


# Group Secret Key Generation Using Physical Layer Security for UAV Swarm Communications

**SOBIA JANGSHER** , Member, IEEE  
Khalifa University, Abu Dhabi, UAE  
Dublin City University, Dublin, Ireland

**ARAFAT AL-DWEIK** , Senior Member, IEEE  
Khalifa University, Abu Dhabi, UAE  
Western University, London, ON, Canada

**YOUSSEF IRAQI** , Senior Member, IEEE  
Mohammed VI Polytechnic University, Ben Guerir, Morocco

**ANSHUL PANDEY** , Member, IEEE  
**JEAN-PIERRE GIACALONE**, Member, IEEE  
Technology Innovation Institute, Abu Dhabi, UAE

**In unmanned aerial vehicle (UAV) swarm networks, a group secret key (GSK) is required to enable secured UAV-UAV communications, multicast, and broadcast transmission. Moreover, it can be used for device authentication. Therefore, this article proposes an efficient**

Manuscript received 3 May 2023; revised 25 July 2023; accepted 14 August 2023. Date of publication 22 August 2023; date of current version 8 December 2023.

DOI. No. 10.1109/TAES.2023.3307092

Refereeing of this contribution was handled by J. Choi.

This work was supported by Technology Innovation Institute (TII) Project “PHY Layer Security for Heterogeneous UAV-Ground Wireless Networks,” under Grant EX2020-036.

Authors’ addresses: Sobia Jangsher is with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, UAE, and also with the Dublin City University, Dublin, Ireland, E-mail: (sobia.jangsher@ku.ac.ae); Arafat Al-Dweik is with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, UAE, also with the Center for Cyber-Physical Systems, Khalifa University, Abu Dhabi 127788, UAE, and also with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada, E-mail: (arafat.dweik@ku.ac.ae); Youssef Iraqi is with the College of Computing, Mohammed VI Polytechnic University, Ben Guerir 43150, Morocco, E-mail: (youssef.iraqi@um6p.ma); Anshul Pandey and Jean-Pierre Giacalone are with Secure Systems Research Center, Technology Innovation Institute, Abu Dhabi 9639, UAE, E-mail: (anshul@ssrc.tii.ae, jean-pierre@ssrc.tii.ae). (*Corresponding author: Arafat Al-Dweik.*)

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>

GSK protocol denoted as the sequential secret group key (SSGK) algorithm for distributed UAV swarm-secured communications. The proposed protocol utilizes network coding to generate cooperation information that is transmitted on the channel and is uncorrelated with the generated secret key. The proposed protocol depends on the pairwise key generation process between pairs of UAVs. However, not all possible pairwise agreements are performed to minimize the time and signaling overhead, i.e., the pairwise key agreement is performed only between selected UAVs as a compromise between the required overhead transmission resources and the achieved redundancy level. The obtained results show that a redundancy level of 4 is sufficient to provide a reliable GSK generation process. The results also show that the performance of the key generation process highly depends on the channel bit error rate (BER), the number of UAVs, and the key length.

## I. INTRODUCTION

### A. Overview

Unmanned aerial vehicle (UAV) mesh communication networks have gained interest for several applications, such as surveillance and security, remote sensing, and rescue missions [1], [2], [3]. Of particular interest is the case of multiple UAVs working collectively to accomplish a mission objective. Widely referred to as UAV swarms, this group of UAVs can find their applicability in a variety of use cases that a single UAV cannot accomplish [4]. However, UAV swarm communications are vulnerable to various security attacks due to their distributed and cooperative nature. Any security attack can be fatal as it can fail the ongoing mission, perform some mischievous tasks, and even lead to crashing or seizing the UAVs. UAV swarms have recently started a new era of applications in the defense industry. For example, the U.S. military recently launched a huge project to use autonomous UAV swarms on a massive scale to defend against various types of attacks [5]. In such applications, a certain command might need to be shared securely among all members of the swarm. However, direct communications between the central control center and all UAVs might be infeasible, and hence, distributed key sharing will be indispensable. Therefore, data communicated between the swarm members over the mesh should be secured, and the members’ authenticity must be continuously validated [6]. To this end, a group secret key (GSK) can be a promising solution to ensure secure communications and device authentication.

Traditional cryptography-based infrastructure can generate a GSK within the UAV swarm mesh networks. However, UAV swarm mesh networks, usually characterized as infrastructureless, distributed, and dynamic, can challenge a practical implementation of such cryptography-based security solutions. There is also some work on infrastructureless authentication and key agreement schemes for vehicular networks [7], [8] as well as UAV [9], [10], [11]. Vehicular networks have some work related to group key generation (GKG), but in the UAV network domain, most of the work is on authentication and key agreement between two UAVs. Recently, physical layer security (PLS) based key generation has received considerable attention over the past decade due to its suitability for distributed and infrastructureless wireless networks [12], [13], [14], [15],

[16]. The main advantage of physical layer security (PLS) is that it can allow users to share a secret random key based on the unique characteristics of the channel between them. If the channel varies frequently over time, which is usually the case for several wireless applications, users can update the shared secret key with reasonable communication overhead and reduced computational complexity. Consequently, system security can be improved because varying the key frequently enables approaching Shannon's perfect secrecy condition [17]. Nevertheless, the air-to-air (A2A) channel entropy for UAV communications can be low, which can lead to a low key generation rate. To overcome such problems, Assaf et al. [18] proposed using physically unclonable function (PUF) where the channel coefficients are used as the challenge for the PUF and PUF emulator (PUFe). Although the reliability of PUFs can deteriorate when experiencing a wide variation in temperature or supply voltage variations, new research results show promising results. For example, the 28 nm CMOS PUF proposed in [19] demonstrated robust performance for temperatures in the range of  $-40$  to  $100$  °C and voltages of  $0.5$ – $1.4$  V.

However, PLS-based key generation is primarily suitable for point-to-point communications because PLS relies on the characteristics of the shared channel randomness and the underlying channel reciprocity (CR) principle between pairs of users. On the contrary, GSK is envisioned to ensure a secure information transfer, irrespective of the mode, whether it is broadcast, multicast, point-to-point, or relay-based communications. Beyond fast information transfer, the GSK can further find its application for device authentication in a network [20]. For instance, the GSK can be used as a secret token for continuous authentication. Toward this, some practical usages of GSK are for ensuring information security in platoon-based vehicular and UAV swarm networks. Unlike peer-to-peer communications, no common wireless channel exists between all UAVs in a mesh network based on which a group key (GK) can be generated. More specifically, in the presence of  $M$  UAVs, there will be  $M(M - 1)/2$  different channels associated with them. Consequently, direct key generation and distribution processes based on the mutual channel between the pairs of UAVs are practically infeasible.

## B. Motivation and Contribution

Unlike the work reported in the literature, this work proposes a novel algorithm for distributed GSK generation with application to UAV swarm mesh communications. The proposed algorithm is denoted as the sequential secret GK (SSGK). The main contributions of this article are as follows.

- 1) The SSGK couples the randomness of the wireless channels and PUFs to generate and share a GSK for UAV swarm communications.
- 2) Due to the complexity encountered by deploying a PUFe in each UAV, an efficient group partitioning is considered where each UAV is equipped with a small subset of emulators. The subset size is a design parameter that is selected based on the desired

complexity and redundancy levels. However, it is typically much smaller than the total number of UAVs in the network.

- 3) Network coding with a leader selection process is adopted to ensure secure and robust key generation for all members of the swarm. The algorithm depends on the pairwise key generation (PKG). However, it is not required to generate pairwise keys between all pairs of UAVs [21].
- 4) The UAVs partitioning provides an additional degree of freedom to tradeoff complexity with connectivity. The minimum complexity and lowest probability of connectivity are obtained when each UAV is equipped with only one PUFe. In this case, each UAV can communicate only with another predefined UAV. The other extreme, that is, the maximum complexity and probability of connectivity, is obtained when each UAV is loaded with the PUFe of all other UAVs in the swarm. The UAVs partitioning offers a flexible compromise between the two extremes. The selection of system parameters depends on the available resources and performance requirements. As compared with the single PUFe case, having multiple PUFs in each UAV offers each UAV the choice to communicate with other multiple UAVs, i.e., it creates connection redundancy.
- 5) The system performance is evaluated in terms of the probability of GK disagreement, the average number of UAVs that managed to share the GK, and system complexity. The results are presented for various key lengths and redundancy orders.
- 6) The performance of the proposed SSGK algorithm is compared with the state-of-the-art, and the results obtained confirm its efficiency.

Based on the extensive literature search and to the best of the authors' knowledge, there is very little work that proposes GSK generation using PUFs, which are used to generate the pairwise keys between certain UAV pairs efficiently and securely. The main advantage of PUFs is their ability to operate efficiently in near-flat fading channels, which is not the case for most other existing algorithms [18]. However, configuring all UAVs with all PUFes can be prohibitively complex. Therefore, we propose set partitioning to reduce complexity with negligible performance degradation. Another crucial feature is that no central UAV is required in the proposed scheme; therefore, the GK can be generated in a distributed manner. Moreover, redundancy is exploited to improve the probability of successfully sharing the GK. Such a performance improvement is proportional to the redundancy order, as illustrated later in the numerical results.

## C. Article Organization

The article is organized as shown in Fig. 1.

## D. Notations

Table I presents the notations used throughout the article.

<b>I. Introduction</b>	<b>IV. Proposed SSGK Protocol</b>	<b>V. Performance</b>
A. Overview	A. Protocol Terminologies	A. Secrecy
B. Mot. & Contrib.	B. Protocol Overview	B. Complexity
C. Organization	C. Protocol Procedure	C. Overhead
D. Notations	1 Initialization	<b>VI. Num. Results</b>
<b>II. Related Work</b>	2 Contacted UAV Selection	<b>VII. Conclusion</b>
<b>III. System Model</b>	3 Pairwise Key Generation	<b>VIII. Appendices</b>
A. Pairwise Keys	4 Cooperative Information	A. Acronyms
B. Set Partitioning	5 Key Extraction Process	B. Symbols
	6 Group Key ID Update	C. References
	7 No Redundancy	
	D. Verification & Disconnect.	

Fig. 1. Article organization.

TABLE I  
Notation Used Throughout the Article

Description	Definition	Example
Blackboard symbols	Sets	$\mathbb{X}$
Overlined blackboard symbols	Subsets	$\overline{\mathbb{X}}$
Bold letters	Vectors	$\mathbf{x}, \mathbf{X}, \mathcal{X}$
Upper/lower case letters	Scalars	$k, K$
Calligraphic $K$	Secret keys	$\mathcal{K}$

## II. RELATED WORK

There are several studies on authentication and key agreement for group communication in drone communication using different approaches, such as classical cryptographic [7], [22] based approaches, PLS, and blockchain [23]. Table II presents a summary of the relevant literature on authentication and key sharing using different techniques. The studied protocols have been proposed for UAV network with centralized and distributed approaches.

The work on GSK generation using PLS can be broadly classified into two categories: GSK based on PKG [21], [24], [25], [26]; and GSK based on shared common randomness [24], [27], [28]. Most of the work belongs to the first category as it can utilize pairwise key-generated schemes. GSK based on pairwise keys is generated by creating all possible pairwise keys between all UAVs. However, generating a key between all pairs and sharing the keys between all UAVs require significant transmission resources, particularly for a large dynamic network. Xu et al. [21] propose an efficient algorithm for GSK generation for a three-UAV network, multinode ring network, and multinode mesh network. All pairwise keys are generated and divided into small segments to generate a one-time pad sequence. The algorithm depends on all generated pairwise keys, significantly increasing the signaling overhead group users. In [25], the channel between the central UAV and the reference UAV is considered as the common secret. The GK is leaked if the link between the two UAVs is compromised. A UAV addition is impossible without increasing the number of broadcasts on the network. Peng et al. [26] proposed a GKG for self-organizing networks based on the PKG of the legitimate users with two neighbors. However, the proposed scheme is not robust because the key cannot be exchanged

with the entire group if any link is lost. The difference of signal strength (DOSS) is used in [24] as the common secret. Jagadeesh et al. [27] have designed a consensus algorithm for a three-UAV network called the entropy-maximization error-minimization algorithm to maximize the entropy of the secret key such that the mismatch rate is less than a certain bound. Furthermore, Thai et al. [28] have proposed a GSK generation framework over mesh networks, wherein each UAV is equipped with multiple antennas. Specifically, the scheme required every network UAV to estimate the channel with every other UAV and then perform postprocessing cooperatively to arrive at a shared common group secret. However, the schema is limited to one-hop mesh networks and may incur much overhead regarding channel estimation.

Furthermore, the inherited randomness associated with the wireless channel may be limited in specific scenarios, such as rural areas, aerial-to-ground station line-of-sight communications, and UAVs-to-UAVs communications. This may limit the generation of a truly random key over the air and even make it easier for any intruding UAVs flying in the near vicinity to clone such a secret key. This requires a complementary source of randomness, which is also almost unclonable. To this end, PUFs can be an interesting solution. PUFs are integrated circuits with unique and unclonable structures due to the fabrication conditions and process. PUFs are characterized by a set of channel-response pairs, that is, for each challenge input to PUF, there is a unique response obtained as the output of PUF. PUFs can be used to improve the characteristics of the pairwise keys generated using the conventional PLS techniques. In such configurations, PLS can be used to generate an intermediate key, which is fed to the PUF that generates the final key [18]. Any two UAVs can generate a pairwise key if one of them has a PUF and the other has the corresponding emulator [29], [30], [31], [32]. The concept of coupling the random wireless channels and the PUF function to generate GSK is yet to be explored in the literature.

## III. SYSTEM AND CHANNEL MODELS

This work considers a UAV swarm mesh network with  $M$  UAVs distributed uniformly in a given geographical area. The set of UAVs is represented by  $\mathbb{M} = \{U_1, U_2, \dots, U_M\}$ , and all UAVs must generate and share a common GSK. In this work, it is assumed that all UAVs are trusted and each UAV has at least one other UAV in its transmission range to initiate the PKG process. The assumption that all nodes are trusted can be justified by noting that using PUFs with PLS provides inherent authentication, as described in Section V-A. The transmission time is divided into frames, where each frame is divided into two time slots,  $T_{\text{GKG}}$  and  $T_{\text{DT}}$ , as shown in Fig. 2. The slot  $T_{\text{GKG}}$  is the period for the GKG and  $T_{\text{DT}}$  is the data transmission time,  $T_{\text{GKG}} \ll T_{\text{DT}}$ . The channel between  $U_i$  and  $U_j$  is denoted as  $\mathbf{h}_{i,j} = h_{i,j}^{(1)}, h_{i,j}^{(2)}, \dots, h_{i,j}^{(Q)}$ , where  $Q$  is the number of frequency slots or subcarriers used for communications [18].

TABLE II  
Tabulated Comparison of Selected State-of-the-Art Articles Versus the Proposed Scheme

Ref.	Key Sharing	Authentication	Distributed	PLS	CR	PUF	Complexity
[23]	✓	✓	—	—	—	—	mid
[22]	✓	✓	—	—	—	—	high
[8]	✓	✓	✓	—	—	—	low
[33]	✓	✓	—	✓	-	✓	low
[34]	✓	✓	✓	✓	-	✓	low
[35]	✓	✓	—	✓	-	✓	low
[21]	✓	—	—	✓	✓	-	low
[25]	✓	—	—	✓	✓	-	high
[26]	✓	—	✓	✓	✓	-	high
SSGK (proposed)	✓	✓	✓	✓	✓	✓	low

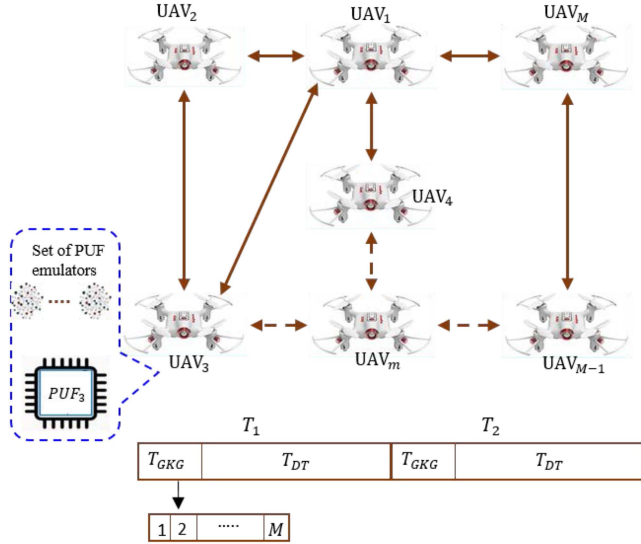


Fig. 2. System model for UAV swarm GSK generation, where  $T_j$  is the  $j$ th transmission frame,  $T_{GKG}$  and  $T_{DT}$  are the GKG and data transmission time slots, and  $T_{GKG,j,i}$  is the time subslot allocated for UAV  $i$  in frame  $j$ .

The channel is considered to be quasi-static, that is, the channel is fixed for at least  $T_C > T_{GKG}$  s, where  $T_C$  is the channel coherence time. Moreover, given that the channel is reciprocal, then  $\mathbf{h}_{i,j} = \mathbf{h}_{j,i} \forall \{U_i, U_j\} \in \mathbb{M}_i, i \neq j$ . Although the channel is quasi-static in the time domain, it may vary in the frequency domain due to the frequency selectivity caused by the multipath reflections.

Each of the  $M$  UAVs is equipped with a single PUF device [36], and due to complexity constraints, the corresponding PUF is installed only at a subset of UAVs. Therefore, the set  $\mathbb{M}$  can be partitioned into  $1 \leq L \leq M$  subsets such that  $\mathbb{M} = \{\overline{\mathbb{M}}_1 \cup \overline{\mathbb{M}}_2 \cup \dots \cup \overline{\mathbb{M}}_L\}$ , where  $\overline{\mathbb{M}}_i$  represents the subset of UAVs that have the PUFs of  $U_i$ . The  $i$ th UAV is considered to be aware of the channel state information (CSI) of all UAVs that belong to its subset, that is,  $U_i$  is aware of  $\mathbf{h}_{i,\ell} \forall \ell$ , where  $U_\ell \in \overline{\mathbb{M}}_i$  [37].

#### A. Pairwise Key Generation

The proposed protocol is based on the PKG process [18], as shown in Fig. 3. The pairwise keys are generated based on the principle of CR and the PUF-based key generation

algorithm [18]. For the generation of pairwise keys between  $U_m$  and  $U_n$ , it is required that  $U_n$  have PUF, while  $U_m$  have PUFs of  $U_n$ . The main steps of the PKG are as follows.

- 1) *Intermediate PKG*: Channel probing is performed by  $U_m$  and  $U_n$  using time-division duplexing (TDD) within the channel coherence time. The received signal strength (RSS)  $\triangleq \gamma$  is computed and the bit extraction is implemented to generate the intermediate key at both UAVs. Error reconciliation is applied to correct the bit errors at both UAVs. The resulting intermediate key is denoted as  $\tilde{\mathcal{K}}_{m,n}$ .
- 2) *Final PKG*: The intermediate pairwise key  $\tilde{\mathcal{K}}_{m,n}$  is used as a challenge to the PUF at  $U_n$  and to the PUFs at  $U_m$  to generate similar responses at both UAVs. Hashing is applied to the PUF output to generate the final pairwise key  $\mathcal{K}_{m,n}$ .

The keys generated using this protocol are random and have a high key rate as compared with other PLS-based key generation algorithms [18].

#### B. Set Partitioning

Generally speaking, the UAV subsets do not have to be uniform, that is, each subset may contain a different number of PUFs. However, such partitioning creates a complexity imbalance, where the UAVs in large subsets have to be equipped with a larger number of PUFs, and the gain is a higher probability of successful GK sharing. The extreme scenario that would result in the highest complexity is to have only one set, where each UAV has the PUFs of the other  $M - 1$  UAVs. The other extreme scenario provides the lowest complexity when only one PUF is installed on each UAV. In such scenarios, if  $U_i$  is down, then  $U_1, U_2, \dots, U_{i-1}$  and  $U_{i+1}, U_{i+2}, \dots, U_M$  will never be able to generate and share a GK. Furthermore, if  $U_i$  does not obtain the GK, then  $U_{i+1}, U_{i+2}, \dots, U_M$  will not be able to obtain the GK during this trial.

Practically speaking, all UAVs in a swarm typically have similar and limited computational capabilities. Consequently, it is considered in this work that all UAVs in the swarm can support the same number of PUFs, defined as  $\zeta$ . The value of  $\zeta$  is critical for specifying the reliability and complexity of the network. In this context, the optimum  $\zeta$

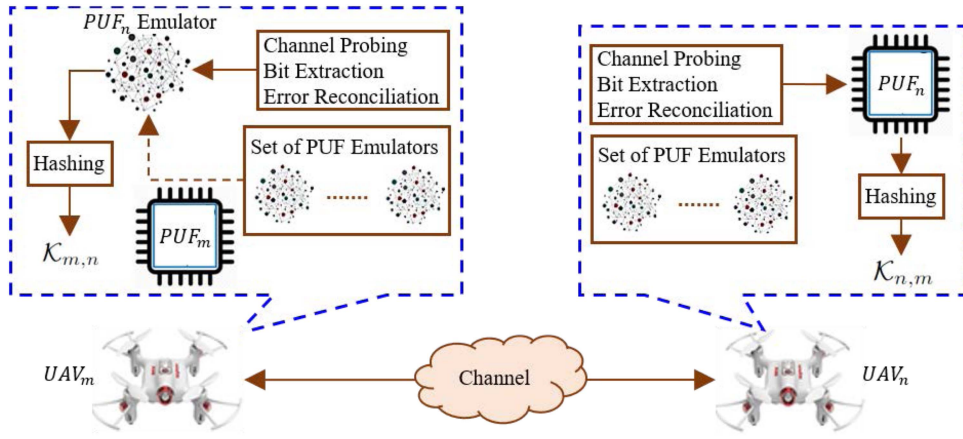


Fig. 3. System model for PKG between  $U_m$  and  $U_n$ , where  $U_n$  has the  $PUF_n$  and  $U_m$  has the corresponding  $PUF_n$  emulator.

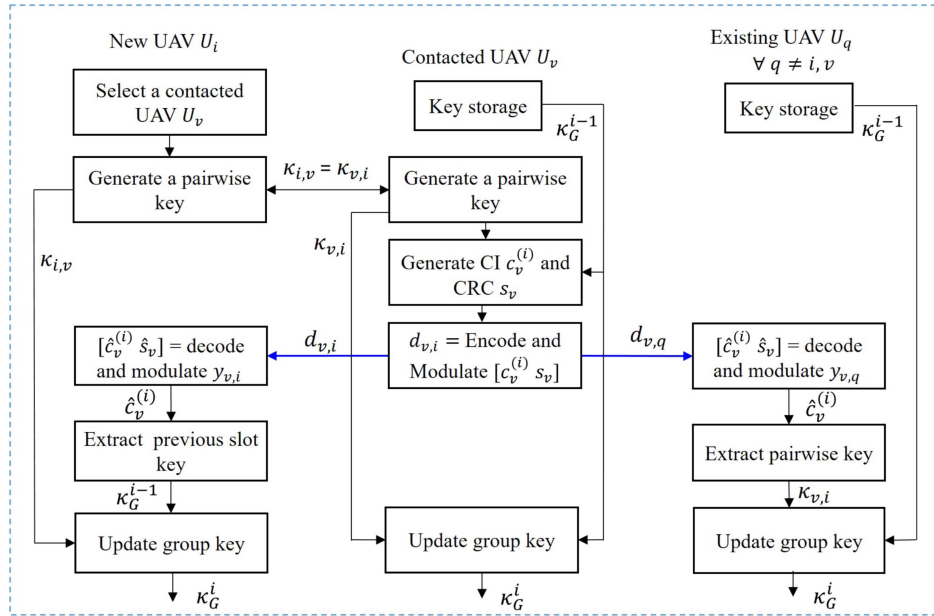


Fig. 4. Flowchart of the SSGK generation process.

can be defined as the value that maximizes the mesh connectivity while satisfying certain complexity constraints. To simplify the discussion, Fig. 5(a) presents the partitioning for the case of  $M = 5$  and  $\zeta = 1$ . The five subsets in this case are  $\bar{M}_1 = \{U_2\}$ ,  $\bar{M}_2 = \{U_3\}$ ,  $\bar{M}_3 = \{U_4\}$ ,  $\bar{M}_4 = \{U_5\}$ , and  $\bar{M}_5 = \{U_1\}$ . Fig. 5(b) shows the same swarm for  $\zeta = 2$ . The subsets in this case are  $\bar{M}_1 = \{U_2, U_3\}$ ,  $\bar{M}_2 = \{U_3, U_4\}$ ,  $\bar{M}_3 = \{U_4, U_5\}$ ,  $\bar{M}_4 = \{U_5, U_1\}$ , and  $\bar{M}_5 = \{U_1, U_2\}$ . Consequently, the redundancy order, in this case, is equal to the cardinality of the subsets, i.e.,  $|\bar{M}_i| \triangleq \zeta_i = \zeta \forall i$ . It is worth noting that, according to Assaf et al. [18], the PKG process should be initiated by the UAV that has the PUF. As an example, in Fig. 5(a),  $U_5$  can initiate the PKG process with  $U_4$ , but not vice-versa. Therefore, each UAV in Fig. 5(b) has two possible UAVs with which it can connect to initiate the pairwise key sharing process. The detailed description of the proposed scheme, denoted as SSGK, is given as follows.

#### IV. PROPOSED SSGK PROTOCOL

Due to the set partitioning process, it is necessary to manage the GKG process such that each UAV joins the group in a certain order. Moreover, the UAVs that fail to connect or become disconnected in a certain time frame  $T_i$  should be allowed to rejoin in the time frame  $T_{i+1}$ . Toward this goal, the time slot  $T_{GKG}$  in each time frame is divided into  $M$  subslots, as shown in Fig. 2, where  $T_{GKG} = \{\tau_1, \tau_2, \dots, \tau_M\}$ . Subslot  $\tau_i$  is reserved for  $U_i$  to attempt joining the mesh key agreement process. Without the loss of generality, consider that the UAVs join the GKG process sequentially in the order of their indices, i.e., the process starts with  $U_1$  in  $\tau_1$ , and then  $U_2$  attempts to connect with  $U_1$  in  $\tau_2$ , to generate the pairwise key  $\mathcal{K}_{1,2}$ . If  $U_i$  becomes disconnected or fails to obtain the GK, it can retry connecting in  $\tau_i$  in the next time frame.

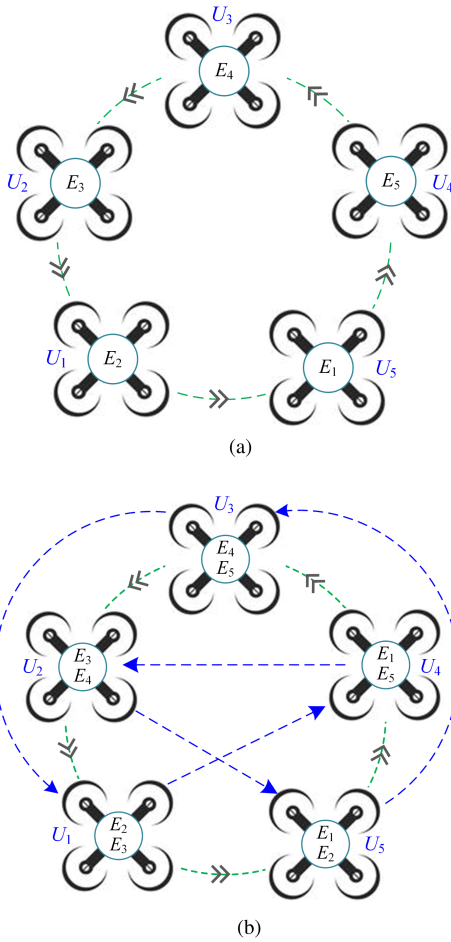


Fig. 5. Proposed SSGK without (a) and with (b) redundancy for a mesh size  $M = 5$ . The single blue arrows in (b) refer to the redundant links. (a) No redundancy ( $\zeta = 1$ ). (b) Second-order redundancy ( $\zeta = 2$ ).

### A. Protocol Terminologies

The following terminologies will be used to explain the protocol.

- 1) *New UAV*: The UAVs of the swarm are added in a sequential order, where  $U_i$  attempts to connect during  $\tau_i$ , such a UAV is referred to as the new UAV.
- 2) *Existing UAV*: The UAVs that managed to generate and share the GK in their respective slots.
- 3) *Contacted UAV*: A new UAV will select a certain existing UAV to generate a pairwise key, such a UAV is referred to as the contacted UAV.
- 4) *Subslot GK*: The GK might be updated every time subslot as a result of adding a new UAV. The GK after subslot  $i$  in which  $U_i$  is added is represented as  $\mathcal{K}_G^{(i)}$ .

### B. Protocol Overview

The flowchart of the algorithm with the communication sequence is presented in Fig. 4. A new UAV selects a contacted UAV based on the channel conditions and generates a pairwise key with it. The contacted UAV can be

selected from a set of UAVs that has the emulator of the new UAV present. The contacted UAV generates cooperation information (CI) using the pairwise keys present at the UAV and broadcasts it in the network. The generation of the CI is the core of the protocol and is used to extract the GK.

### C. Protocol Procedure

The proposed protocol has the following main steps.

1) *Initialization*: In  $T_1$ , the process starts when  $U_1$  and  $U_2$  attempt to generate the pairwise key  $\mathcal{K}_{1,2} = \mathcal{K}_{2,1}$  using the algorithm, as presented in Section III-A in  $\tau_2$ . The GK after  $\tau_2$  is  $\mathcal{K}_G^{(2)} = \mathcal{K}_{1,2}$ . The sequential key generation process requires certain operations to be performed by the new, contacted, and existing UAVs in each time subslot. Algorithms 1, 2, and 3 present the pseudocode of the new, contacted, and existing UAV, respectively.

2) *Contacted UAV Selection*: The new  $U_i$ ,  $i \in \{3, 4, \dots, M\}$ , joins the mesh in its respective  $\tau_i$  and will select the contacted UAV from subset  $\overline{\mathcal{M}}_i$  to generate a pairwise key. The contacted UAV is selected based on the channel conditions between the new UAV and the other UAVs in  $\overline{\mathcal{M}}_i$  to ensure that the PKG step is performed with the UAV that has the best channel conditions, i.e., the maximum average channel gain  $\bar{h}_{i,j} = \frac{1}{Q} \sum_{k=1}^Q h_{i,j}^{(k)}$ . Therefore, the index of the contacted UAV can be selected such that

$$v = \ell \left| \left[ |\bar{h}_{i,\ell}|^2 > |\bar{h}_{i,n}|^2 \right] \forall n \right. \\ \left. \{U_\ell, U_n\} \in \overline{\mathcal{M}}_i, \ell \neq n. \right. \quad (1)$$

Although the selection of the contacted node based on the channel strength is the defacto standard for such applications, channel probing and RSS computation may cause a time delay and increase the computational complexity. Furthermore, the A2A channels for communicating UAVs may exhibit equivalent strengths due to the line-of-sight signals in such channels [38]. Therefore, the selection based on RSS can be inefficient. As an alternative approach, the new  $U_i$  broadcasts a dummy key to all UAVs in  $\overline{\mathcal{M}}_i$ , and the one that acknowledges that it received the dummy key correctly will be considered as the contacted UAV. The proposed contacted UAV selection is called the successful link selection algorithm (SLSA).

3) *PKG*: The new  $U_i$  and the contacted  $U_v$  generate a pairwise key in  $\tau_i$ , which gives  $\mathcal{K}_{i,v} = \mathcal{K}_{v,i}$ . For a mesh network with  $M$  UAVs, a total of  $M - 1$  pairwise keys should be generated in  $T_{GKG}$  s in every time frame.

4) *CI Generation and Broadcasting*: It is the core of the protocol where the contacted UAV generates CI, which is broadcasted to the existing UAVs and new UAV. The CI should not leak any information about the GK and should only be meaningful to legitimate UAVs. In this work, CI is generated by performing an exclusive-OR (XOR) operation between the GK generated in  $\tau_{i-1}$  and the new pairwise key of the new UAV generated in  $\tau_i$ . The XOR process is uncorrelated in nature and decreases the leakage of information on the channel. Therefore,  $U_v$  generates the CI to add  $U_i$   $\tau_i$

as follows:

$$\mathbf{c}_v^{(i)} = \mathcal{K}_G^{(i-1)} \oplus \mathcal{K}_{v,i}. \quad (2)$$

The cyclic redundancy check (CRC) of  $\mathbf{c}_v^{(i)}$ , referred to as  $\mathbf{s}_v$ , is computed and appended to  $\mathbf{c}_v^{(i)}$ . The CRC is used to verify the transmitted data on the channel, which will eventually verify the correctness of the selected GSK at the existing UAVs. The CRC code consists of  $b$  bits and it approaches a misdetection probability of  $2^{-b}$  over the binary symmetric channel (BSC) for large bit error rate (BER) probabilities. Generally speaking, using 16-bit CRC provides near-ideal error detection [39]. The cooperative data along with the CRC,  $[\mathbf{c}_v^{(i)} \ \mathbf{s}_v]$ , are broadcasted by the contacted UAV to the new and existing UAVs,  $U_1, U_2, \dots, U_i$ . The signal received at  $U_i$  is denoted as  $\mathbf{y}_{v,i}$ .

5) *Key Extraction Process*: The key extraction process is performed at the existing and new UAVs apart from the contacted UAV. The existing  $U_q$  extracts  $\mathbf{c}_v^{(i)}$  and the estimated sequence is denoted as  $\hat{\mathbf{c}}_v^{(i)}$ . Then, the CRC of  $\mathbf{c}_v^{(i)}$  is computed, which gives  $\mathbf{s}_q$ , as described in Algorithm 1. After CRC matching,  $\mathcal{K}_{v,i}$  can be extracted as

$$\mathcal{K}_{v,i} = \mathcal{K}_G^{(i-1)} \oplus \mathbf{c}_v^{(i)}. \quad (3)$$

Similarly, at  $U_i$ ,  $\mathcal{K}_G^{(i-1)}$  is extracted as follows:

$$\mathcal{K}_G^{(i-1)} = \mathcal{K}_{v,i} \oplus \mathbf{c}_v^{(i)}. \quad (4)$$

Therefore, all the UAVs in the mesh until  $\tau_i$  will have two keys: a key for slot  $i - 1$ ,  $\mathcal{K}_G^{(i-1)}$ ; and a pairwise key shared between the new  $U_i$  and the contacted  $U_v$ .

6) *GK and Key Identity Number Update*: Each UAV has two keys present after the key extraction process in  $\tau_i$ ,  $\mathcal{K}_G^{(i-1)}$  and  $\mathcal{K}_{v,i}$ . The UAVs update the GK such that

$$\mathcal{K}_G^{(i)} = \min(\mathcal{K}_G^{(i-1)}, \mathcal{K}_{v,i}). \quad (5)$$

The sequential GKs generated with the addition of each UAV for a total of  $M$  UAVs are  $\{\mathcal{K}_G^{(1)}, \mathcal{K}_G^{(2)}, \dots, \mathcal{K}_G^{(M)}\}$ , and the GSK for  $T_1$  is considered to be  $\mathcal{K}_G^{(M)}$ . As can be noted, the number of pairwise keys generated by some GSK algorithms, such as [21], is  $\frac{M(M-1)}{2}$ , whereas in the worst case scenario, the number of pairwise keys required for the proposed SSGK is  $\zeta_i \times (M - 1) \forall i$  in a time frame.

7) *Special Case: No Redundancy*: For the special case of no redundancy,  $\zeta = 1$ , the new UAV PUF<sub>e</sub> is available only in one other UAV, as shown in Fig. 5(a). Because there is only one UAV that has the emulator, the step of selecting the contacted UAV is not required, and the UAV added in  $\tau_{i-1}$  is the contacted UAV, that is,  $v = i - 1$ . In each slot, the pairwise key is generated between the new  $U_i$  and the previously added  $U_{i-1}$ ,  $\mathcal{K}_{i,i-1} = \mathcal{K}_{i-1,i}$ . The CI is generated at the connected  $U_{i-1}$  as  $\mathbf{c}_{i-1}^{(i)} = \mathcal{K}_{i,i-1} \oplus \mathcal{K}_G^{i-1}$ . It should be emphasized that the proposed protocol requires generating a total of  $M - 1$  pairwise keys between all UAVs, and a similar number is required to transmit the XOR bits over the broadcast channel. The pairwise keys are generated as reported in [18], which requires less time to generate as compared with other PLS techniques.

## D. Verification and Disconnected Nodes

The CI is transmitted with a CRC to enable verification of the integrity of the received data. Each UAV that receives CI verifies its integrity using CRC. In case of error, the UAV may request a retransmission of CI if an automatic repeat request (ARQ) is adopted. If a CI is ultimately not received, the UAV becomes disconnected from the swarm and should wait for its time subslot in the next time frame to connect as a new UAV. The time frame number should be transmitted with the CI to create a sequence of frame GKs on all the UAVs. This will enable the disconnected UAV to identify that the key it has is outdated.

## V. PROPOSED SYSTEM PERFORMANCE

### A. Informal Security Analysis

In the literature, the resilience of GKG schemes has been discussed against various security attacks. Broadly, an attacker can be classified either as passive or active. From the context of the proposed GKG framework, any passive attacker may attempt to overhear the transmissions occurring during the key establishment phase and determine the GSK. On the contrary, an active attacker may inject harmful signals to interrupt the GKG process or manipulate the environment per his requirements. This work assumes that all participating nodes are initially mutually authenticated and the adversary is primarily passive. Apart from the passive adversary case, we also consider a case of active attack where an adversary tries to impersonate and attempts to participate in the key exchange process to determine the GSK. Furthermore, we do not consider denial-of-service (active attacks), such as jamming. Jamming is a typical attack in wireless communications, and several jamming avoidance solutions, such as frequency hopping and spread spectrum, can be incorporated into the proposed GKG framework to mitigate it.

The underlying GK generated from the proposed GKG framework applies equally to encryption and authentication use cases. Specifically, the key can encrypt common messages intended for a group/cluster of UAVs. Moreover, the key can serve as a secret token to identify the group members and provide a means of continuous authentication. Consequently, it is critical to test the strength of the final GSK. Moreover, the robustness and reliability of the proposed scheme from the security perspective must be ensured. The GKG framework must not leak or reveal any information that can come to the aid of the adversary in determining the key. Furthermore, the GKG framework should be resistant to impersonation attacks, machine-learning-based attacks, and stalking attacks [40]. Accordingly, an informal analysis of the security of the proposed protocol under various attacks and key randomness is discussed in the following.

- 1) *Key Randomness*: The security of the generated keystream can be verified by invoking the randomness test. Specifically, consider first the security of the keystream generated for the case, where there exist only two UAVs, i.e.,  $U_i$  and  $U_j$ . In such a case,

$\mathcal{K}_G = \mathcal{K}_{i,j}$ , where  $\mathcal{K}_{i,j}$  is the pairwise secret key and  $\mathcal{K}_G$  is the GSK. Consequently,  $\mathcal{K}_{i,j}$  was subjected to the randomness test using the National Institute of Science and Technology (NIST) standard test suite. Due to the constraint imposed by the minimum input length for the NIST test suite, eight NIST tests were executed. The corresponding  $p$  values for the tests were much larger than 0.01, which shows that the secret keys are random with 99% confidence. More exhaustive details on the randomness test are given in [18]. Furthermore, for the case with more than two UAVs, the final  $\mathcal{K}_G^{(i)}$  in the  $i$ th time slot will be obtained by first sharing the XOR of the generated pairwise key  $\mathcal{K}_{i,j}$  and a previously existing GSK  $\mathcal{K}_G^{(i-1)}$ , and then operating  $\min(\mathcal{K}_{i,j}, \mathcal{K}_G^{(i-1)})$ . Now, the minimum operation is executed at every node between the keystreams  $\mathcal{K}_G^{(i-1)}$  and  $\mathcal{K}_{i,j}$  each of which individually satisfies the NIST randomness test. Consequently, the final GSK, that is,  $\mathcal{K}_G^{(i)} = \min(\mathcal{K}_{i,j}, \mathcal{K}_G^{(i-1)})$  will satisfy the randomness test.

- 2) Forward and Backward Secrecy: The proposed protocol considers a dynamic group of UAVs and assumes that the participating members are authenticated. In addition, the transmission time is divided into frames, where the  $i$ th frame  $T_i$  is divided into two phases,  $T_{\text{GKG}}$  for the generation of GK and  $T_{\text{DT}}$  for the transmission of data.  $T_{\text{GKG}}$  is further divided into  $M$  subslots, and each subslot  $\tau_m$ , where  $m \in \{1, 2, \dots, M\}$ , is reserved for the  $m$ th UAV trying to join the mesh key agreement process. Every new incoming UAV  $m$  is permitted to join its dedicated subslot  $\tau_m$ , resulting in a new GSK. Furthermore, if a UAV $_m$  gets disconnected or is not able to join in the  $m$ th subslot, then UAV $_m$  is allowed to join only in the next time frame. Therefore, the proposed protocol distributively generates a GSK in a time-bound manner and is unaffected if a new member joins or leaves within the present time frame. Moreover, under quasi-static channel considerations, generally  $T_i \gg T_C$  ( $T_{\text{GKG}} < T_C$ ), where  $T_C$ , and hence, the GSK generation process in each time frame should be independent. Consequently, owing to the constraints imposed due to  $T_C$  and also since the joining and leaving of any member do not affect the GSK generation process, the previous and future GSK remain unaffected and uncorrelated. Accordingly, the proposed GSK protocol ensures forward and backward secrecy [41].
- 3) Resilience to Eavesdropping Attacks: During the GSK protocol, having more than two UAVs, the cooperative information  $\mathbf{c}_v^{(i)} = \mathcal{K}_G^{(i-1)} \oplus \mathcal{K}_{v,i}$  must be broadcast over the wireless channel so that all the network UAVs may agree to a common GSK. Notably, it is worthwhile to mention that the uncorrelated nature of the XOR operation ensures that the channel does not leak information. This is because the XOR of a random key stream with another random

and uncorrelated key stream yields another random stream [42]. For instance, let us denote the  $m$ th bit of  $\mathcal{K}_G^{(i-1)}$  and  $\mathcal{K}_{v,i}^i$  as  $g_m$  and  $k_m$ , respectively. Now, the probability that  $\Pr[g_m \oplus k_m = 0] = \Pr[g_m = k_m]$  can be further expressed as follows:

$$\begin{aligned} \Pr[g_m = k_m] &= \Pr[g_m = 0] \Pr[k_m = 0] \\ &\quad + \Pr[g_m = 1] \Pr[k_m = 1] \\ &= \left(\frac{1}{2}\right) \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) \left(\frac{1}{2}\right) = 0.5. \end{aligned} \quad (6)$$

Therefore, each bit in the XOR of  $\mathcal{K}_G^{(i-1)}$  and  $\mathcal{K}_{v,i}$  is chosen independently with a probability of 0.5, which means  $\mathbf{c}_v^{(i)}$  is a random string. Consequently, knowing  $\mathbf{c}_v^{(i)}$  gives no information about  $\mathcal{K}_G^{(i-1)}$  and  $\mathcal{K}_{v,i}$  except its length, even to an eavesdropper with unlimited time and power.

- 4) Resistance to Impersonation/Spoofing Attacks: The information transmitted on the channel is the XOR of the two pairwise keys stored at any UAV. The uncorrelated nature of the XOR operation ensures that the channel does not leak information. Even if an eavesdropper receives the message, it will not be able to extract the key from the transmitted messages. If a malicious user sends a joining request as described in the proposed SSGK to the mesh, it will not be able to generate the pairwise key with any other UAV in the mesh because the malicious user PUF $_e$  should be installed at certain legitimate UAVs. Consequently, the proposed scheme has inherent authentication that makes it improve its security against certain threats, such as the spoofing attack [34], [43], [44], [45], [46], [47], [48]. It is also worth noting that the PKG is based on the CR concept, which also improves the immunity against spoofing. Furthermore, if a malicious UAV receives the CI transmitted on the channel, it will not be able to extract the key from the received information because it needs to know either the pairwise key or the slot GK to extract the information of the other UAV. The sequential keys being generated with the addition of each UAV in a slot is  $\mathcal{K}_G^{(2)} = \mathcal{K}_{1,2}$ ,  $\mathcal{K}_G^{(3)} = \min(\mathcal{K}_G^{(2)}, \mathcal{K}_{i,3})$ , ...,  $\mathcal{K}_G^{(M)} = \min(\mathcal{K}_G^{(M-1)}, \mathcal{K}_{i,M})$ .
- 5) Resilience to Machine-Learning-Based Attacks: Recently, some studies have shown that PUF security can be compromised by using several machine-learning-based strategies [49]. Specifically, here the attacker by continuously monitoring the challenge-response tries to model the PUF behavior. However, in the presented work, the challenge is generated by exploiting the CR concept, thereby restricting machine-learning-based attacks on the PUFs.
- 6) Resilience to Stalking Attack: Any adversary, called a stalker, may follow the legitimate nodes' trajectory and may attempt to measure the corresponding



wireless channel [40]. Closer is the stalker node, easier for him to accurately estimate the channel. However, the involvement of PUFs in the process of deriving the key fails any attempt of the stalker to exploit the knowledge of the channel. Furthermore, in static and line-of-sight scenarios, the artificial fading component involved in the pairwise secret key generation protocol further provides inherent security against such types of attacks.

Furthermore, as depicted in Fig. 5(a), a group of PUFes is preconfigured at a certain UAV during the enrollment phase for the PKG process. Consequently, UAV anonymity is not guaranteed within that particular subset of UAVs, but it is not the case for UAVs in other sets. To guarantee anonymity across the entire swarm, the PKG can be performed while adopting anonymity enhancement schemes [35], [41], [50].

### B. Complexity

The proposed system complexity can be evaluated in terms of storage requirement, computational complexity, and hardware complexity.

The storage requirements for the proposed scheme can be evaluated by counting the size of the data that needs to be regularly updated. The static data can be stored in lookup table (LUT), and thus, it is considered a hardware overhead [51]. By referring to Algorithm 1, each UAV should store the following.

- 1) CSI: The channel vector  $\mathbf{h}$  consists of  $(\zeta - 1)Q$  samples. Given that each sample is represented by 8 bits, then the total number of bits is  $8(\zeta - 1)Q$ .
- 2) Intermediate and final pairwise keys  $\tilde{\mathcal{K}}_{m,n}$  and  $\mathcal{K}_{m,n}$ : Both keys have equal length of  $K$  bits. Therefore, the total is  $2K$  bits.
- 3) Old and new GKs: The total is  $2K$  bits.
- 4) CI: The length of the CI is equal to the GK length, which is  $K$  bits.
- 5) CRC bits: The length of the CRC is  $b$  bits.

Therefore, the total storage required is  $8(\zeta - 1)Q + 5K + b$  bits. For example, given that  $Q = K = 256$ ,  $b = 16$ , and  $\zeta = 1, 2, \dots, 5$ , then the total storage is, respectively, equal to 1.26, 3.26, 5.26, 7.26, and 9.26 kB.

For a pair of UAVs to share a pairwise key, one UAV should have a PUF and the other UAV should have access to the challenge–response pairs (CRPs) for that PUF. A common approach is to use LUTs that securely stores certain CRPs. The size of the LUT can be varied based on the available hardware resources. For UAVs with limited hardware resources, small-size LUTs can be used; however, such LUT should be updated for every new mission. Generally speaking, if the number of UAVs is less than 100 and the number of CRPs is less than 50 000, the total size of the LUTs is relatively small [34]. It is worth noting that if a PUF could be associated with a secret model that emulates the PUF behavior, then the secure storage requirements could be waived [52].

TABLE III  
Complexity Comparison

Ref.	Probing	Pairwise key	XOR	Broadcast
[21]	$\frac{M(M-1)}{2}$	$\frac{M(M-1)}{2}$	$M$	$M$
[25]	$M - 1$	1	$2(M - 1)$	$2(M - 1)$
[26]	$M$	$M$	$M$	$M$
SSGK	$M - 1$	$M - 1$	$M - 1$	$M - 1$

### Algorithm 1: SSGK Steps to Add New $U_i$ in $\tau_i$ .

- 1: Input:  $\mathbf{h}_{i,\ell} \forall \{U_i, U_\ell\} \in \overline{\mathbb{M}}_i, \mathbf{y}_{v,i}$
- 2: Compute  $v$  using (1) or SLSA  $\triangleright$  Select the contacted UAV for the PKG
- 3: Compute  $\mathcal{K}_{v,i} = \text{PKG}(\mathbf{y}_{v,i}, \text{PUF}_i)$   $\triangleright$  Generate the pairwise key [18]
- 4: Compute  $[\hat{\mathbf{c}}_v^{(i)} \hat{\mathbf{s}}_v]$  using  $\mathbf{y}_{v,i}$
- 5: Compute  $\mathbf{s}_i$  for  $\hat{\mathbf{c}}_v^{(i)}$   $\triangleright$  Verify the CRC at  $U_i$
- 6: **if**  $\mathbf{s}_i = \hat{\mathbf{s}}_v$  **then**
- 7:      $\mathcal{K}_G^{(i-1)} = \mathcal{K}_{v,i} \oplus \hat{\mathbf{c}}_v^{(i)}$   $\triangleright$  Extract GK for  $\tau_{i-1}$
- 8:      $\mathcal{K}_G^{(i)} = \min(\mathcal{K}_G^{(i-1)}, \mathcal{K}_{v,i})$   $\triangleright$  Generate new GK for  $\tau_i$
- 9: **end if**
- 10: Output:  $\mathcal{K}_G^{(i)}$

The complexity comparison of the proposed scheme with the PLS-CR is presented in Table III. As can be noted from the table, the complexity of the SSGK is significantly less than [21], while it is equivalent to [25] and [26].

### C. Communications Overhead

To share a GK, all UAVs in the mesh should exchange certain information, which forms communications overhead. For the proposed SSGK, the process starts with the channel probing process to generate the intermediate pairwise key between new and contacted UAVs. This process requires exchanging a secured version of the intermediate pairwise key  $\tilde{\mathcal{K}}_{m,n}$  multiple times until both UAVs agree on a key. The overhead of this process is equal to the number of iterations used to generate  $\tilde{\mathcal{K}}_{m,n}$  times the number of bits in the key  $K$ . However, based on the results in [18], the number of iterations is typically limited to one iteration. A similar process is applied to share the final pairwise key  $\mathcal{K}_{m,n}$ . Therefore, noting that  $M - 1$  UAVs have to share the pairwise key, the overhead for this process is  $2K(M - 1)$  bits. The proposed SSGK also requires sharing the pairwise key  $\mathcal{K}_{m,n}$  whenever a new UAV joins. Therefore, the corresponding overhead for this step is  $K(M - 2)$  because the GK sharing starts when the third UAV requests to join the mesh, and hence, the total overhead is  $K(3M - 4)$ . The communication overhead reported in [33] is  $832M + 960$  bits, independently of the key size  $K$ . In [33], the key is always hashed and, therefore, does not affect the size of the transmitted messages. Therefore, for a key size of up to 256 bits, the proposed SSGK still has a communication overhead lower than [33]. Consequently, the proposed SSGK can be considered efficient in terms of communication overhead compared to [33] and the references listed therein.

**Algorithm 2:** SSGK Algorithm at the Contacted  $U_v$  in  $\tau_i$ .

- 1: Input:  $\mathbf{y}_{v,i}, \mathcal{K}_G^{(i-1)}$
- 2:  $\mathcal{K}_{v,i} = \text{PKG}(\mathbf{y}_{v,i}, E_i) \triangleright$  Generate pairwise key for  $U_i$  and  $U_v$
- 3:  $\mathbf{c}_v^{(i)} = \mathcal{K}_{v,i} \oplus \mathcal{K}_G^{(i-1)} \triangleright$  Generate CI data for  $U_i$
- 4: Compute  $\mathbf{s}_v = \text{CRC}(\mathbf{c}_v^{(i)})$
- 5: Broadcast  $\mathbf{d}_v = [\mathbf{c}_v^{(i)} \ \mathbf{s}_v]$
- 6:  $\mathcal{K}_G^{(i)} = \min(\mathcal{K}_G^{(i-1)}, \mathcal{K}_{v,i})$
- 7: Output:  $\mathcal{K}_G^{(i)}$

**Algorithm 3:** GK Update for  $U_q \forall q \neq \{i, v\}$ .

**Input:**  $\mathbf{y}_{v,q}$

**Output:**  $\mathcal{K}_G^{(i)}$

- 1: Compute  $\hat{\mathbf{c}}_v^{(i)} \hat{\mathbf{s}}_v$  using  $\mathbf{y}_{v,q}$
- 2: Compute  $\mathbf{s}_q = \text{CRC}(\hat{\mathbf{c}}_v^{(i)})$
- 3: **if**  $\mathbf{s}_q = \hat{\mathbf{s}}_v$  **then**
- 4:  $\mathcal{K}_{v,i} = \mathcal{K}_G^{(i-1)} \oplus \mathbf{c}_v^{(i)} \triangleright$  Extract the pairwise key for  $U_i$  and  $U_v$
- 5:  $\mathcal{K}_G^{(i)} = \min(\mathcal{K}_G^{(i-1)}, \mathcal{K}_{v,i}) \triangleright$  Update GK
- 6: **end if**

TABLE IV  
Simulation Parameters' Values

Variable	Value
$K$	8, 12, 16, 24, 32, 64
$M$	3, 4, ..., 20
$\zeta$	1, 2, 3, 4
$p$	$10^{-1} - 10^{-5}$
Simulation run	$10^6$

## VI. NUMERICAL RESULTS

This section presents the numerical results to evaluate the performance of the proposed SSGK protocol. The results are generated using Monte Carlo simulation where each simulation point is generated using  $10^6$  key generation trials. Table IV presents the parameter sets considered in the simulation. The contacted UAV selection is performed using the SLSA and the channel is modeled as a BSC with a transition probability  $10^{-1} \geq p \geq 10^{-5}$ . The considered range of  $p$  covers a wide range of channel fading conditions, modulation, and coding schemes. The performance of the SSGK is compared with the efficient algorithm reported in [21]. To exclude the impact of the PKG process, the pairwise key is considered ideal, i.e., the new and existing UAVs consistently generate the pairwise keys successfully. To reduce the simulation complexity, CRC process is assumed to be perfect, i.e., the probabilities of false alarm and miss detection are equal to zero. Such performance can be obtained using 16 or 32 bits CRC [39]. The redundancy order is selected such that  $1 \leq \zeta \leq 4$ . The case of  $\zeta = 1$  is the minimum to enable connectivity between all the mesh UAVs and  $\zeta = 4$  is generally sufficient to provide high connectivity probability while maintaining a reasonable computational complexity. In addition to GK

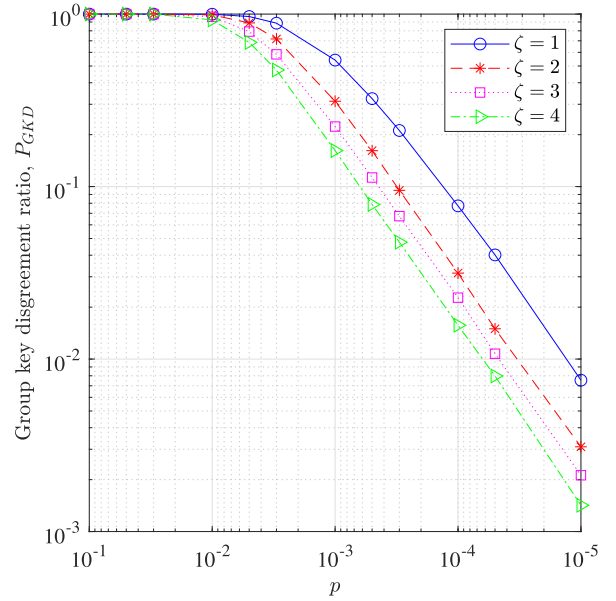


Fig. 6. GKD ratio and  $M_A$  for  $\zeta = [1, 2, 3, 4]$  using  $K = 64$ .

sharing, the proposed protocol can be used to share certain commands securely without the need for encryption. Therefore, the key/command lengths used covers a wide range of key lengths, which are  $\{8, 12, 16, 24, 32, 64\}$ . The work considers two performance evaluation metrics: the group key disagreement (GKD) ratio, which is defined as  $P_{\text{GKD}} = 1 - \frac{N_A}{N_T}$ , where  $N_A$  is the total number of times the GK is generated successfully for the  $M$  UAVs in  $T_1$  and  $N_T$  is the number of attempts made to generate the GK; and average UAVs in agreement, which is defined as the average number of UAVs that managed to share the GK successfully during  $T_1$ . The GKD is a commonly used metric for the GKG protocols [26].

Fig. 6 shows the GKD ratio and average UAVs in agreement versus  $p$  for  $\zeta = 1, 2, 3, 4, M = 10$ , and  $K = 64$ . The values of  $p$  are presented on the  $x$ -axis in decreasing order. As can be noted from the figure, the GKD ratio improves by decreasing  $p$ , and  $P_{\text{GKD}} = 1$  for  $p \gtrsim 10^{-2}$  for  $\zeta = 4$  and  $p \gtrsim 5 \times 10^{-2}$  for  $\zeta = 1$ . The figure shows that the gain obtained by increasing  $\zeta$  becomes smaller for large values of  $\zeta$ . For example, at  $p = 10^{-4}$ , increasing  $\zeta$  from 1 to 2 reduces  $P_{\text{GKD}}$  by about 62%. However, increasing  $\zeta$  from 2 to 3 reduces  $P_{\text{GKD}}$  by about 27%. It is worth noting that  $\zeta$  does not affect the system efficiency because it only increases the number of possible connections for the new UAV. Therefore, increasing  $\zeta$  may improve the key sharing success probability at the expense of some hardware complexity due to the increase in the number of PUFs.

Fig. 7 shows the average number of UAVs in agreement using the same settings as Fig. 6(a). It can be observed that, for  $p = 10^{-2}$ , 30% of the UAVs can connect in  $T_1$  when  $\zeta = 1$ , whereas 73% are able to connect using  $\zeta = 3$ . The impact of redundancy can also be observed at high values of  $p$ , such as 0.1, where using  $\zeta = 4$  offered about 45% connectivity, while using  $\zeta = 1$  offered only 13%.

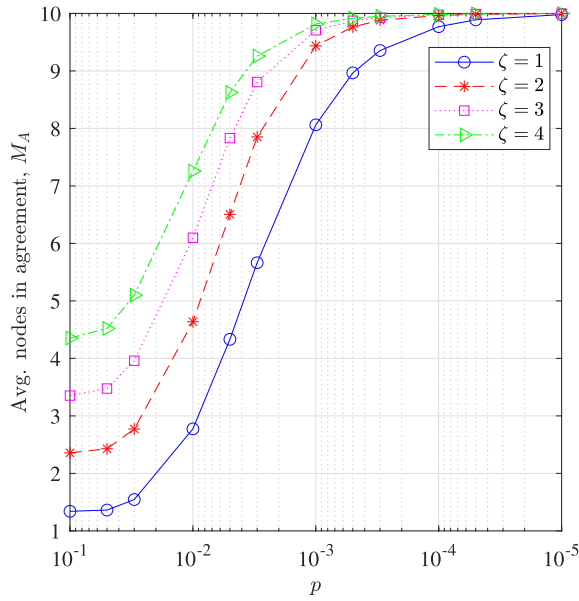


Fig. 7.  $M_A$  versus  $p$  for  $\zeta = [1, 2, 3, 4]$  using  $K = 64$ .

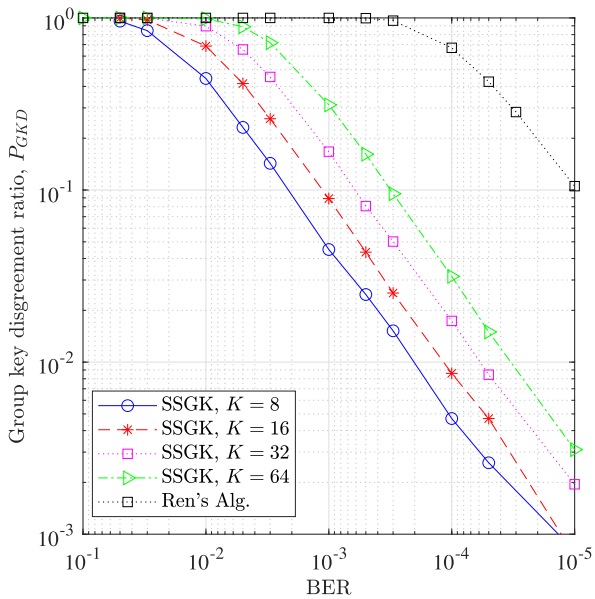


Fig. 8. SSGK disagreement ratio compared with Ren's algorithm [33] with varying key length and redundancy order  $\zeta_i = 2\forall i$ .

Fig. 8 is plotted for various values of  $K$  with  $\zeta = 2$ . As the figure shows, the key length has a significant impact on  $P_{GKD}$ . For example, at  $p = 10^{-3}$ , increasing  $K$  from 8 to 64 increases  $P_{GKD}$  by a factor of 6.6. For  $K > 16$ , the value of  $p$  should be less than  $10^{-4}$  to obtain  $P_{GKD}$  of  $10^{-2}$ . Moreover, the degradation ratio versus the key lengths seems roughly fixed for a wide range of  $p$ . It is worth noting that these results can be significantly improved when the performance is evaluated for multiple time frames. The same trends can also be noted for  $M_A$  in Fig. 9. For example, using  $K = 64$  provides  $M_A = 2.2$ , while for  $K = 8$ , it gives  $M_A = 4.2$ , i.e., 22% and 42% connectivity, respectively. Roughly speaking, the system provides connectivity of more than 90%, given that  $p < 10^{-3}$  for all the considered values of  $K$ .

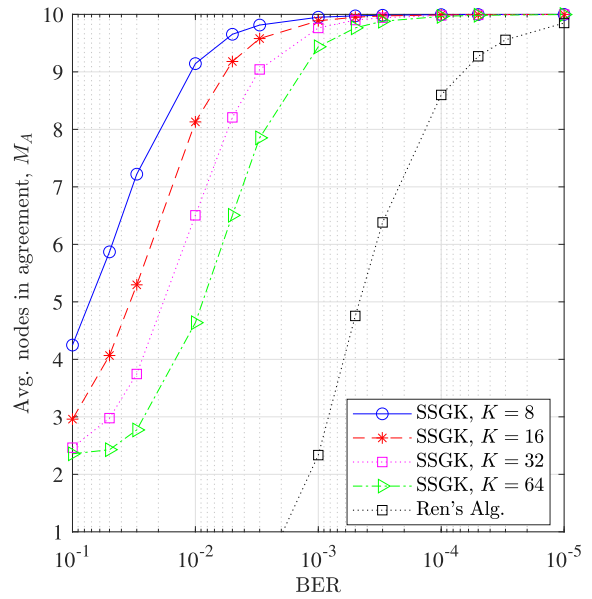


Fig. 9. SSGK average number of UAVs in agreement with various key lengths and redundancy order  $\zeta_i = 2\forall i$ . The SSGK results are compared with Ren's algorithm [33].

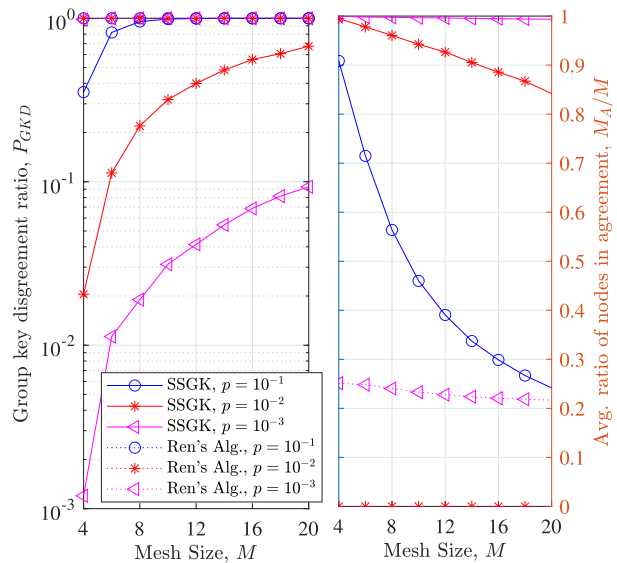


Fig. 10. Relationship of key disagreement ratio and the average ratio of UAVs in agreement with the mesh size for the key length of  $K = 64$  and redundancy order  $\zeta_i = 2$ . The comparison with Ren's algorithm [33] is also presented.

Figs. 8 and 9 also present the results of Ren's algorithm [33]. Ren et al. [33] present a group authentication and data transmission scheme using the PUF for NB-Internet of things (IoT) in which the output of the PUF is viewed as a shared root key for mutual authentication and key agreement. In this scheme, a group leader (GL) is used to aggregate and relay authentication information to the wired network side. The article assumes that the PUF has ideal stability and response. So, if the output of the PUF changes for any reason, the algorithm will fail. Although the proposed scheme can support a key of different sizes

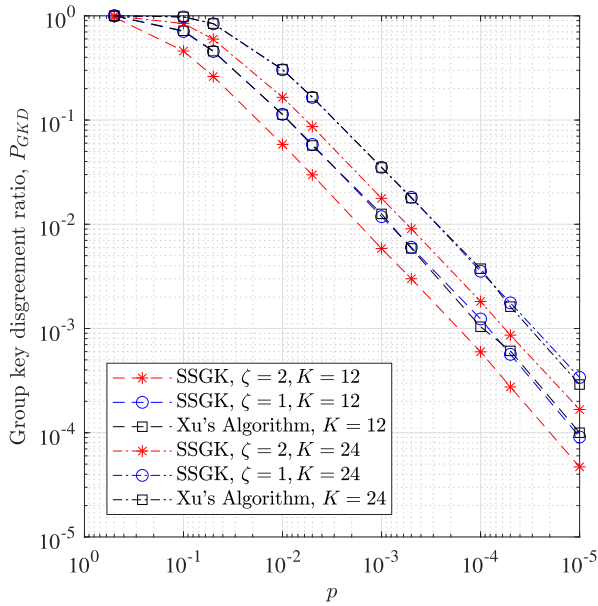


Fig. 11. Comparison of the SSGK with Xu's algorithm [21] for  $M = 3$ .

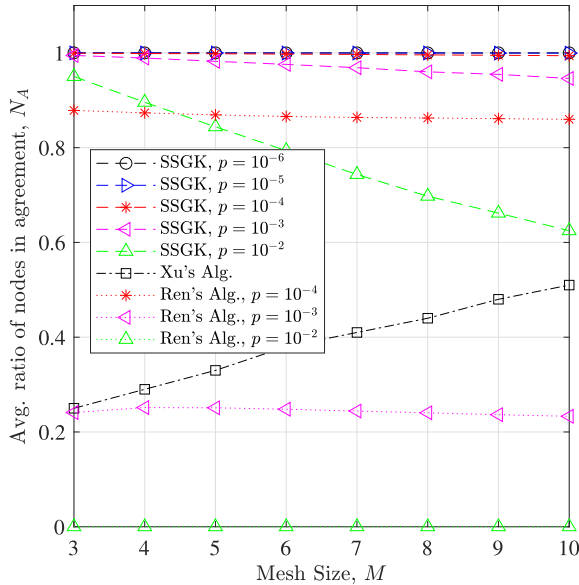


Fig. 12. Comparison of the SSGK with Xu's algorithm [21] and Ren's algorithm [33] for various mesh sizes and key length of  $K = 15$ .

because the key is always hashed, the size of the transmitted messages does not depend on the size of the key.

Fig. 10 shows the effect of the mesh size  $M$  on the  $P_{GKD}$  and the ratio  $M_A/M$ , where  $M_A$  are the UAVs that have the same key. The considered case assumes that  $\zeta_i = 2\forall i$  and  $K = 64$ . The key disagreement ratio is proportional to the mesh size, and the average ratio of UAVs in agreement is inversely proportional to the mesh size. The decrease in the ratio of UAVs in the agreement is significant at high BER.

In [33], although the use of a GL reduces signaling, it creates a single point of failure. Since a wireless link is never error-free, the protocol can suffer greatly from errors in the wireless link between IoT devices and the GL, and

even severe failures if the errors occur on the wireless link between the GL and the wired network.

Another scenario of  $K = 24$  and  $K = 12$ , and  $M = 3$  is presented in Fig. 11. For this, a comparison is performed with an existing protocol for GKG. It can be observed that the special case of no redundancy provides the same performance as in [21], but with more redundancy, the performance gain of the proposed protocol increases.

Fig. 12 shows the comparison of Xu et al.'s [21] work with the proposed SSGK algorithm in terms of the average ratio of nodes in agreement. For the proposed algorithm, the average ratio of nodes in agreement decreases with the size of the mesh. Xu's algorithm depends on all the pairwise keys generated compared with the proposed algorithm, which selects the GK as the minimum value of the pairwise keys generated. Due to this, a drastic improvement is observed in our algorithm compared with [21].

## VII. CONCLUSION AND FUTURE WORK

This work studied the problem of GSK generation for UAV swarm communications and proposed an efficient protocol based on the PLS. The proposed protocol follows a distributed approach in which no central node is used to control or coordinate the key generation process. To reduce the complexity of deploying PUFes in all UAVs, set partitioning is used where only a small number of emulators is deployed at each UAV. Having multiple PUFes at each UAV, as opposed to a single PUFe provides connection redundancy, which allows providing a performance improvement in the key disagreement ratio and the average number of UAVs in agreement. The obtained results showed that the proposed protocol can provide a low GKD ratio of about  $1.8 \times 10^{-2}$  for a channel transition probability of  $10^{-4}$ . For a channel transition probability of  $5 \times 10^{-3}$ , the average number of nodes approaches 100% when the swarm is composed of ten nodes.

Optimizing the cluster size based on the desired performance and mesh size is an interesting problem that will be investigated in future work. Moreover, reducing the communications overhead can be performed by using nonorthogonal multiplexing where the key generation bits can be combined with information bits to improve the system's spectral efficiency.

## APPENDIX A

### ACRONYMS

A2A	Air-to-air.
ARQ	Automatic repeat request.
BER	Bit error rate.
BSC	Binary symmetric channel.
CI	Cooperation information.
CR	Channel reciprocity.
CRC	Cyclic redundancy check.
CSI	Channel state information.
DOSS	Difference of signal strength.
GK	Group key.
GKD	Group key disagreement.

GKG	Group key generation.
GL	Group leader.
GSK	Group secret key.
IoT	Internet of Things.
LUT	Lookup table.
NIST	National Institute of Science and Technology.
PKG	Pairwise key generation.
PLS	Physical layer security.
PUF	Physically unclonable function.
PUFe	PUF emulator.
RSS	Received signal strength.
SLSA	Successful link selection algorithm.
TDD	Time-division duplexing.
UAV	Unmanned aerial vehicle.

#### LIST OF SYMBOLS

$T_{DT}$	Data transmission time.
$\mathcal{K}_G^{(i)}$	Group key after adding UAV <sub><i>i</i></sub> .
$\mathcal{K}_{m,n}$	Final pairwise key between UAV <sub><i>m</i></sub> and UAV <sub><i>n</i></sub> .
$\gamma_{v,i}$	Received signal strength for UAV <sub><i>v</i></sub> and UAV <sub><i>i</i></sub> .
$\hat{c}_v^{(i)}$	Estimated version of $c_v^{(i)}$ .
$\hat{s}_q$	CRC of $\hat{c}_v^{(i)}$ .
$\mathbb{M}$	Set of UAVs, $\mathbb{M} = \{\text{UAV}_1, \text{UAV}_2, \dots, \text{UAV}_M\}$ .
$\mathbf{c}_v^{(j)}$	CI for the addition of UAV <sub><i>i</i></sub> in time subslot <i>j</i> .
$\mathbf{s}_v$	CRC bits generated for $c_v^{(j)}$ .
$\mathbf{y}_{v,i}$	Signal received at UAV <sub><i>i</i></sub> from UAV <sub><i>v</i></sub> .
$\mathbb{M}_i$	Subset of UAVs that have the PUFes for UAV <sub><i>i</i></sub> .
$\mathcal{K}_{m,n}$	Intermediate pairwise key for UAV <sub><i>m</i></sub> and UAV <sub><i>n</i></sub> .
$\zeta$	Unified system cardinality.
$\zeta_i$	Cardinality of subset $\mathbb{M}_i$ .
$b$	Number of CRC bits.
$E_i$	Emulator for the PUF of UAV <sub><i>i</i></sub> .
$h_{i,j}$	Channel gain between UAV <sub><i>i</i></sub> and UAV <sub><i>j</i></sub> .
$i$	Index of the new UAV.
$K$	Key length.
$M$	Number of UAVs in the swarm.
$M_A$	Number of UAVs that have the same key.
$N_A$	Number of times the $M$ UAVs are in agreement.
$N_T$	Number of times the group key is generated for $M$ UAVs.
$p$	Channel transition probability.
$Q$	Number of frequency slots per transmission.
$q$	Index of an exiting UAV.
$T_C$	Channel coherence time.
$T_i$	<i>i</i> th time frame.
$T_{GKG}$	Time period for the GKG.
$T_{GKG}$	GKG time slot.
$U_i$	<i>i</i> th UAV.
$v$	Index of the contacted UAV.
$\text{PUF}_{v,i}$	PUF of UAV <sub><i>i</i></sub> at UAV <sub><i>v</i></sub> .

#### REFERENCES

- [1] J. Chang, N. Dong, D. Li, W. H. Ip, and K. L. Yung, "Skeleton extraction and greedy algorithm based path planning and its application in UAV trajectory tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, pp. 4953–4964, Dec. 2022.
- [2] S. Jangsher, M. Al-Jarrah, A. Al-Dweik, E. Alsusa, and P.-Y. Kong, "Energy constrained sum-rate maximization in IRS-assisted UAV networks with imperfect channel information," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 3, pp. 2898–2908, Jun. 2023, doi: 10.1109/TAES.2022.3220493.
- [3] S. Jangsher, M. Al-Jarrah, A. Al-Dweik, E. Alsusa, and M.-S. Alouini, "BER reduction using partial-elements selection in IRS-UAV communications with imperfect phase compensation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 1, pp. 623–633, Feb. 2023, doi: 10.1109/TAES.2022.3188590.
- [4] J. Choi, M.-G. Seo, H.-S. Shin, and H. Oh, "Adversarial swarm defence using multiple fixed-wing unmanned aerial vehicles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, pp. 5204–5219, Dec. 2022.
- [5] D. Hambling, "US military plan to create huge autonomous drone swarms sparks concern," Accessed on: Jun. 22, 2023. [Online]. Available: <https://www.newscientist.com/article/2357548>
- [6] "IP mesh OEM digital data link for UAV wireless transmitting video and control data," Accessed on: Jun. 15, 2023. [Online]. Available: <https://www.iwavecomms.com/>
- [7] H. Tan, W. Zheng, P. Vijayakumar, K. Sakurai, and N. Kumar, "An efficient vehicle-assisted aggregate authentication scheme for infrastructure-less vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2022.3176406.
- [8] H. Tan, W. Zheng, and P. Vijayakumar, "Secure and efficient authenticated key management scheme for UAV-assisted infrastructure-less IoVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6389–6400, Jun. 2023.
- [9] C. Pu, A. Wall, K.-K. R. Choo, I. Ahmed, and S. Lim, "A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of drones environment," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9918–9933, Jun. 2022.
- [10] 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. 31420–31440, 2021.
- [11] S. U. Jan, I. A. Abbasi, and F. Algarni, "A key agreement scheme for IoD deployment civilian drone," *IEEE Access*, vol. 9, pp. 149311–149321, 2021.
- [12] J. M. Hamamreh, H. M. Furqan, and H. Arslan, "Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 21, no. 2, pp. 1773–1828, Apr./Jun. 2019.
- [13] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 17–30, Jan. 2010.
- [14] A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of physical layer security in multiuser wireless networks: A survey," *IEEE Commun. Surv. Tut.*, vol. 16, no. 3, pp. 1550–1573, Jul./Sep. 2014.
- [15] N. Aldaghri and H. Mahdavi, "Physical layer secret key generation in static environments," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2692–2705, Feb. 2020.
- [16] C. F. E. de Melo et al., "UAVouch: A secure identity and location validation scheme for UAV-networks," *IEEE Access*, vol. 9, pp. 82930–82946, 2021.
- [17] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.
- [18] T. Assaf et al., "High-rate secret key generation using physical layer security and physical unclonable functions," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 209–225, Jan. 2023.
- [19] Y.-C. Lai, C.-Y. Yao, S.-H. Yang, Y.-W. Wu, and T.-T. Liu, "A robust area-efficient physically unclonable function with high machine learning attack resilience in 28-nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 1, pp. 347–355, Jan. 2022.
- [20] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. C. Bansal, "A comprehensive review of authentication schemes in vehicular ad-hoc network," *IEEE Access*, vol. 9, pp. 31309–31321, 2021.

- [21] P. Xu, K. Cumanan, Z. Ding, X. Dai, and K. K. Leung, "Group secret key generation in wireless networks: Algorithms and rate optimization," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 8, pp. 1831–1846, Aug. 2016.
- [22] T. Li et al., "Energy-efficient and secure communication toward UAV networks," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10061–10076, Jun. 2022.
- [23] K. Gai, Y. Wu, L. Zhu, K.-K. R. Choo, and B. Xiao, "Blockchain-enabled trustworthy group communications in UAV networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4118–4130, Jul. 2021.
- [24] H. Liu, J. Yang, Y. Wang, Y. Chen, and C. E. Koksall, "Group secret key generation via received signal strength: Protocols, achievable rates, and implementation," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2820–2835, Dec. 2014.
- [25] S. Xiao, Y. Guo, K. Huang, and L. Jin, "Cooperative group secret key generation based on secure network coding," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1466–1469, Jul. 2018.
- [26] S. Peng, B. Han, C. Wu, and B. Wang, "A secure communication system in self-organizing networks via lightweight group key generation," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 182–192, Sep. 2020.
- [27] H. Jagadeesh, R. Joshi, and M. Rao, "Group secret-key generation using algebraic rings in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1538–1553, Feb. 2021.
- [28] C. D. T. Thai, J. Lee, J. Prakash, and T. Q. S. Quek, "Secret group-key generation at physical layer for multi-antenna mesh topology," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 1, pp. 18–33, Jan. 2019.
- [29] D. Deng, S. Hou, Z. Wang, and Y. Guo, "Configurable ring oscillator PUF using hybrid logic gates," *IEEE Access*, vol. 8, pp. 161427–161437, 2020.
- [30] R. Pappu, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.
- [31] P. Mall, R. Amin, A. K. Das, M. T. Leung, and K.-K. R. Choo, "PUF-based authentication and key agreement protocols for IoT, WSNs, and smart grids: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8205–8228, Jun. 2022.
- [32] V. Suresh, R. Kumar, M. Anders, H. Kaul, V. De, and S. Mathew, "A 0.26% BER, 1028 challenge-response machine-learning resistant strong-PUF in 14 nm CMOS featuring stability-aware adversarial challenge selection," in *Proc. IEEE Symp. VLSI Circuits*, 2020, pp. 1–2.
- [33] X. Ren, J. Cao, M. Ma, H. Li, and Y. Zhang, "A novel PUF-based group authentication and data transmission scheme for NB-IoT in 3GPP 5G networks," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3642–3656, Mar. 2022.
- [34] H. Yildiz, M. Cenk, and E. Onur, "PLGAKD: A PUF-based lightweight group authentication and key distribution protocol," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5682–5696, Apr. 2021.
- [35] T.-F. Lee, X. Ye, and S.-H. Lin, "Anonymous dynamic group authenticated key agreements using physical unclonable functions for Internet of Medical Things," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 15336–15348, Aug. 2022.
- [36] K. Lounis and M. Zulkernine, "More lessons: Analysis of PUF-based authentication protocols for IoT," *Cryptology ePrint Archive*, 2021, Paper no. 2021/1509, [Online]. Available: <https://eprint.iacr.org/2021/1509>
- [37] G. Wang, F. Gao, W. Chen, and C. Tellambura, "Channel estimation and training design for two-way relay networks in time-selective fading environments," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2681–2691, Aug. 2011.
- [38] M. Al-Jarrah, A. Al-Dweik, E. Alsusa, Y. Iraqi, and M.-S. Alouini, "On the performance of IRS-assisted multi-layer UAV communications with imperfect phase compensation," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8551–8568, Dec. 2021.
- [39] H. Mukhtar, A. Al-Dweik, and M. Al-Mualla, "CRC-free hybrid ARQ system using turbo product codes," *IEEE Trans. Commun.*, vol. 62, no. 12, pp. 4220–4229, Dec. 2014.
- [40] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *Proc. IEEE IN-FOCOM*, 2013, pp. 3048–3056.
- [41] H. Zhang, W. Bian, B. Jie, D. Xu, and J. Zhao, "A complete user authentication and key agreement scheme using cancelable biometrics and PUF in multi-server environment," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 5413–5428, Dec. 2021.
- [42] J.-P. Aumasson, *Serious Cryptography: A Practical Introduction to Modern Encryption*. San Francisco, CA, USA: No Starch Press, 2017.
- [43] M. A. Qureshi and A. Munir, "PUF-RAKE: A PUF-based robust and lightweight authentication and key establishment protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2457–2475, Jul./Aug. 2022.
- [44] O. Millwood, J. Miskelly, B. Yang, P. Gope, E. B. Kavun, and C. Lin, "PUF-phenotype: A robust and noise-resilient approach to aid group-based authentication with DRAM-PUFs using machine learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 2451–2465, Apr. 2023.
- [45] S. Li, T. Zhang, B. Yu, and K. He, "A provably secure and practical PUF-based end-to-end mutual authentication and key exchange protocol for IoT," *IEEE Sensors J.*, vol. 21, no. 4, pp. 5487–5501, Feb. 2021.
- [46] C. Gu, C.-H. Chang, W. Liu, S. Yu, Y. Wang, and M. O'Neill, "A modeling attack resistant deception technique for securing lightweight-PUF-based authentication," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1183–1196, Jun. 2021.
- [47] M. Ebrahimabadi, M. Younis, and N. Karimi, "A PUF-based modeling-attack resilient authentication protocol for IoT devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3684–3703, Mar. 2022.
- [48] Y. Zheng, W. Liu, C. Gu, and C.-H. Chang, "PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 3299–3316, Jul./Aug. 2023.
- [49] P. Santikellur, A. Bhattacharyay, and R. Chakraborty, "Deep learning based model building attacks on arbiter PUF compositions," 2019. [Online]. Available: <https://eprint.iacr.org/2019/566>
- [50] U. Chatterjee, D. Mukhopadhyay, and R. S. Chakraborty, "3PAA: A private PUF protocol for anonymous authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 756–769, 2021.
- [51] U. Chatterjee et al., "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 3, pp. 424–437, May/Jun. 2019.
- [52] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.



**Sobia Jangsher** (Member, IEEE) received the Ph.D. degree in wireless communication from The University of Hong Kong, Hong Kong, in 2015.

From 2015 to 2021, she was an Assistant Professor with the Institute of Space Technology, Islamabad, Pakistan. She was associated with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, UAE, 2021 to 2023. She is currently an Assistant Professor with the School of Electronic

Engineering, Dublin City University, Dublin, Ireland. Her research mainly focuses on optimization/resource allocation for IRS systems, multiple access schemes, and small cell networks.



**Arafat Al-Dweik** (Senior Member, IEEE) received the M.S. degree (*summa cum laude*) and the Ph.D. degree (*magna cum laude*) in electrical engineering from Cleveland State University, Cleveland, OH, USA, in 1998 and 2001, respectively.

He was with Efficient Channel Coding, Inc., Cleveland, USA, the Department of Information Technology, Arab American University in Palestine, and the University of Guelph, ON, Canada. Since 2003, he has been with the Department

of Electrical Engineering, Khalifa University, Abu Dhabi, UAE. He is currently an Adjunct Research Professor with Western University, London, ON, Canada, the University of Guelph, Guelph, ON, Canada, and the University of Manchester, Manchester, U.K.

Dr. Al-Dweik is a recipient of Fulbright Scholarship, the Hijjawi Award for Applied Sciences, the Fulbright Alumni Development Grant, Dubai Award for Sustainable Transportation, and the Leader-Founder Award in UAE. He serves as an Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and *IET Communications*. He is a registered Professional Engineer in the province of Ontario, Canada.



**Youssef Iraqi** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Montreal, Montreal, QC, Canada, in 2000 and 2003, respectively.

He is currently an Associate Professor with the College of Computing, Mohammed VI Polytechnic University, Ben Guerir, Morocco. Before that, he was with the Department of Electrical Engineering and Computer Science, Khalifa University (KU), UAE, for 12 years. Before joining KU, he was the Chair of the Department

of Computer Science, Dhofar University, Oman, for four years. From 2004 to 2005, he was a Research Assistant Professor with the David R. Cheriton School of Computer Science, University of Waterloo, Canada. He has authored or coauthored more than 130 research papers in international journals and refereed conference proceedings. His research interests include resource management in wireless networks, blockchain, trust and reputation management, cloud computing, and stylometry.

Dr. Iraqi was a recipient of the IEEE Communications Society Fred W. Ellersick Paper Award in the field of communications systems in 2008. He is on many technical program committees of international conferences and is often approached for his expertise by international journals in his field.



**Anshul Pandey** (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the Indian Institute of Information Technology Design and Manufacturing, Jabalpur, India, in 2012, the M.Tech. degree in advance networks from the Indian Institute of Information Technology and Management, Gwalior, India, in 2016, and the Ph.D. degree from the Department of Electronics and Communication Engineering, Indian Institute of Information Technology Allahabad, Allahabad,

India, in 2021.

He is currently a Senior Researcher with Secure Systems Research Center, Technology Innovation Institute, Abu Dhabi, UAE. His research interests include cooperative relaying for wireless vehicular networks, physical layer security, reconfigurable intelligent surfaces, and signal processing.



**Jean-Pierre Giacalone** (Member, IEEE) received the B.Sc. degree in micro-electronics from the École Nationale Supérieure d'électrotechnique, d'électronique, d'informatique, d'hydraulique et des télécommunications, Toulouse, France, in 1985.

He is the Vice President of secure communications engineering with Secure Systems Research Centre, Technology Innovation Institute, Abu Dhabi, UAE. He is responsible for researching secure communications, focusing on improving

the resilience of cyber-physical and autonomous systems. He has worked as an expert in software architecture for advanced driving assistance systems with Renault and as a Principal Engineer and Architect within Intel's Mobile Systems Technologies Group. He holds 19 patents and has coauthored 15 research papers accepted for publication in international journals and conference proceedings.