+1}
		$\xrightarrow{\text{Send M,P}}$
		$\xleftarrow{\text{Receive } Re_{j+1}}$
		$C_j, Ts, Enc([N_B, Ts], R_j) \Longleftarrow$ Extract from M
		$T \Longleftarrow$ Current Timestamp
		if $((T - Ts) \leq \phi)$ **then**
$\delta \Longleftarrow PRNG(Ts)$		$R_j \Longleftarrow$ PUF(C_j)
$\pi_1, \ldots \pi_j, \ldots \pi_m \Longleftarrow$ Mem$_1(\delta), \ldots$ Mem$_j(\delta), \ldots$ Mem$_m(\delta)$		Using R_j as decryption key
$csum_1, \ldots csum_j, \ldots csum_m \Longleftarrow$ H$(\pi_1), \ldots$ H$(\pi_j), \ldots$ H(π_m)		$N_B \Longleftarrow$ Dec(Enc([N_U, Ts],R_j), R_j)
		$N_U \Longleftarrow$ Generate Nonce
		$(C'_j, R'_j) \Longleftarrow$ Extract (C'_j, R'_j) from PUF
		$\delta \Longleftarrow PRNG(Ts)$
		$\pi^*_j \Longleftarrow Mem^*_j(\delta)$
		$csum^*_j \Longleftarrow$ H(π^*_j)
		$Re_j \Longleftarrow C_j \vert\vert T \vert\vert Enc([N_U, T, C'_j, R'_j, csum^*_j, R_j)$
		$\mathbf{Sk_j} \Longleftarrow \mathbf{N_U} \oplus \mathbf{N_B}$
	$\xleftarrow{\text{Send } Re_j \vert\vert Re_{j+1}}$	
$(C_j, R_j) \Longleftarrow$ Extract (C_j, R_j) from memory		
$Ts \Longleftarrow$ Current Timestamp		
if $((T - Ts) \leq \phi)$ **then**		
$(C'_j, R'_j) \Longleftarrow$ Dec(Enc([N_U, T],C'_j, $R'_j, csum^*_j$), R_j)		
$N_U, csum^*_j \Longleftarrow$ Dec(Enc([N_U, T],C'_j, $R'_j, csum^*_j$), R_j)		
Verify **if** $(csum_j == csum^*_j)$ **then**		
Successfully Attested		
$\mathbf{Sk_j} \Longleftarrow \mathbf{N_U} \oplus \mathbf{N_B}$		

Fig. 8: SHOTS Authentication & Attestation Protocol

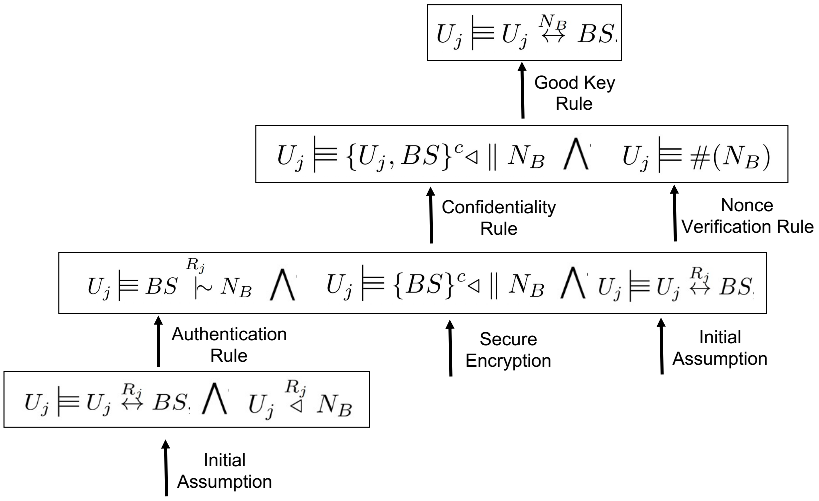
$U_j \models U_j \overset{N_B}{\leftrightarrow} BS.$
Good Key Rule
$U_j \models \{U_j, BS\}^c \triangleleft \parallel N_B \bigwedge U_j \models \#(N_B)$
Confidentiality Rule
Nonce Verification Rule
$U_j \models BS \overset{R_j}{\mid\!\sim} N_B \bigwedge U_j \models \{BS\}^c \triangleleft \parallel N_B \bigwedge U_j \models U_j \overset{R_j}{\leftrightarrow} BS.$
Authentication Rule
Secure Encryption
Initial Assumption
$U_j \models U_j \overset{R_j}{\leftrightarrow} BS. \bigwedge U_j \overset{R_j}{\triangleleft} N_B$
Initial Assumption

Fig. 9: Mao Boyd Proof of the proposed mutual authentication protocol between UAV U_j and base station BS.

2) $U_j \overset{K_{ij}}{\mid\!\sim} M$: U_j encrypted M using the key K_{ij}.
3) $U_j \overset{K_{ij}}{\triangleleft} M$: U_j extracts M using key K_{ij}.
4) $U_j \overset{S_k}{\leftrightarrow} BS$: S_k is valid shared key.
5) $\#(N_U)$: Nonce N_U is unique and not used before.
6) $sup(BS)$: BS is assumed to be secure and trustworthy.
7) $U_j \triangleleft \parallel M$: U_j cannot get the message M.

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

1) Authentication rule :

$$\frac{\mathcal{X} \models \mathcal{X} \overset{K}{\leftrightarrow} \mathcal{Y} \wedge \mathcal{X} \overset{K}{\triangleleft} M}{\mathcal{X} \models \mathcal{Y} \overset{K}{\mid\!\sim} M}.$$

2) Confidentiality rule :

$$\frac{\mathcal{X} \models \mathcal{X} \overset{K}{\leftrightarrow} \mathcal{Y} \wedge \mathcal{X} \models S^c \triangleleft \parallel M \wedge \mathcal{X} \overset{K}{\mid\!\sim} M}{\mathcal{X} \models (S \cup \{\mathcal{Y}\})^c \triangleleft \parallel M}.$$

3) Good key rule :

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

Lemma: U_j knows that N_B is a valid shared and secure message between U_j and BS.

Proof. We assume that PUF is secure and R_j is known only to the base station and the corresponding UAV. We also assume that the base station is trusted and cannot be compromised. Using the communication depicted in Fig. 8, we now describe the proof for the authentication phase described in Section V.

During the initialization phase or registration 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_B using R_j (ii). The Mao Boyd logic equivalents of these statements are shown below:

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

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

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_B using the key R_j.

$$U_j \models BS \overset{R_j}{\mid\!\sim} N_B. \tag{iii}$$

U_j believes BS is the super principal with respect to N_B. As per the protocol assumptions, the nonce N_B generated by BS must be fresh and unused.

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

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

As per the protocol, BS generates nonce N_B and encrypts it with R_j (only known to U_j). So, U_j is aware that no one other than BS knows N_B.

$$U_j \models \{BS\}^c \triangleleft \parallel N_B. \tag{vi}$$

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

$$U_j \models \{U_j, BS\}^c \triangleleft \parallel N_B. \tag{vii}$$

Finally, applying the good-key rule to (v), and (vii) we have,

$$U_j \models U_j \overset{N_B}{\leftrightarrow} BS. \tag{viii}$$

Hence, it is proved that U_j is convinced of the shared secret N_B 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. 9.

TABLE I: Timings of different Cryptographic Operations

Cryptographic Operations	**[28]**	**[29]**	**[30]**	**[31]**	**[32]**	**Our**
Bitwise XOR (64-bit)	3.37E-05	2.52E-05	0.00E+00	2.52E-05	2.52E-04	1.68E-05
Addition(64-bit)	9.64E-06	9.64E-06	9.64E-06	9.64E-06	0.00E+00	0.00E+00
Multiplication (64-bit binary)	0.00E+00	0.00E+00	5.90E-05	0.00E+00	0.00E+00	0.00E+00
PRNG	2.03E-05	2.03E-05	0.00E+00	2.03E-05	4.06E-05	4.06E-05
Hash SHA256 (64-Bytes)	3.30E-04	3.00E-04	2.10E-04	2.10E-04	6.00E-05	1.46E-05
Encryption/Decryption(AES) (32-Bytes)	0.00E+00	0.00E+00	2.46E-04	0.00E+00	0.00E+00	1.23E-04
PUF	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.00E-07	8.00E-07
Total computation time (s)	**3.94E-04**	**3.55E-04**	**5.25E-04**	**2.65E-04**	**3.54E-04**	**1.96E-04**

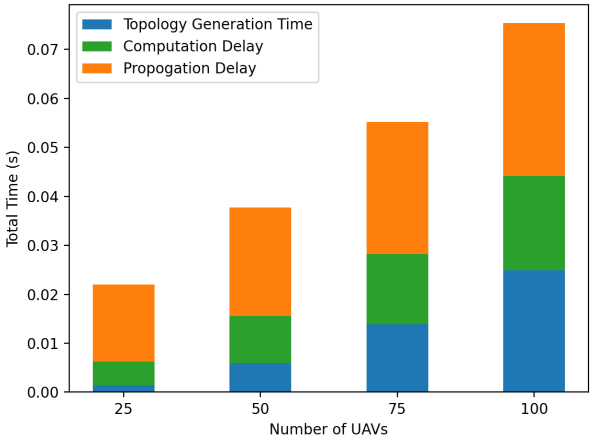

Fig. 10: Comparison of total execution time

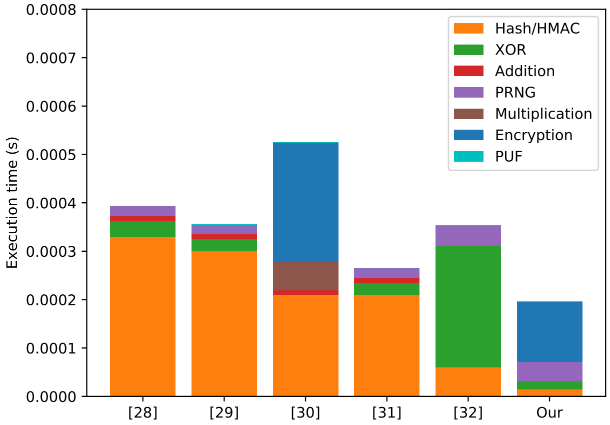

Fig. 11: Comparison of total computation time

VII. Results & Discussion

In this section, we evaluate the performance of our protocol and provide a comparison with other state-of-the-art works. The operation times for UAVs were evaluated on a Raspberry Pi 3B device. The base station operations were evaluated on a Mac OS (1.8 GHz Dual-Core Intel Core i5, 8 GB 1600 MHz DDR3) device.

Fig. 10 depicts the total time taken for protocol execution with a varying number of UAV devices. Total authentication time includes topology generation time, message propagation delay, and time spent in computation. For the works [28, 29, 30, 31, 32], the topology generation time and propagation times are same because they all provide single UAV-base station authentication.

In Fig. 10, we observe that time of authentication is majorly dictated by propagation delay. As the number of UAVs increase from 25 to 100, the propagation delay increases from $15.7ms$ to $31.2ms$. The computation delay is the aggregate time used in the computation of protocol operations. The computation delay is 192μs per device. Hence, as the number of UAVs increases, computation delay increases linearly. A detailed description of computation time is presented in Fig. 11. Also, we observe in Fig. 10 that the rate of increase of topology generation time decreases as the number of UAVs increases. The time increases 4x when UAVs are increased from 25 to 50 and 2.5x when 50 to 75. As the number of UAVs increase beyond 75, the topology generation time tends to flatten and increase very slowly. This trend is observed primarily due to ease of path formation as the number of UAVs increases. The initial position of the UAVs also affects the topology generation time. To remove this factor from our simulations, we consider 100 random initial deployments of the UAVs to determine the average time taken for topology generation.

Fig. 11 provides a comparison of the computation delay of our approach with other state-of-the-art works. Apart from our proposed method, PUF operations are also used in [32]. Hash operations take the major portion of time computation of the protocols, constituting 83.75%, 72.6% and 65% of the time in [28], [29] and [31] respectively. In contrast, the proposed protocol employs hash operations only for attestation. 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 use the AES encryption scheme (taking $61.6\mu s$ for one execution). We also use a pseudo-random function for generating nonce (128 bits) to provide freshness, rather than message exchanges, resulting in speeding up of the protocol by $20.3\mu s$ and $40.6\mu s$ as compared to [28] and [31] respectively. We have also used PRNG operation for nonce generation and memory attestation. Only [30] uses multiplication operations ($60\mu s$) instead of XOR or PRNG. More details of the timing

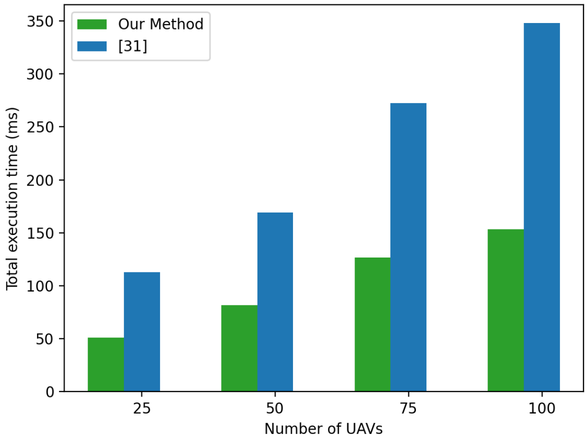

Fig. 12: Comparison of Total execution Time

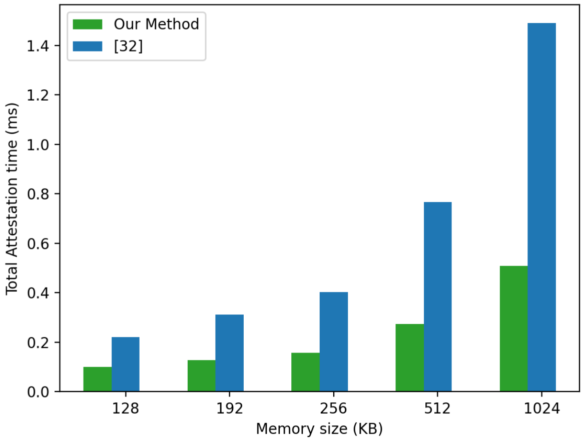

Fig. 13: Comparison of Total Attestation Time

for each operation execution are presented in Table I.

The total computation time of SHOTS is the lowest among the above-mentioned schemes ($1.96E-04$ vs $3.94E-04$, $3.55E-04$, $5.25E-04$, $2.65E-04$, and $3.54E-04$ seconds respectively). Please note that [30] has the lowest computation time among the other works, and therefore we choose this work to compare the total execution time by varying the number of UAVs in Fig 12. [31] is the only work which uses attestation, hence only [31] has been used for total attestation time comparison in Fig. 13.

Fig. 12 provides a comparison of the total execution time of the proposed protocol with [31] by varying the number of UAVs. [31] has the least computation cost in comparison to other state-of-the-art protocols. In Fig. 12, we provide a comparison of performance as the number of UAVs scales up. From the figure, we can observe that our protocol took $50.95ms$ for 25 UAVs and $153.37ms$ for 100 UAVs. Whereas, [31] took $113.025ms$ for 25 UAVs and $347.80ms$ for 100 UAVs. As the number of UAVs increases, the improvement in the performance of our protocol with respect to [31] increases.

Fig. 13 provides a comparison of the total attestation time of the proposed protocol with [32] by varying the size of memory that is attested. As more regions of memory are attested, the computation time increases. In Fig. 13, we observe that our protocol took $50.1ms$ for attesting 128 KB memory and $0.507ms$ for attesting 1 MB of memory. Whereas, [32] took $0.222ms$ for attesting 128 KB and $31.49ms$ for attesting 1024 KB. As attestation is, the improvement in the performance of our protocol with respect to [32] increases. It can also be seen that the proposed scheme is quite scalable, even for higher memory sizes up to 1024KB.

VIII. Conclusion

In this paper, we present a scalable protocol for mutual authentication cum attestation in UAV swarm networks. UAV communication is highly susceptible to security threats because it relies on wireless communication technologies. Therefore there is a need for establishing trust in communication with the base station that is lightweight, robust, and secure enough to mitigate these threats. This work used a Christofides algorithm-based optimal trajectory to achieve scalability by determining an optimal message flow path. The proposed protocol ensures physical security using Physical Unclonable Functions and is also resistant to man-in-the-middle attacks, replay attacks, and physical attacks. Its security was verified using the formal Mao Boyd logic proof. Our model overcomes challenges in computation cost and performance compared to other state-of-the-art protocols for UAV swarm networks.

IX. Acknowledgement

This work was supported by ASEAN - India Collaborative RD scheme (ASEAN-India ST Development Fund (AISTDF) sponsored) received by Dr. Vinay Chamola and Prof. Biplab Sikdar under Project Grant File no. CRD/2020/000369.

References

[1] T. Alladi, G. Bansal, V. Chamola, M. Guizani *et al.*, “Secauthuav: A novel authentication scheme for uav-ground station and uav-uav communication,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 068–15 077, 2020.

[2] 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.

[3] M. N. Aman, U. Javaid, and B. Sikdar, “A privacy-preserving and scalable authentication protocol for the internet of vehicles,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1123–1139, 2020.

[4] 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.

[5] M. Burrows, M. Abadi, and R. M. Needham, “A logic of authentication,” *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[6] E. Brickell, J. Camenisch, and L. Chen, “Direct anonymous attestation,” in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 132–145.

[7] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O’Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, “Principles of remote attestation,” *International Journal of Information Security*, vol. 10, no. 2, pp. 63–81, 2011.

[8] Y. Zeng, S. Jin, Q. Wu, and F. Gao, “Network-connected uav communications,” *China Communications*, vol. 15, no. 5, pp. iii–v, 2018.

[9] 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.

[10] 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.

[11] 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.

[12] C. Jiang, Y. Fang, P. Zhao, and J. Panneerselvam, “Intelligent uav identity authentication and safety supervision based on behavior modeling and prediction,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6652–6662, 2020.

[13] M. Yahuza, M. Y. I. Idris, A. W. A. Wahab, T. Nandy, I. B. Ahmedy, and R. Ramli, “An edge assisted secure lightweight authentication technique for safe communication on the internet of drones network,” *IEEE Access*, vol. 9, pp. 31 420–31 440, 2021.

[14] 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.

[15] T. Alladi, V. Chamola, N. Kumar *et al.*, “Parth: A two-stage lightweight mutual authentication protocol for uav surveillance networks,” *Computer Communications*, vol. 160, pp. 81–90, 2020.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, “A puf-based secure communication protocol for iot,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–25, 2017.

[21] 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.

[22] C. Pu and Y. Li, “Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system,” in *2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN*. IEEE, 2020, pp. 1–6.

[23] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, Tech. Rep., 1976.

[24] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[25] C. Hierholzer, “Euler, mei-ko kwan, königsberg, and a chinese postman.”

[26] S. Wøhlk and G. Laporte, “Computational comparison of several greedy algorithms for the minimum cost perfect matching problem on large graphs,” *Computers & Operations Research*, vol. 87, pp. 107–113, 2017.

[27] 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/

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] T. Alladi, S. Chakravarty, V. Chamola, and M. Guizani, “A lightweight authentication and attestation scheme for in-transit vehicles in iov scenario,” *IEEE Transactions on Vehicular Technology*, 2020.