Scalable Topologies for Time-Optimal Authentication of UAV Swarms

Gaurang Bansal, *Member, IEEE,* Vinay Chamola, *Senior Member, IEEE,*
Nirwan Ansari, *Fellow, IEEE* and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Swarm-based Unmanned Aerial Vehicle (UAV) applications require a large number of UAVs to be deployed across a region to work cooperatively. To operate a large number of unattended UAVs in hostile environments, it is critical to secure UAV-BS (base station) communications. UAV authentication based on Physical Unclonable Functions (PUFs) has recently emerged as a potential solution for overcoming adversarial attacks. The performance of PUF-based authentication protocols is strongly influenced by various factors, including the time required to generate the topology, the number of bottleneck connections, and the network's traffic load. This article investigates how the authentication time for a UAV swarm is affected by various factors such as the type of topology, number of UAVs in the swarm and the number of parallel connections.**

***Index Terms*—UAVs, authentication, topologies, performance analysis**

I. Motivation

Unmanned Aerial Vehicles (UAVs) are aerial devices that have become increasingly popular and have a wide range of applications. Although there has been rapid development in UAV-based technologies and applications, their deployment has not achieved its full potential due to several security challenges. Bringing UAVs closer to the users exposes them to increased threats and vulnerabilities, thus compromising the security of the UAV deployments [1]. Moreover, UAVs rely on wireless channels for communications, which can easily be intercepted and attacked. These attacks may disrupt the proper operation of UAVs, thus resulting in economic and societal losses..

Authentication is the forefront of security measures in mitigating these attacks. As shown in Fig. 1, UAVs, via authentication, are verified from time to time by a trusted source such as the base station (BS). As UAVs move from one location to another during their operations, the state (e.g., the state of their links, the base station serving them.) changes over time.

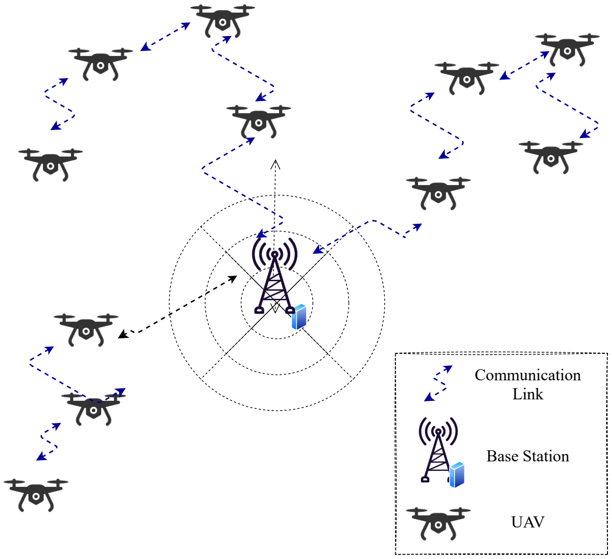


Fig. 1: System Model.

Thus, continuous authentication of devices is necessary to prevent a malicious adversary from derailing the resources and information related to the UAV application.

In recent years, there has been considerable research work in securing UAVs [2]. However, these proposed authentication techniques are inefficient due to high computation requirement, complexity, and mobility issues. [3]. Recently, physical unclonable functions (PUFs) have been widely used for authentication protocols [4]. PUF-based authentication protocols have been proven to provide better security features and computationally lightweight authentication in UAV scenarios. PUFs exploit the inherent randomness that is unique to a device and cannot be cloned or forged. This intrinsic randomness is generated during the the fabrication of chips used in the device [4]. A PUF can be modeled as a challenge-response system, where the PUF uses its internal characteristics to map a challenge C to response R. Scalability is an important required feature of an authentication protocol when it comes to the task of authenticating a swarm of UAVs. Such authentication has to be completed with minimal delay. In the past, researchers have proposed scalable protocols [5, 6]

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

Vinay Chamola is with the Department of Electrical and Electronics Engineering & APPCAIR, BITS-Pilani, Pilani Campus, 333031, India. (e-mail: vinay.chamola@pilani.bits-pilani.ac.in).Vinay Chamola is also with APPCAIR, BITS-Pilani, Pilani Campus, 333031, India.

Nirwan Ansari is with NJIT, USA (email: nirwan.ansari@njit.edu)

by varying network topologies to improve the performance of simultaneous authentication of multiple devices. Scalable protocols vary depending on how UAVs are deployed, the nature of authentication, and spatial location. Some performance metrics such as nature of authentication, and spatial location have been analyzed [7, 8] that have analysed some of the metrics. However, to the best of our knowledge, the performance analysis for various network topologies under different parameter such as number of UAVs in the swarms, number of parallel connections, and type of topology has not been studied previously.

This work provides a comparative performance analysis among three network topologies employed for a scalable PUF-based authentication protocol. The performance analysis evaluates various parameters related to topologies such as time for topology generation, number of connections, and total authentication time. The major contributions of this work are summarized below:

1) This work presents an analysis on how important network parameters for operating a UAV swarm is affected by various factors like the number of UAVs in the swarm, number of parallel connections, and type of topology. Considered network parameters include the time required to generate the topology, the number of bottleneck connections, and the authentication time.
2) This work presents the performance analysis of three realistic swarm topologies, i.e., K-means clustering topology, Spanning Tree topology, and Christofides topology. The analysis can guide network administrators to tune their network parameters as per their specific use cases.
3) A realistic scenario, where Raspberry Pi 3B is used to model the onboard computer on the UAV is simulated. The communication time for the UAVs and the base station were evaluated on Mac OS (1.8 GHz Dual-Core Intel Core i5, 8 GB 1600 MHzDDR3).

II. System Model

This paper considers the scenario of UAV-base station communications (shown in Fig. 1). The system model consists of multiple UAVs connected to a single base station. The base station is considered stationary and trusted. UAVs are heterogeneous computing devices and have different levels of storage, memory, computing and security. UAVs are enabled with PUF and are deployed for various operations, but they are vulnerable to attacks. Each UAV is located at a particular coordinate in the 3-dimensional space at a given instant of time and broadcasts its location to the base station with its (x, y, z) coordinate information, i.e., the relative location of UAV with respect to its serving base station.

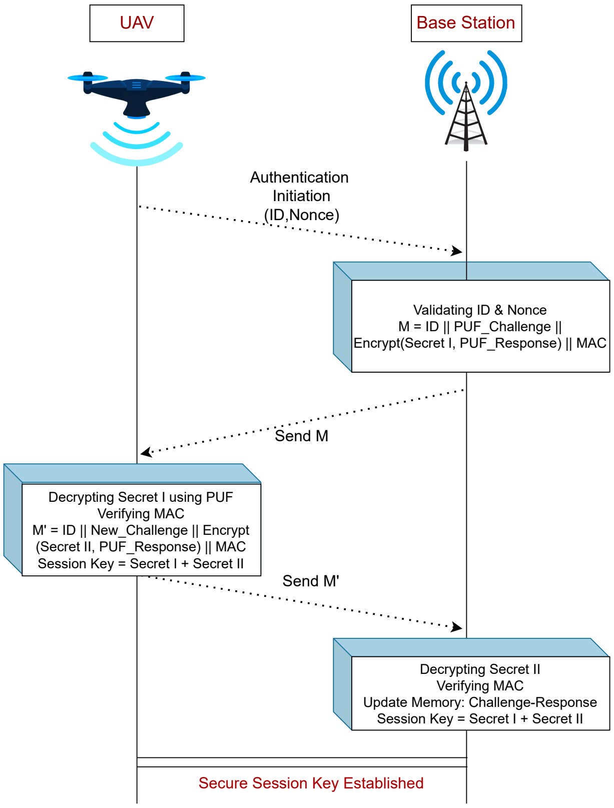

UAV
Base Station
Authentication Initiation (ID,Nonce)
Validating ID & Nonce
M = ID || PUF_Challenge || Encrypt(Secret I, PUF_Response) || MAC
Send M
Decrypting Secret I using PUF
Verifying MAC
M' = ID || New_Challenge || Encrypt (Secret II, PUF_Response) || MAC
Session Key = Secret I + Secret II
Send M'
Decrypting Secret II
Verifying MAC
Update Memory: Challenge-Response
Session Key = Secret I + Secret II
Secure Session Key Established

Fig. 2: Abstraction of mutual authentication protocol using PUFs.

The base station and UAVs mutually verify each other's identity through an authentication protocol. The system model employs a PUF-based mutual authentication protocol inspired from [4]. The mutual authentication protocol is presented in Fig. 2. The UAV sends its ID, GPS location and nonce to the base station. Once the base station receives the message from the UAV, it sends a challenge and a secret encrypted message to authenticate the UAV. The secret encrypted message can be decrypted only by the valid UAV by using its PUF. Once the UAV decrypts the secret message, it sends its secret encrypted message to the BS. Only a valid BS can decrypt the message successfully, as BS is the only entity besides the device that knows the PUF response. Once both parties have verified each other, they agree upon a session key using a combination of secrets from the BS and the UAV.

To improve scalability, it is important to decrease the number of communication exchanges during the authentication process, and also to authenticate multiple devices at once. This is possible when messages are communicated in a hop-by-hop manner. The work considers three topologies (K-means clustering topology, Christofides topology, and Spanning Tree topology) that can be employed in different scenarios to authenticate a swarm of UAVs in the coming sections. These three topologies are chosen since any UAV swarm design can be categorized into one of these topologies [9]. These topologies are present in realistic scenarios of UAV swarms.

III. K-means Clustering Topology

In this section, the base station creates clusters based on the UAVs' locations using the K-means clustering algorithm [10]. For example, for K=3, the protocol employs the 3-means clustering algorithm to partition all the UAVs into 3 clusters (as shown in Fig. 3). These clusters correspond to aggregating UAVs into subgroups based on their current geographic locations. The clustering algorithm starts by randomly choosing coordinates of K UAVs. These initial points form the initial centroids of K clusters.

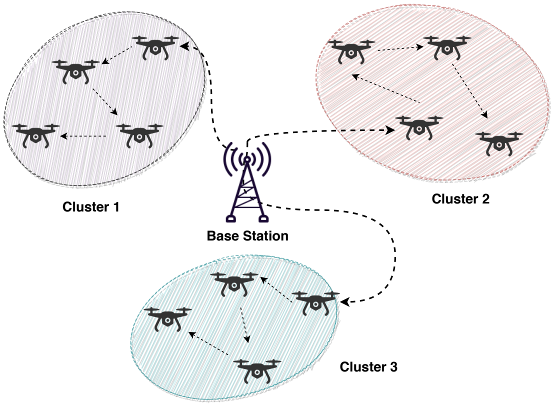


Fig. 3: K-means clustering topology.

In each iteration, the distances of each UAV from the K centroids are evaluated. After the distance evaluation, each UAV is assigned to the nearest cluster. Then, each cluster's centroid coordinates are updated by taking the mean of UAVs' coordinates belonging to the cluster. The update process continues until all distances remain constant from the center-points and the centers are not updated anymore.

Once clusters are fixed, the nearest UAV to the base station is identified. The BS generates a path flow covering all the UAVs in their corresponding clusters. This path is decided by selecting UAVs in the cluster at random. The communication from the BS to UAVs happens in a hop-by-hop manner. The BS sends a message to its closest UAV in a cluster. This UAV (closest to BS) forwards the communication to another UAV in the cluster and so on. Thus, for K clusters, K paths are formed. Each of the paths starts from the base station and covers all the UAVs in its corresponding cluster. In each path, the messages flow from the parent to the child, originating from the base station. Here, the parent refers to the transmitting UAV, and the child refers to the receiving UAV. The resulting topology is referred to as the K-means clustering topology.

IV. Christofides Topology

This section delineates how the Christofides algorithm [11] is employed to determine the Christofides topology. The heuristics used for optimizing the message flow path leverage the triangle inequality. Once the base station knows each UAV's location, it creates an optimal path by creating a minimum spanning tree (shown in Fig. 4). The problem of creating an optimal path can be considered as 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 generating the Christofides topology are explained in the following subsections.

A. *Creation of minimum spanning tree*

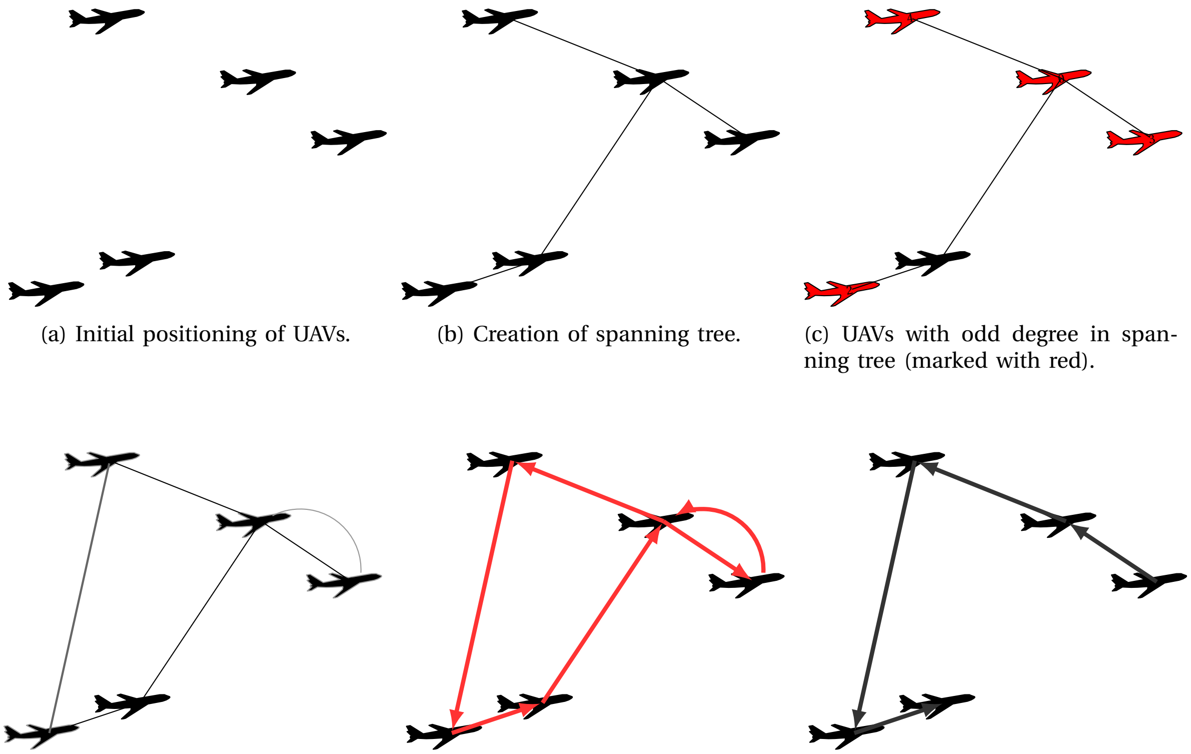
A minimum spanning tree of a graph is a subset of the graph containing all the vertices with the minimum number of edges. The Kruskal algorithm [12] uses the greedy approach to find the minimum spanning tree. The resultant spanning tree has the number of edges one less than the number of vertices or nodes (shown in Fig. 4(b)). A vertex with an odd number of edges incident on it is called an odd degree vertex. In the next subsection, the protocol finds odd degree vertices.

B. *Finding vertices with an odd degree*

After creating the spanning tree, the resultant graph covers all the vertices and ensures that there is no cycle. To convert a spanning tree into an Eulerian tour, the protocol finds vertices with odd degrees in linear search as shown in Fig. 4(c). Then, these odd degree vertices are converted to even degrees by forming the perfect matching as described in the following subsection. The need to convert the degree of vertices to even is to find an Eulerian path. Carl Hierholzer, in 1873, proved that all vertices must have even degrees for the Eulerian circuit to exist. An elaborate description of Eulerian circuit is described in Section IV-D.

C. *Perfect matching and handshaking lemma*

Figure 4(c) shows a minimum spanning tree (denoted by T), where odd degree vertices are marked red and even degree vertices are black. All the odd degree vertices are connected to obtain a subgraph. A perfect matching [13] covers 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. While there are many possibilities of



(a) Initial positioning of UAVs.

(b) Creation of spanning tree.

(c) UAVs with odd degree in spanning tree (marked with red).

(d) Find a perfect minimal matching on odd vertices.

(e) Formation of Euler circuit.

(f) Final Hamiltonian path by skipping visited nodes.

Fig. 4: Creation of Cristofides topology.

perfect matching, Fig. 4(d) consider one such case. Let this perfect matching be represented by M. Next, the protocol use the union function 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 [14]. In performing perfect matching on a graph formed by odd vertices, each vertex forms part of only one edge. So, the union of T and M ensures that the degree of every odd vertex is increased by one. Thus, all odd degree vertices will eventually become even degree vertices.

D. Eulerian tour

With even degree at all vertices the protocol now creates an Eulerian circuit or Eulerian cycle. An Euler circuit constructs a path from the initial vertex and visits all the edges exactly once. The protocol chooses the closest UAV to the BS as the starting vertex. The current vertex is removed and pushed to the stack. Next, the closest neighbor is chosen and remove the edge between the current vertex and the closest neighbor is removed. Now, the closest neighbor becomes the current vertex. This process is repeated until there is no neighbor remaining. The current vertex is then added to the path, and the vertex from the stack is popped and made the current vertex. This process is repeated until the stack is empty. Hence, the resultant graph gives the Eulerian circuit shown in Fig 4(e).

E. Hamiltonian path

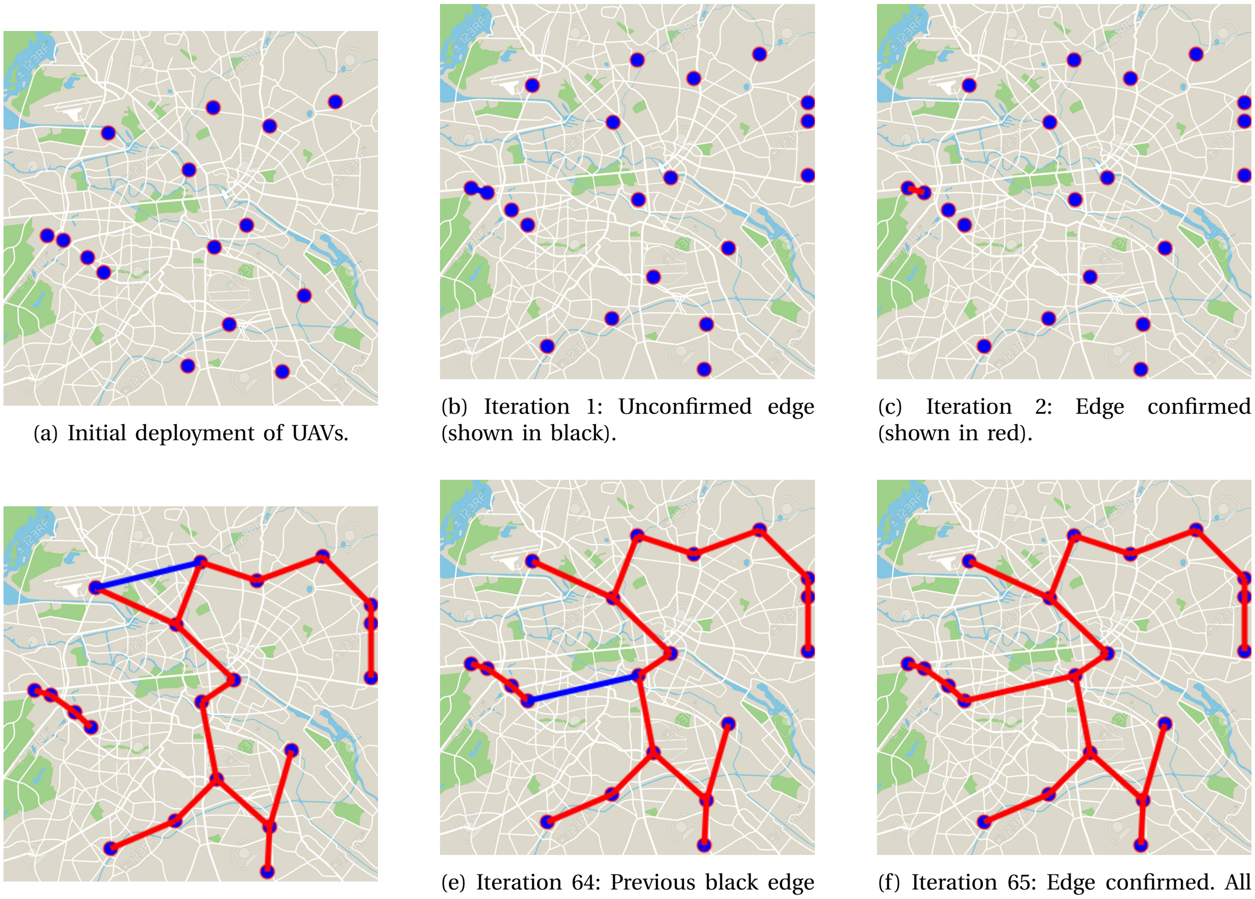
Finally, while moving along the Euler circuit, the algorithm checks if a node has been visited or not. If it has not been visited, it is added to the Hamiltonian path as depicted in Fig. 4(f). Else, the node is skipped and the algorithm moves on. This skipping will not increase the length of the path as the graph satisfies the triangle inequality. This resulting Hamiltonian path is also referred to as the Cristofides topology.

V. Kruskal Spanning Tree Topology

This section discusses the generation of minimum spanning tree using the Kruskal algorithm [12]. As discussed earlier, a minimum spanning tree of a graph is commonly referred to as a subset of the graph containing all the vertices with the minimum number of edges without any cycles.

The Kruskal algorithm's output provides a list of routes containing paths starting from the base station covering all the devices. A route is defined as a finite sequence of edges that traverse (join) a sequence of devices and terminate at a node (with degree one).

Figure 5(a) depicts the initial stage when the system is deployed, but the topology execution has not started



(a) Initial deployment of UAVs.

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

(c) Iteration 2: Edge confirmed (shown in red).

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

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

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

Fig. 5: Different iterations of the execution of the Kruskal algorithm. black dots are UAVs that are deployed in a region. Red line segments represent the selected and confirmed edges of the resulting spanning tree. black line segments represent probable spanning tree edges which can be rejected from or accepted into joining the spanning tree based on the Kruskal algorithm.

Number of UAVs	Topology	Topology Generation Time	#connections	Communication Time	Authentication Time (Total)	#connections	Communication Time	Authentication Time (Total)
25	Kruskal Spanning Tree	0.66	**3**	30.10	34.78	**1**	33.86	**38.55**
	Christofides Topology	1.54		44.61	50.18		44.61	50.18
	K-means clustering topology	13.25		14.10	**31.38**		24.00	41.28
	One to One Topology	0.0029		35.30	39.33		106.40	110.43
50	Kruskal Spanning Tree	0.56	**3**	46.20	54.81	**1**	57.75	**66.36**
	Christofides Topology	5.89		66.40	80.34		66.40	80.34
	K-means clustering topology	12.49		24.30	**44.84**		47.60	68.14
	One to One Topology	0.0031		61.50	69.55		155.90	163.95
75	Kruskal Spanning Tree	0.76	**3**	54.20	**67.04**	**1**	108.40	121.24
	Christofides Topology	29.55		73.70	115.32		73.70	**115.32**
	K-means clustering topology	14.12		44.60	70.79		95.40	121.59
	One to One Topology	0.01		92.40	104.48		252.70	264.78
100	Kruskal Spanning Tree	1.12	**3**	56.58	**73.80**	**1**	169.75	186.97
	Christofides Topology	24.04		104.62	144.76		104.62	**144.76**
	K-means clustering topology	14.30		54.50	84.90		127.60	158.00
	One to One Topology	0.0139		94.20	110.30		321.30	337.40

TABLE I: Performance comparison of different topologies under different UAV network scenarios. All timings are in ms (millisecond). Topology generation time is time taken into joining the topology. Communication time is the time spent in propagating authentication messages.

yet. UAVs are represented as vertices; these vertices, together with edges connected among the vertices, constitute a graph. The Kruskal algorithm randomly creates an edge between the initial vertex and other vertices shown as the black edge in Fig. 5(b). Then, the algorithm checks: if there already exists an alternative path between these vertices in Fig. 5 (b), the black edge is confirmed (red edge in Fig. 5(c)); else, the black edge is dropped, and a new edge is chosen. This process is repeated until a minimum spanning tree is formed. Fig. 5(d) describes the case when the black line or trial edge fails to be confirmed since an alternative path by which vertices of the edge are connected. Then, the algorithm chooses another random edge (shown in Fig. 5 (e)), which gets confirmed in Fig. 5 (f), marking the confirmation of the edge (shown in red), thus completing the construction of the minimum spanning tree.

VI. One-One topology

One-to-one authentication is a benchmark where a particular device authenticates directly with the base station as used in [4]. In scenarios where the base station supports parallel connections, the maximum number of devices that can simultaneously be authenticated is equal to the number of parallel connections. This technique does not use any topology to improve the scalability of authentication. Next, we provide a comparison of different topologies presented in Section III, IV, and V with one-to-one authentication.

VII. Results and Discussion

This section compares the different topologies in authentication protocol for UAV networks. The mutual authentication protocol considered for all the four topologies is the same, and has been explained in the system model (also depicted in Fig. 2). UAV operations were performed on a Raspberry Pi 3B. The base station and communication time were simulated on Mac OS (1.8 GHz Dual-Core Intel Core i5, 8 GB 1600 MHz DDR3). The processes were coded in the Python programming language.

Table 1 presents the comparison of the total authentication time taken by different topologies in different UAV scenarios. The timing for the protocol execution is evaluated for four topologies:

- K-means clustering topology
- Christofides Topology
- Kruskal Spanning Tree Topology
- One-to-One authentication

The time for executing the authentication protocol, referred to as the authentication time, includes the topology generation time, message propagation time, and time spent in computation at devices. The computation time for authentication is 0.161 ms per device for all three topologies. The number of connections refer to the number of UAVs that can simultaneously receive and send messages from and to the BS. During the authentication procedure, if the number of connections is 3, then the BS can simultaneously send authentication messages to 3 UAVs, thus improving authentication performance.

A. Case I: BS can communicate with 3 UAVs simultaneously

By inspecting different scenarios, it can be observed that when the number of UAVs is small (25 or 50), K-means clustering topology incurs the lowest total authentication time. The performance of the K-means clustering topology is attributed to low communication time. Here, communication time refers to the time spent in propagating authentication messages. Although the time taken to generate clusters using the K-means algorithm is more than other topologies, its performance is still dictated by the small propagation delay.

As the number of UAVs exceeds 75, the Kruskal spanning tree topology outperforms others. The K-means clustering topology incurs considerable increase in the propagation delay. In contrast, the Christofides topology incurs the longest authentication time, primarily because of the high complexity of generating the Christofides topology. Moreover, it does not utilize parallel connections. The performance of one-to-one authentication worsens as the number of UAVs increases.

B. Case II: BS can communicate with only 1 UAV

In the scenarios where the number of UAVs is less than 75, the Kruskal topology outperforms the K-means clustering topology because the K-means clustering topology cannot leverage parallel connections, thus incurring high communication time.

As the number of UAVs increase (75 or more), the total authentication time is primarily dominated by the communication time. Among all the topologies, the Christofides topology achieves the least propagation delay and hence it outperforms other techniques.

As the UAV network scales up with more complex topologies, multiple spanning trees [15] may be leveraged to establish a reliable and efficient authentication topology.

VIII. Future Works

To support future advancements in this area of study, numerous factors associated with UAV swarms such as the energy charge remaining in the UAVs, mobility pattern of the UAVs, and energy efficiency of the UAVs will be considered. Future works may also focus on quantum-based cryptography, which aims to replace public-key cryptosystems. Additionally, the challenge-response pair behavior of various kinds of

PUFs under various climatic and physical situations should be carefully explored to reduce authentication errors. Finally, small-scale, realistic experimental testbeds with various kinds of drones will be developed to confirm the theoretical conclusions obtained so far.

IX. Conclusion

Rising UAV deployments and increasing vulnerabilities in UAV communications call for lightweight authentication mechanisms. These authentication protocols can regularly check UAV status and verify if each UAV is secure or not. Physical Unclonable Function based protocols have been used to address this issue. PUF based protocols exploit the physical randomness in devices that is generated during fabrication. However, the performance of these PUF-based protocols depends on the topology of the network. This article provides a comparative analysis of the performance of different topology designs for their use in authentication protocols.

X. 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] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1594–1606, 2017.

[2] 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.

[3] D. He, S. Chan, and M. Guizani, "Communication Security of Unmanned Aerial Vehicles," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 134–139, 2016.

[4] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for v2g using physical unclonable function," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7234–7246, 2020.

[5] 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.

[6] 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.

[7] K. Gai, Y. Wu, L. Zhu, K.-K. R. Choo, and B. Xiao, "Blockchain-enabled trustworthy group communications in uav networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4118–4130, 2021.

[8] C. F. E. de Melo, T. Dapper e Silva, F. Boeira, J. M. Stocchero, A. Vinel, M. Asplund, and E. P. de Freitas, "Uavouch: A secure identity and location validation scheme for uav-networks," *IEEE Access*, vol. 9, pp. 82 930–82 946, 2021.

[9] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles with switching interaction topologies," *Control Engineering Practice*, vol. 46, pp. 26–36, 2016.

[10] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[11] N. Christofides, A. Mingozzi, and P. Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations," *Mathematical programming*, vol. 20, no. 1, pp. 255–282, 1981.

[12] 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.

[13] 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.

[14] C. Vasudev, *Graph theory with applications*. New Age International, 2006.

[15] N. Ansari, G. Cheng, and R. N. Krishnan, "Efficient and reliable link state information dissemination," *IEEE Communications Letters*, vol. 8, no. 5, pp. 317–319, 2004.

Gaurang Bansal is doctoral student and recipient of the President Graduate Fellowship at the National University of Singapore (NUS) under Prof. Biplab Sikdar at Department of Electrical and Computer Engineering. Previously, he had completed his Master's and Bachelor's from BITS Pilani in 2020 2018, respectively. Currently, he is also serving as Web Editor for ACM XRDS Magazine. His research interests include wireless IoT, network security, queuing theory, blockchain and machine learning. He also serves as a member of Eta Kappa Nu and Internet Engineering Task Force (IETF). He has also served as co-organiser and TPC chair for various reputed workshops like IEEE Globecom workshop and IEEE INFOCOM workshop.

Vinay Chamola is an Assistant Professor in the Electrical and Electronics Department, Birla Institute of Technology & Science (BITS), Pilani, India, and is also a part of APPCAIR, BITS-Pilani. He received his B.E. (2010) and M.E. (2013) degrees from BITS, Pilani and PhD (2016) from National University of Singapore (NUS), Singapore. His research interests include Internet of Things, 5G network provisioning, Blockchain and Security. He has over 100 publications in high ranked SCI Journals including more than 60 IEEE Transaction, Journal and Magazine articles. He is an Area Editor of Ad Hoc Networks, Elsevier and the IEEE Internet of Things Magazine. He also serves as Associate editor in various journals like IEEE Networking Letters, IET Networks, IET Quantum Communications etc.

Nirwan Ansari [S'78, M'83, SM'94, F'09] (nirwan.ansari@njit.edu), Distinguished Professor of Electrical and Computer Engineering at the New Jersey Institute of Technology (NJIT), received his Ph.D. from Purdue University, M.S.E.E. from the University of Michigan, and B.S.E.E. (summa cum laude with a perfect GPA) from NJIT. He is also a Fellow of the National Academy of Inventors. His current research focuses on green communications and networking, cloud computing, drone-assisted networking, and various aspects of broadband networks.

Biplab Sikdar received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He currently serves as an Associate Editor for the IEEE Internet of Things Journal. He has served as a TPC in various conferences such as IEEE LANMAN, GLOBECOM, BROADNETS and ICC to name a few. He is a member of Eta Kappa Nu and Tau Beta Pi.