SACRED EYE: Secure Communication and Decentralized Monitoring for Swarm UAV Mission Completion

Hugo Le Dirach
University of Toulouse
Toulouse, Occitanie, France
hugo.le_dirach@onera.fr

Amin Rezaei
California State University Long Beach
Long Beach, California, USA
amin.rezaei@csulb.edu

***Abstract*—The flexibility, scalability, and robustness of Unmanned Aerial Vehicle (UAV) swarms make them highly adaptable to dynamic environments. However, these advantages also introduce a range of optimization challenges, including constraints on size, weight, and energy consumption, as well as less explored topics such as lightweight, secure communication and reliable mission completion. Inspired by the ant identification methods, we propose a decentralized solution to enhance communication security and ensure mission completion within UAV swarms. Unlike existing methods that rely on Physical Unclonable Functions (PUFs), our solution introduces a PUF-less framework. Specifically, we define private IDs and group keys for individual UAVs, enabling secure communication while minimizing the risk of intrusion. The effectiveness of our method is validated through simulations, which highlight the importance of periodically updating the group key. Additionally, we address targeted attacks on private IDs by incorporating behavioral analysis and shared monitoring among swarm members. Our proposed approach paves the way for realizing the full potential of UAV swarms.**

***Index Terms*—Swarm UAV Security; Decentralized Monitoring; Behavioral Analysis; Mission Completion**

I. Introduction

The fusion of Unmanned Aerial Vehicles (UAVs) with interconnected networks has ushered in a new era of possibilities, where UAVs seamlessly communicate and collaborate within a vast ecosystem. From enhancing efficiency in various industries to unlocking novel applications in emergency response and surveillance, the Internet of Drones (IoD) heralds a future where interconnected flying machines play a pivotal role in shaping our interconnected world. Rising interest in the matter of IoD in the research community has raised multiple optimization problems such as size, weight, and energy consumption, as well as the need for lightweight, secure communication and reliable mission completion.

There are three main advantages of swarm UAVs [1]. First is flexibility, which is the ability of swarms to adapt to changing environments thanks to the self-organization of the system. Second is scalability, that is implied by the similarity of behaviors that characterize swarm UAVs and their modes of communication. Adding or subtracting UAVs does not change the behavior of the operating UAVs; it will only affect the efficiency or range of the operations. Finally, the ability for independent task assignment allows the process to continue even if some UAVs fail; the remaining UAV will carry out the work and constitute a fault-tolerant system. Despite the advantages of swarm UAVs, there are challenges such as strict regulations for UAV operation, difficult working environments, and the need for sensible configurations.

Taking inspiration from ant identification methods, in this paper, we develop a robust solution to secure the swarm UAVs entitled **SACRED EYE**: **S**ecure **A**uthentication, flow **C**ont**R**ol, b**E**havior **D**etection, and **E**ncr**Y**ption for dron**E** protection. Our solution objectives are, first, *initialization* to secure the activation of units and their initial connection with the group; second, *continuation* to maintain secure traffic of information between the UAVs; and third, *opacity* to prevail leaking information from a stolen UAV. Our contributions are as follows:

- Leveraging symmetric encryption combined with network topology to secure sensitive information without using Physical Unclonable Functions (PUFs);
- Providing a decentralized monitoring solution for swarm UAVs in order to detect rogue UAVs;
- Proposing the notion of the circular variable in order to secure communication within a swarm;
- Developing a simulator to mimic the behavior of the swarm UAVs and depicting the efficiency and robustness of our solution through multiple simulations.

II. Literature Study

The use of low-cost UAVs and standard RGB cameras mounted on UAVs for monitoring municipal solid waste landfills has been explored [2]. In addition, another study has explored the potential of real-time decision-making using UAVs in precision livestock and farming and suggested that affordable off-the-shelf UAV technology could enhance plant and cattle monitoring in indoor agricultural settings [3]. Moreover, reports such as [4] have provided details on large operations that could be led by a multi-agent system on the subject of collaborative transportation.

The subject of communication security can be generally divided into two underlying domains: software security and hardware security. There are about twice as many software-based solutions as hardware-based solutions on the matter of security and privacy issues for the IoD [5]. The state-of-the-art works on software authentication mechanisms for IoD networks encompass conventional technologies like hash functions [6], Public Key Infrastructure (PKI) [7], and Elliptic-Curve Cryptography (ECC) [8] as well as emerging technologies such as Mobile Edge Computing (MEC) [9], Machine Learning (ML) [10], and Blockchain [11]. Additionally, hardware-based solutions for node identification and authentication within IoD include Trusted Platform Modules (TPMs) [12], Hardware Security Modules (HSMs) [13], and PUFs [14]. In addition, a two-stage lightweight identity authentication and key agreement protocol for UAVs has been introduced to address communication security challenges in harsh natural environments [15]. Additionally, a PUF is embedded in UAV hardware to bolster network communication security against physical capture attacks. Furthermore, a similar architecture is combined with PUF scheme identification to secure communications [16]. The aim of such architecture is to consolidate the security hardware tools in a single UAV, removing them from the member UAVs, because such protections are weight-consuming and can add a non-negligible computation overhead. In addition to the dependence of these works

979-8-3315-8924-0/25/$31.00 ©2025 IEEE

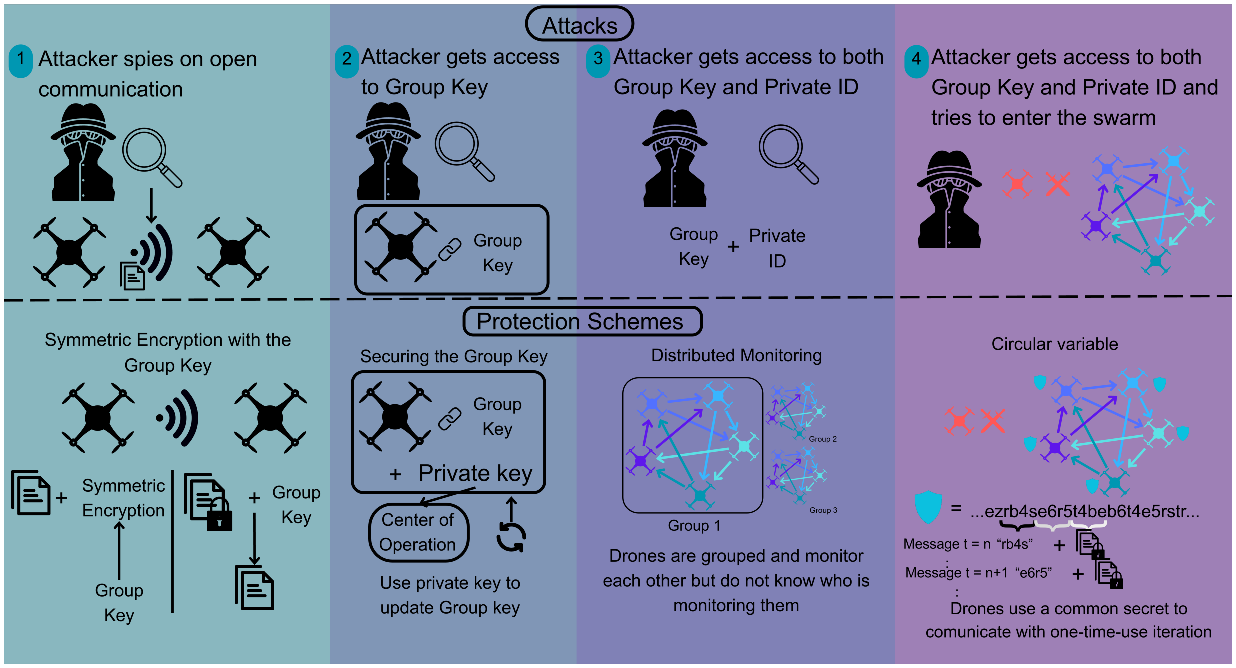

Attacks
1 Attacker spies on open communication
2 Attacker gets access to Group Key
Group Key
3 Attacker gets access to both Group Key and Private ID
Group Key + Private ID
4 Attacker gets access to both Group Key and Private ID and tries to enter the swarm
Protection Schemes
Symmetric Encryption with the Group Key
+ Symmetric Encryption
Group Key
+ Group Key
Securing the Group Key
Group Key
+ Private key
Center of Operation
Use private key to update Group key
Distributed Monitoring
Group 1
Group 2
Group 3
Drones are grouped and monitor each other but do not know who is monitoring them
Circular variable
= ...ezrb4se6r5t4beb6t4e5rstr...
Message t = n "rb4s" +
Message t = n+1 "e6r5" +
Drones use a common secret to comunicate with one-time-use iteration

Fig. 1: Visual representation of SACRED EYE

on noise-prune PUFs, the choice to conglomerate the weight on a computing UAV that centralizes communication encryption opens the system to a single point of failure. While PUFs seem to be secure for authentication, the environment in which a UAV evolves can have frequent changes in temperature, pressure, and humidity that may all impact the integrity of a PUF system. In terms of PUF-less approaches, a centralized global security method is proposed using the internet Transport Layer Security (TLS) protocol [17]. In addition, other papers [18] [19] use blockchain to secure access to shared information. Our goal is to put forward a PUF-less and decentralized hardware security system to create a robust solution for any environment without compromising on the reliability.

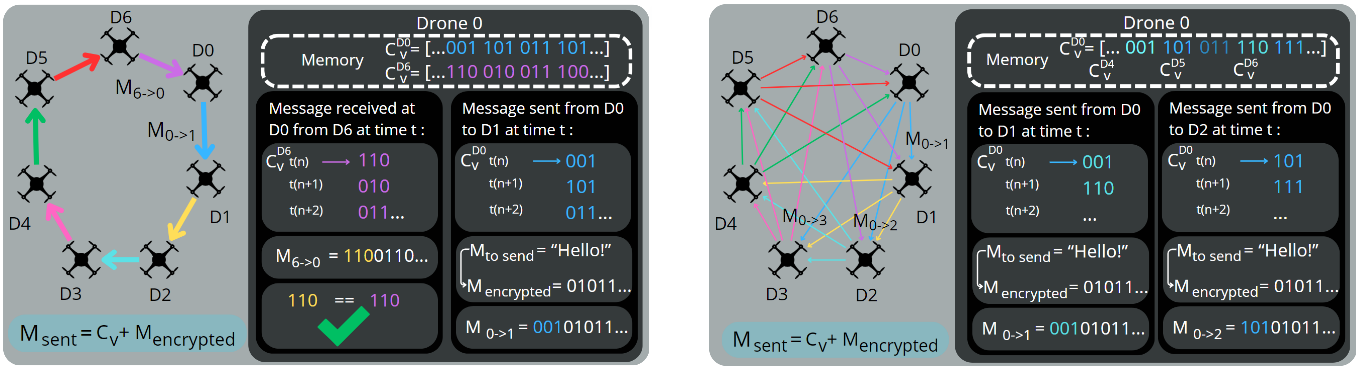
III. SACRED EYE

In this section, we propose a PUF-less decentralized solution based on ant colonies called **SACRED EYE** for UAV communication security and mission completion shown in Fig. 1. **SACRED EYE** uses four securing elements: ❶ a private ID, stored on hardware, that allows the UAV to be identified by the center of command; ❷ a group key shared by the UAVs to communicate with each other with a symmetric encryption scheme; ❸ a circular variable that is used to synchronize communications and prevent any intrusions once the exchanges have started; and ❹ a behavior analysis method to secure the completion of the tasks.

Our goal is to secure the exchange of information without using command and computing UAVs. Here are the postulates we consider in our proposed method: (1) The swarm of UAVs is decentralized, meaning that the planning algorithm is set up by the base upon initialization of the UAVs, and then they operate autonomously with no communication with a control operator. Of course, each UAV must go back to the base at intervals to report and get the update. (2) The only communication each UAV exchanges is in proximity with the surrounding UAVs that share the same communication parameters. (3) Both telecommunication and hardware attacks are considered, assuming they are instantaneous and complete. Ants have served as an inspiration for a variety of approaches, the most well-researched and effective of which is ant colony optimization [20]. While ant colonies mostly use pheromones for communication, they can have various communication methods depending on the environment in which they are evolving [21], [22]. The way the UAVs identify each other and exchange information can be mimicked to copy the communication network architecture of an ant colony. We choose to simulate the theory of our approach by emulating the communication of ants because, first, it allows us to limit ourselves to two degrees of spatial freedom of movement that is easier to create and more observable, and second, the collective task of seeking and retrieving food (or any other given size appropriate object) gives us a measurement point for a possible overhead of our solution compared with a non-secure communication protocol. Also, in this paper, we are not aiming for an optimal path finding. Other works, such as [23] have followed such an aim.

A. *First Line of Defense*

The first concept is the establishment of a group key and a private ID for each UAV within the swarm. The private ID, held at the center of operation and in the hardware of the UAV, serves as the foundation for UAV recognition, enabling the center to identify individual UAVs within the swarm. The group key, on the other hand, enables secure communication among the UAVs; it plays a vital role in authenticating and encrypting messages exchanged among UAVs, ensuring that only authorized UAVs can participate in communication, and preventing eavesdropping by malicious entities. **In this first scenario, we assume that the private ID is protected against unauthorized access, but the group key can be leaked.** The group key will be updated in the swarm base and is used as a symmetric key to encrypt and decrypt communications received and emitted within the group of UAVs. If an attacker gained access to the group key, communication would be compromised. While the group key will be updated, the ants that have not benefited from the update



(a) One-path configuration

(b) Three-path configuration

Fig. 2: Circular variable principle

yet (since they have not come back to the base yet) will not be able to read the new public path, thus essentially dividing the population in two. We consider two update plans: a time-based update and a population-based update. In the first one, after a predetermined time, the group key is updated for each ant that comes back to the base. In this case, if the period of that update is too short, the efficiency of the swarm may drop. The second method is to count the number of ants that have been updated with the new group key and artificially push the timer until a percentage of the population is updated. The first method prioritizes security, while the second puts a limit on efficiency loss.

B. Second Line of Defense

In the second scenario, we consider the case where the attacker would have had access to both the private ID and the group key. This situation is harmful to the security of the colony because if the attacker succeeded without being spotted and if a UAV with a stolen private ID reached the colony, it would gain access to communication and could operate within the swarm freely. We suggest the following communication protocol to prevent the attackers from inserting a UAV into the swarm, even if they had a perfect copy of the stolen UAV. When in operation, the UAVs are grouped by sections of n UAVs in which all hold a predetermine random variable at initialization of the mission stored in the internal memory that we will call the circular variable C_v^i, where i is the index of the UAV. Within the section, the communication follows a circular shape where a UAV receives a message from the previous p UAVs and then sends that message to the next p UAVs. The proposed message generation protocol is shown in Algorithm 1. The message sent through is composed of two components: (1) the encrypted message consisting of the public ID of the UAV, its coordinates, and its current task (using the group key with symmetric low-cost encryption), and (2) a fragment of the circular variable C_v^i. The message is sent through a Bluetooth radio signal so that any UAV can hear it, but only the predesignated observer UAV would be monitoring those messages. Thanks to this method, the UAV that emits the message is not aware of which UAVs are monitoring it and therefore cannot target them to break from the surveillance. Every time the UAVs send data, they use a fragment of the circular variable. A UAV holds in its memory the C_v used to broadcast its messages and an additional number p of C_v corresponding to the amount of UAV monitored. In fig. 2a, two C_v are held in D0, C_v^{D0} for sending messages and C_v^{D6} to check synchronization. Since the variable is randomly generated, it allows the message to have a unique mark that only the receiver of the message can authenticate. However the length of the fragment added to the message is common among the group, this is levered to assess the time frame at which it was sent, assuring synchronization. The number of paths p needs to be odd in order to allow democratic decision-making by the base after collecting data from this monitoring process. If the attacker attempts to insert a UAV with a stolen ID, they would need to correctly guess the segment of the circular variable for this iteration of the communication between the UAVs. Because the variables are computed before the flight and not on the UAV itself, it only costs comparison checking, which takes minimal computation. The size needed for the circular variable is also flexible. The bit length of the circular variable is as follows:

$$|C_v^i| \rightarrow \text{length} \times \text{update rate} \times \text{time} \qquad (1)$$

Algorithm 1: Message Generation Protocol

```
Input: $C_v^i$: String representing randomly generated value used
       for the process. $length$: Length of each fragment of $C_v$
       to be added to the messages. $M_c$: Encrypted message. $p$:
       Number of paths that will be used. $t$: Time iteration.
Output: Message: List of generate messages
Message ← [];
for $i$ in $range(0, p)$ do
    start_index ← $t \times p \times length + i \times length$;
    end_index ← $t \times p \times length + i \times length + length - 1$;
    fragment ← $C_v^i[start\_index : end\_index]$;
    M ← fragment + $M_c$;
    Message.append(M);
end
return Message;
```

Fig. 2a shows a simplified version of the problem where the number of paths p is one. Suppose a message $M_{6->0}$ is sent from UAV D6 to UAV D0. UAV D0 needs to assess whether the message has been sent from an insider. D0 finds C_v^{D6}, the circular variable assigned to D6, in its memory, and then looks at the fragment of the circular variable that corresponds to the current time frame and compares it to the one in front of the received message. If they match, the data is valid and is sent from the expected UAV within the group. Fig. 2b shows another example where the number of paths p is three. In the figure, only $M_{0->1}$ and $M_{0->2}$ are dissected, but the process is the same for $M_{0->3}$. D0 holds its circular variable C_v^{D0} with 3-bit long fragments. At time t, Algorithm 1 will extract the right fragment from C_v^{D0} by iterating through the number of paths and taking a part of $length$ to add in front of M_c. Here, we have set $length = 3$, so the first fragment is *001* for $M_{0->1}$, *101* for $M_{0->2}$, and *011* for $M_{0->3}$. Then, on the next time frame, Algorithm 1 will extract the next fragments because they can only be used once. This is why it takes a variable t as an argument to be able to jump to the right segment of the circular variable.

C. Third Line of Defense

In the third scenario, we consider an attacker who has access to the group key, private ID, and the circular variable. To verify the information, the emitter should be monitored by observer UAVs using Bluetooth and signal power monitoring. We aim to implement a Mobile-to-Mobile Localization (MtML) method that enables UAVs to monitor each other and increase security in real-life applications. To assess its feasibility, we propose using the fingerprinting-based technique proposed in [24], involving dedicated Receiving Stations (RS) within an estimated radius of the Mobile Station (MS). In our case, the RS would be other UAVs within the group. This method allows the RS to detect and process signals from the MS without synchronization, providing localization without the cooperation of other nearby devices or a fixed base station. To make the MtML method fully operational, UAVs need to be able to locate themselves to have a reference for analyzing others displacements. In urban environments, there is a high probability of mixed Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) situations. Therefore, we utilize a similar approach that is used in [25] that addresses robust cooperative localization utilizing Time-of-Arrival (ToA) and Angle-of-Arrival (AoA) measurements.

Assuming a reliable MtML method, Fig. 3 depicts the UAV monitoring protocol using a combination of a self-localization method and a behavior analyses protocol. The self-localization method could be done multiple ways ; CNN-based methods have been brought forward [26], [27], however the computation cost of such methods could be high so, in order to aim for a minimal impact on weight and performance, we will take the assumption of a GPS-based self-localization method. By employing ToA/AoA to assess the tarhet UAV position based on its communication signals [28], [29], the monitoring UAV compares its observed trajectory with the expected one from Section III-B. If the targeted UAV significantly deviates, it's flagged as rogue and reported. Algorithm 2 is designed to monitor the positions of a set of UAVs in real-time and is running on each UAVs. The protocol takes as input the set of UAVs to monitor, the set of public targets (the public tasks of UAVs), and the current time iteration. For each UAV, it calculates the expected trajectory based on the public targets and the observed trajectory based on the current time iteration. The protocol then checks if the observed trajectory deviates from the expected trajectory, using a threshold value to determine whether the deviation is significant. If the deviation is too large, the UAV is flagged as rogue and added to a set of rogue UAVs. The protocol returns a set of rogue UAVs, which can be used to take appropriate action. The novelty of our solution resides in this principle to secure the task's accomplishment by comparing three pieces of information: the expected behavior known to all at initialization, the statement made by the surveillance UAV continuously during operation, and the monitoring of the behavior.

Algorithm 2: Position Monitoring Protocol

```
Input: $D$: Set of UAVs to survey, $T$: Set of public targets, $t$:
      Time iteration
Output: $R$: Set of rogue UAVs
$R \leftarrow \emptyset$;
for $i$ in range($0, |D|$) do
    // Calculate expected trajectory of UAV $i$
    $E_i \leftarrow$ expected_trajectory($D[i]$, $T$);
    // Calculate observed trajectory of UAV $i$
    $O_i \leftarrow$ observed_trajectory($D[i]$, $t$);
    // Check if observed trajectory deviates
       too much from expected trajectory
    if deviation($E_i$, $O_i$) $>$ threshold then
        $R \leftarrow R \cup \{D[i]\}$;
    end
end
return $R$;
```

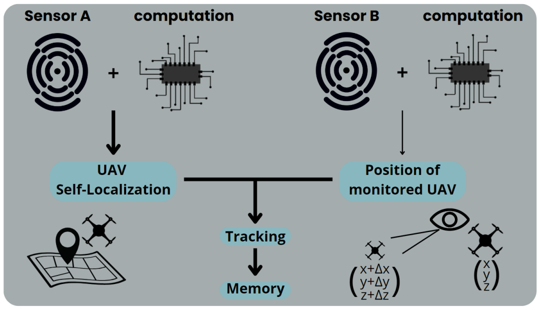
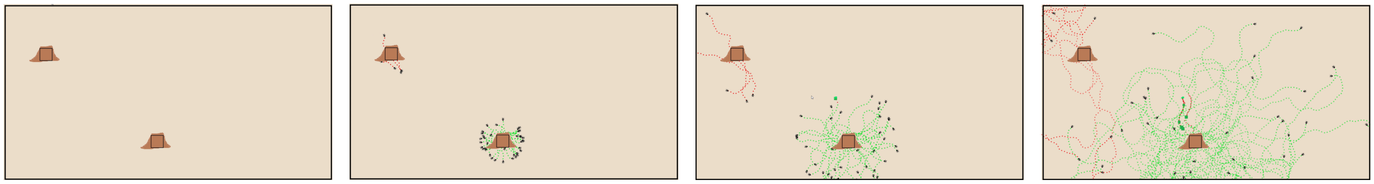


Fig. 3: Behavior monitoring based on localization

Now, the attacker has two choices: rebel against the swarm or become a slave. If the attacker does not follow the task and behaves unexpectedly, the observers will catch it and report it to the base upon return. If many reports point to misbehavior by one of the UAVs, it will be disabled and banned during operations. The data collected from the MtML method can be used to increase security in real-life applications. For example, in a search and rescue mission, the UAVs can use the MtML method to locate and track each other, ensuring that all UAVs are accounted for and operating within the designated area. If a UAV deviates from the expected path or behaves unexpectedly, the other UAVs can immediately detect and report it, allowing for a quick response to potential threats. The decentralized structure mitigates the impact of potential UAV hijacks. Should an attacker breach the initial defenses and commandeer a UAV, the swarm's functionality remains largely unaffected. Compliant behavior from the hijacked UAV would see misinformation attempts nullified by cross-verification from other UAVs. Non-compliant behavior triggers protective measures, such as retreating to a safe zone. Additionally, the MtML method can be used to monitor the health and status of each UAV, enabling preventive maintenance and reducing the risk of equipment failure during the mission.

IV. Experimental results

To visualize the application of our solution, we have developed a simulator using the Python interpreter and the Pygame library that mimics the behavior of an ant colony, with a focus on the communication security methods discussed. We ran it on an 11th Gen Intel Core i7 with a 12 GB RAM processor and the Windows 11 OS. The values of the private IDs and the group key are chosen by the user upon initialization of the simulation. To allow a better visualization of which key is used and by whom, the keys are converted to a tuple in 255-color format. Fig. 4 shows one run of the simulation with one peaceful colony and one rogue colony.



(a) Initialization of the anthills with random position

(b) Ants leaving their hill with the behavior *Random Walk*; green pheromones are allies ants and red pheromones are for the rogue ants

(c) Food appears in the simulation, and ants can capture it and change behavior to *Go Back Home*

(d) Food is brought back to the anthill

Fig. 4: The developed simulator for ant-based swarm UAV communication

TABLE I: Communication attacks on swarm UAVs

Attack Category	SACRED EYE Protection
Listening Attacks	
Spoofing	✓
Man-in-the-Middle	✓
Inserting Attacks	
Noise Interference	✗
Data Insertion	✓
Physical Attacks	
Reverse Engineering	✓
Component Derivation	✓

A. *Security Evaluation*

Regarding the security of the swarm UAVs, they can face three types of communication attacks [30]. As shown in Table I the three lines of defenses in **SACRED EYE** can prevent the mentioned attacks except for the noise interference. Attacks during the operation are tackled thanks to the proposed methods. The circular variable prevents any entity from joining the UAV section after it has engaged secured communication, even if the attacker has access to the group key. On the other hand, the behavior analysis prevents the mission from being compromised if the communication were to be corrupted.

In addition, we created a program to run a simulation in which 20 UAVs are controlled by a common known behavior, except for a rogue UAV that will behave in an unexpected way. The purpose of the code is to demonstrate the pace at which rogue UAVs are detected. Fig. 5 shows the Average Relative Distance (ARD) (no units) of a UAV from its expected position with different configurations of Standard Deviation (SD) on the noise applied for 1000 cycles in the simulation. Each UAV computes the next coordinates expected of its neighbor based on its current position, and then calculates the distance between their real and expected coordinates. We have progressively increased the noise applied to the recording of the position to see the effect on the accuracy of the prediction and found out that after crossing an SD superior to 0.1, it is no longer possible to reveal the identity of the rogue UAV. By considering higher-order derivatives and multiple steps in the trajectory, we could detect more subtle changes in behavior and identify longer-term trends.

B. *Overhead Evaluation*

Using the "Route Validation in MANETs using MLP" add-on in Matlab, we tested different sizes of UAV groups to get a glimpse of the ideal size for the UAV subgroups shown in Fig. 6. For each node, we sent 50 packets to the sink node with a random data rate between 1kb and 100kb. The position is random, as is the trajectory of the UAV while the evaluation is made. As can be seen, groups of 50 UAVs seem to be more efficient than others in terms of time and energy cost. Then, using the NS3 simulator [31] with OLSR and a simple UDP protocol, we simulated data flow rates per node in increasingly dense UAV groups. Results shown in Fig. 7 indicate that excessive UAV density leads to traffic overhead and delays, highlighting the need for optimal group sizes to minimize latency.

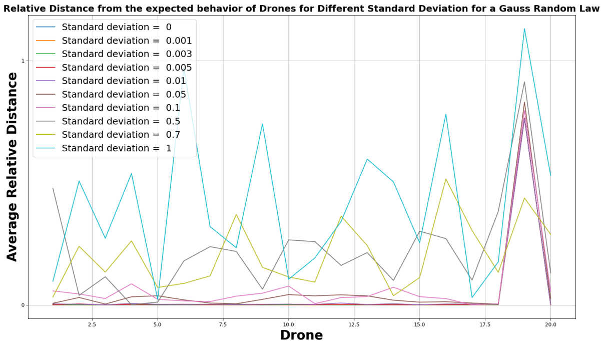


Fig. 5: ARD from the expected behavior per UAV and per SD for a 1000-cycle simulation

In addition, we examined the overhead for the following elements: private ID, group key, and circular variable.

Private ID: The private ID is a bit-type value stored in the UAV hardware. Its impact would be solely in terms of memory storage on the individual UAV and the amount of memory derived from the size of the ID. The ID could be as short as $log_2(n)$ with n the number of UAVs in the IoD.

Group Key: The average energy cost per byte of various encryption methods varies depending on the payload size [32]. For a payload size of 4 bytes, the energy cost is 1.2 µJ per byte, while for larger payload sizes (40, 400, 2000 bytes), the energy cost per byte

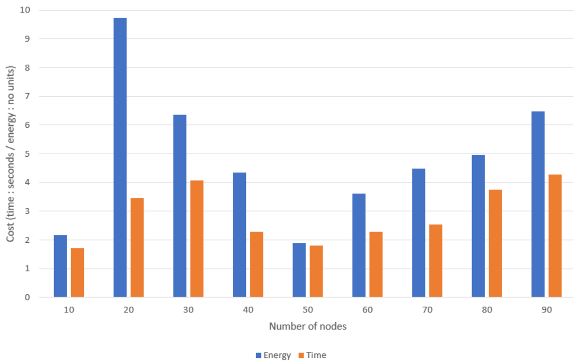


Fig. 6: Time (s) and energy (no units) cost per number of UAVs

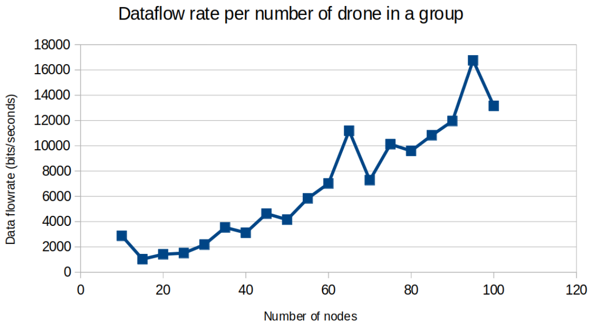


Fig. 7: Dataflow rate per UAV in groups of different size

is negligible. However, when a data authentication protocol is added, the energy consumption increases significantly for short messages. The energy cost for a 4-byte message with authentication can rise to a maximum of 8.5 µJ per byte. In our case, it is important to set a minimum segment size to ensure that the authentication process is energy-efficient.

Circular Variable: The circular variable is stored in the internal memory of the UAVs, and its weight is defined by Equation (1). Additionally, there will be a computation overhead to run the algorithm 1 that needs to be executed each time a message is sent or received. The most expensive operation is the string slicing and concatenation, which are $O(\text{length}$ and $O(\text{length} + |M_c|)$ respectively. Combining these, the total time complexity for each iteration is $O(\text{length}+|M_c|)$.

V. Conclusion

In this paper, we introduced a novel amalgamation of security measures called **SACRED EYE** to ensure secure communication and safeguard the successful completion of missions within the realm of UAVs. Our approach extended beyond securing communication channels to incorporate distributed behavioral monitoring, crucial for task success within mission parameters.

References

[1] Seeja G, Arockia Selvakumar A, and Berlin Hency V. A survey on swarm robotic modeling, analysis and hardware architecture. *Procedia Computer Science*, 133:478–485, 2018. International Conference on Robotics and Smart Manufacturing (RoSMa2018).

[2] T. Filkin, N. Sliusar, M. Ritzkowski, and M. Huber-Humer. Unmanned aerial vehicles for operational monitoring of landfills. 2021.

[3] Krul S., Pantos C., Frangulea M., and Valente J. Visual slam for indoor livestock and farming using a small drone with a monocular camera: A feasibility study. 2021.

[4] Mohamed Abdelkader, Samet Güler, Hassan Jaleel, and Jeff S. Shamma. Aerial swarms: Recent applications and challenges. *Current Robotics Reports*, 2:309–320, 2021.

[5] D Michailidis, E.T.; Vouyioukas. A review on software-based and hardware-based authentication mechanisms for the internet of drones. In *Drones*, volume 6, 2022.

[6] Ashutosh Singandhupe, Hung Manh La, and David Feil-Seifer. Reliable security algorithm for drones using individual characteristics from an eeg signal. *IEEE Access*, 6:22976–22986, 2018.

[7] Saeed Ullah Jan, Irshad Ahmed Abbasi, and Fahad Algarni. A key agreement scheme for iod deployment civilian drone. *IEEE Access*, 9:149311–149321, 2021.

[8] Sajid Hussain, Shehzad Ashraf Chaudhry, Osama Ahmad Alomari, Mohammed H. Alsharif, Muhammad Khurram Khan, and Neeraj Kumar. Amassing the security: An ecc-based authentication scheme for internet of drones. *IEEE Systems Journal*, 15(3):4431–4438, 2021.

[9] Zaiba Shah, Umer Javed, Muhammad Naeem, Sherali Zeadally, and Waleed Ejaz. Mobile edge computing (mec)-enabled uav placement and computation efficiency maximization in disaster scenario. *IEEE Transactions on Vehicular Technology*, 72(10):13406–13416, 2023.

[10] Ursula Challita, Aidin Ferdowsi, Mingzhe Chen, and Walid Saad. Machine learning for wireless connectivity and security of cellular-connected uavs. *IEEE Wireless Communications*, 26(1):28–35, 2019.

[11] Tarun Rana, Achyut Shankar, Mohd Kamran Sultan, Rizwan Patan, and Balamurugan Balusamy. An intelligent approach for uav and drone privacy security using blockchain methodology. In *International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 162–167, 2019.

[12] Di Lu, Ruidong Han, Yulong Shen, Xuewen Dong, Jianfeng Ma, Xiaojiang Du, and Mohsen Guizani. xtseh: A trusted platform module sharing scheme towards smart iot-ehealth devices. *IEEE Journal on Selected Areas in Communications*, 39(2):370–383, 2021.

[13] Dominic Pirker, Thomas Fischer, Christian Lesjak, and Christian Steger. Global and secured uav authentication system based on hardware-security. In *IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 84–89, 2020.

[14] Reliable puf-based mutual authentication protocol for uavs towards multi-domain environment. *Computer Networks*, 218, 2022.

[15] Li Zhang, Jianbo Xu, Mohammad Obaidat, Xiong Li, and P. Vijayakumar. A puf-based lightweight authentication and key agreement protocol for smart uav networks. *IET Communications*, 16, 11 2021.

[16] Tejasvi Alladi, Vinay Chamola, Naren Naren, and Neeraj Kumar. Parth: A two-stage lightweight mutual authentication protocol for uav surveillance networks. *Computer Communications*, 160, 05 2020.

[17] Dominic Pirker, Thomas Fischer, Christian Lesjak, and Christian Steger. Global and secured uav authentication system based on hardware-security. 2020.

[18] Aiguo Chen, Kun Peng, Zexin Sha, Xincen Zhou, Zhen Yang, and Guoming Lu. Toam: a task-oriented authentication model for uavs based on blockchain. 2021.

[19] Maninderpal Singh, Gagangeet Singh Aujla, and Rasmeet Singh Bali. Odob: One drone one block-based lightweight blockchain architecture for internet of drones. pages 249–254, 2020.

[20] R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in a swarm-bot. In K. Murase, K. Sekiyama, N. Kubota, T. Naniwa, and J. Sitte, editors, *Proc. of the 3rd Int. Symp. on Autonomous Minirobots for Research and Edutainment, AMiRE 2005*, pages 314–322, Berlin, 2006. Springer Verlag.

[21] Julien Dupeyroux, Julien Serres, and Stéphane Viollet. Antbot: A six-legged walking robot able to home like desert ants in outdoor environments. *Science Robotics*, 4:eaau0307, 02 2019.

[22] Andrea Alma, Micaela Buteler, A. Martínez, and Juan Corley. Wind disrupts trail pheromone communication in the leaf-cutting ant acromyrmex lobicornis. *Animal Behaviour*, 192:39–49, 10 2022.

[23] Wei Xiang, Jiaping Ren, Kuan Wang, Zhigang Deng, and Xiaogang Jin. Biologically inspired ant colony simulation. *Computer Animation and Virtual Worlds*, 30(5):e1867, 2019. e1867 cav.1867.

[24] Isabelle Vin, Davy P. Gaillot, Pierre Laly, Martine Liénard, and Pierre Degauque. Overview of mobile localization techniques and performances of a novel fingerprinting-based method. *Comptes Rendus Physique*, 16(9):862–873, 2015.

[25] Behailu Yohannes Shikur and Tobias Weber. Robust cooperative localization in mixed los and nlos environments using toa. pages 1–6, 03 2014.

[26] José Cocoma-Ortega and Jose Martinez-Carranza. A cnn-based drone localisation approach for autonomous drone racing. 09 2019.

[27] Z. Gao, D. Li, G. Wen, Y. Kuai, and R. Chen. Drone based rgbt tracking with dual-feature aggregation network. *Drones*, 7:585, 2023.

[28] Grigoriy Fokin and Grigoriy Fokin. Aoa measurement processing for positioning using unmanned aerial vehicles. *International Black Sea Conference on Communications and Networking*, 2019.

[29] Michael A. Magers and Michael A. Magers. Geolocation of rf emitters using a low-cost uav-based approach. *null*, 2016.

[30] Chaitanya Rani, Hamidreza Modares, Raghavendra Sriram, Dariusz Mikulski, and Frank Lewis. Security of unmanned aerial vehicle systems against cyber-physical attacks. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 13, 11 2015.

[31] ns-3 Consortium. *ns-3 Network Simulator*, 2024. https://www.nsnam.org.

[32] Marc Ohm, Lars Taufenbach, Karsten Weber, and Timo Pohl. Power consumption of common symmetric encryption algorithms on low-cost microchips. 10 2023.