Short: Rethinking Secure Pairing in Drone Swarms

Muslum Ozgur Ozmen*, Habiba Farrukh*, Hyungsub Kim, Antonio Bianchi, and Z. Berkay Celik
Purdue University
{mozmen, hfarrukh, kim2956, antoniob, zcelik}@purdue.edu

***Abstract*—Drone swarms are becoming increasingly prevalent in important missions, including military operations, rescue tasks, environmental monitoring, and disaster recovery. Member drones coordinate with each other to efficiently and effectively accomplish a given mission. To automatically coordinate a swarm, member drones exchange critical messages (e.g., their positions, locations of identified obstacles, and detected search targets) about their observed environment and missions over wireless communication channels. Therefore, swarms need a pairing system to establish secure communication channels that protect the confidentiality and integrity of the messages. However, swarm properties and the open physical environment in which they operate bring unique challenges in establishing cryptographic keys between drones.**

In this paper, we first outline an adversarial model and the ideal design requirements for secure pairing in drone swarms. We then survey existing human-in-the-loop-based, context-based, and public key cryptography (PKC) based pairing methods to explore their feasibility in drone swarms. Our exploration, unfortunately, shows that existing techniques fail to fully meet the unique requirements of drone swarms. Thus, we propose research directions that can meet these requirements for secure, energy-efficient, and scalable swarm pairing systems.

I. Introduction

Drone swarms are groups of drones that coordinate to perform critical missions, including but not limited to military operations, search-and-rescue tasks, environmental monitoring, and disaster recovery [4], [16]. Swarms leverage continuous sensing and communication to achieve localization, navigation, and obstacle avoidance, which allow them to accomplish tasks that may be challenging for a single drone. To automatically coordinate a swarm, member drones exchange critical messages (e.g., their positions, locations of identified obstacles, detected search targets) about their observed environment and missions over wireless communication channels.

The critical nature of missions performed by drone swarms makes them an attractive target for adversaries. The adversarial threats are further exacerbated by (i) the isolated areas in which swarms typically operate and (ii) the long delay or disconnection they experience in communicating with ground control stations (GCSs). Particularly, an adversary can eavesdrop on the drone communication and inject fake messages to disrupt the swarm's operation. For instance, an adversary may eavesdrop on a search target's location during a swarm's search-and-rescue operation and misguide the drones to the wrong location.

Therefore, swarms need secure communication channels to protect the critical information exchanged and the integrity of the swarm's mission. To establish such secure communication channels, swarms require a pairing mechanism that establishes cryptographic keys among the member drones.

Several prior works have investigated pairing schemes for popular computing platforms, including smartphones [19], [26], AR/VR headsets [29], [30], and IoT devices [7], [13], [17]. These works can be grouped into two main categories: (1) human-in-the-loop-based and (2) context-based pairing methods. Human-in-the-loop-based methods require device users to physically interact with the devices (e.g., type passwords, scan QR codes, perform gestures) for pairing. Such approaches have been adopted to drone communication protocols (e.g., MAVLink [18]), where users hard-code the keys into drones for secure communication between the GCS and drone. These approaches, however, suffer from scalability and usability issues with an increasing number of devices. In contrast, context-based pairing methods leverage devices' shared context (e.g., location, observed event timings, system status) to derive cryptographic keys. While such methods offer better scalability and usability, they assume the devices are present within a physical boundary.

Another line of work has proposed public key cryptography (PKC) based techniques to establish keys between devices [1], [6]. Besides traditional public key infrastructure (PKI) based schemes, recent works propose identity-based and certificateless techniques that alleviate the need for public key certificates [12], [23], [27], [32]. These works leverage a trusted third party (e.g., GCS) to issue implicitly certified private/public keys. Yet, they rely on costly PKC operations and, therefore, suffer from high computational overhead and battery consumption.

Unfortunately, applying existing methods directly for pairing drone swarms is infeasible due to the following reasons:

1) In many swarm usage scenarios, an adversary can easily steal a drone in the swarm since it operates in environments where no physical security is enforced. Without proper protection against physical attacks, an adversary can gain access to the cryptographic keys stored in the stolen drone.
2) Member drones in a swarm often change dynamically, including adding new drones to the swarm and removing faulty and crashed drones.
3) Swarms have limited hardware resources (e.g., CPU, memory, and battery), which require pairing to be low-cost and energy-efficient. Thus, swarms require a secure and energy-efficient pairing mechanism to establish secure communication channels.

*Muslum Ozgur Ozmen and Habiba Farrukh contributed equally.

Symposium on Vehicles Security and Privacy (VehicleSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-88-6
https://dx.doi.org/10.14722/vehiclesec.2023.23076
www.ndss-symposium.org

In this paper, we explore the security and design requirements for effective swarm pairing and propose research directions for designing swarm pairing methods that satisfy the unique needs of swarms. First, we highlight that, due to the lack of physical protection, a swarm pairing system must offer protection against stolen credential attacks, where the attacker captures a crashed/landed drone on the ground to steal its keys, as well as traditional network attacks such as MitM. Protecting swarms against stolen credentials requires developing either a low-cost and energy-efficient secure hardware design for key storage in drones or techniques to identify the failed drones to revoke their keys. Second, we emphasize that swarms' continuous sensing capabilities can be leveraged for context-based secure pairing. However, designing context-based systems for swarm pairing requires the identification of sensors that can extract sufficient entropy from environmental conditions (e.g., wind gusts, air pressure) and support continuous authentication between drones to prevent stolen credential attacks. Lastly, we explore the feasibility of identity-based and certificateless cryptographic techniques for swarm pairing. We find that these techniques are also vulnerable to stolen credential attacks, and therefore, they require methods to protect against such attacks.

In summary, we make the following contributions:

- We outline the ideal security and design requirements of a secure pairing system for drone swarms.
- We show that prior human-in-the-loop-based, context-based, and PKC-based pairing techniques are not sufficient to meet the requirements of drone swarms.
- We provide future research directions that can offer secure, energy-efficient, and scalable pairing for drone swarms.

II. Background

A. Background

A drone swarm is a set of aerial robots that coordinate to achieve a given mission [4]. Such coordination can be achieved by two different types of control algorithms: (i) centralized and (ii) decentralized. In centralized swarms, a ground control station (GCS) sends control command messages to the drones, either individually or through a leader drone. In decentralized swarms, the drones do not rely on a GCS and communicate with each other to autonomously determine their control commands (e.g., changing their positions).

To navigate through 3D space and conduct their missions without colliding with each other or static/dynamic objects, swarms rely on their sensing and communication units for coordination. First, swarms can use external sensors such as real-time kinematic GNSS and motion capture to track the drones' positions. However, this necessitates placing such sensors in the swarm's operation area beforehand, which can be impractical for missions that cover large areas. Second, drones leverage onboard sensors (e.g., cameras, LiDARs, optical flow sensors, GNSS receivers, gyroscopes, magnetometers, and barometers) to operate in any environment. Drones use these sensors to localize themselves, other members of the swarm, and obstacles in the environment. Lastly, drones communicate with each other to (1) notify their position, (2) inform other drones about obstacles, (3) alert when a target is detected (e.g., in search and rescue missions), and (4) bid for sub-goals (e.g., while determining which drone will search an area subsection).

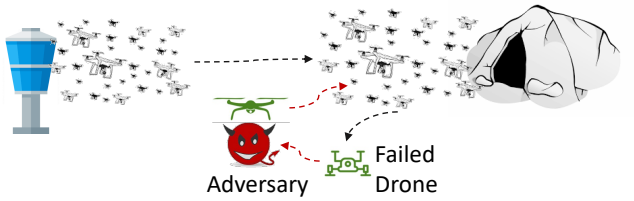

Fig. 1: Overview of our adversarial model.

Problem Statement. Drone swarms require secure wireless communication channels to protect the confidentiality and integrity of the messages they exchange. This protection is critical to prevent various attacks (e.g., MitM and fake message injection), provide privacy for the swarm missions, and ensure the trustworthiness of the swarms. Swarms, therefore, require a pairing mechanism to establish cryptographic keys between drones to enable secure wireless communication.

However, designing a pairing system to establish cryptographic keys between drones in a swarm is a challenging task. First, swarms typically operate in environments where no physical protection is enforced. An adversary can leverage the lack of physical protection to pair with the member drones in a swarm (detailed in Section II-B). Second, new drones may be introduced to the swarm or existing drones may be removed (e.g., due to crashes or low battery). Lastly, drones in the swarm may have computation, memory, and battery consumption constraints.

B. Adversarial Model and Assumptions

The attacker ($\mathcal{A}$) aims to eavesdrop on the communication between the drones in a swarm (e.g., to learn the location of targets in a search-and-rescue operation) and inject messages that disrupt their operation (e.g., by misguiding the drones to wrong locations and spoofing fake obstacles). We assume that $\mathcal{A}$ has complete knowledge of the pairing protocol and has access to the communication channels.

We detail the attacks $\mathcal{A}$ can conduct as follows.

1) $\mathcal{A}$ intercepts the messages between member drones in the swarm and conducts a MitM attack to pair with them.
2) $\mathcal{A}$ leverages a malicious drone to join the swarm and pair with member drones.
3) $\mathcal{A}$ steals a crashed/landed drone on the ground to extract its credentials, such as its ID and cryptographic keys (see Figure 1). Swarms usually consist of hundreds of drones that operate in remote areas. While traditional computing systems are normally used in locations where some form of physical security is enforceable, drones often operate in "open" physical environments. For this reason, they do not have any physical protection, and they are at risk of getting stolen by attackers.

C. Design Requirements

We now detail three key requirements for secure pairing in drone swarms. Later, we will leverage these requirements to

TABLE I: Overview of existing pairing schemes.

Pairing Solution	R1a	R1b	R1c	R2	R3
Human-in-the-loop-based	●	●	○	●	◐
Context-based	●	○	○	●	○
PKC-based	●	●	○	○	◐

assess the existing techniques and guide future research and practice in this direction.

R1: Security. Unlike traditional computing systems, swarms operate in open environments without any physical security, allowing adversaries to steal credentials from landed or crashed drones and leverage a malicious drone to pair with member drones. Thus, swarm pairing systems, as detailed in Section II-B, must offer security against **(a)** traditional network-level attacks (e.g., MitM), **(b)** adversarial drones, **(c)** and stolen credentials.

R2: Energy Efficiency and Scalability. An energy-efficient pairing system is required for swarms since energy efficiency directly translates into longer flight times for drones. Especially small drones, which are commonly used in swarms, may have stringent battery constraints, and an inefficient pairing system may significantly limit their flight times. Additionally, the pairing system must be scalable, where it remains energy-efficient even if the swarm includes a large number of drones.

R3: Drone Addition and Removal. The pairing system must support drone additions, where the existing drones in the swarm establish keys with the new ones, and drone removals, where the swarm revokes the keys of the crashed drones. This is because, in many applications, the GCS may deploy additional drones to help the swarm complete its mission. The newly deployed drones must pair with the existing drones in the swarm to securely communicate and coordinate with them. Additionally, there must exist a mechanism to remove a drone from the swarm if it has been captured by an adversary and its cryptographic material has been compromised.

III. Pairing Solutions

To enable secure communication between member drones in a swarm, we consider using existing (i) context-based pairing, (ii) human-in-the-loop pairing methods, and (iii) PKC-based techniques. We discuss their adaptability for drone swarms by analyzing whether they satisfy the three key design requirements outlined in Section II-C (See Table I).

A. Human-in-the-Loop-based Pairing

Existing Methods. Human-in-the-loop-based pairing approaches require users to interact with the devices physically. For instance, a line of work relies on users to enter passwords, scan QR codes, or press buttons on IoT devices for pairing [3], [22]. Another line of work requires users to perform similar gestures on IoT devices to initiate pairing (e.g., shake two devices at the same time) [17], [19].

Adapting to Drone Swarms. Existing human-in-the-loop-based pairing methods are unfortunately not suitable for drone swarms due to two main reasons: (1) these methods suffer from scalability and usability issues as performing gestures is time-consuming (**R2**), and (2) attackers can be present within the same physical environment and observe or get involved in the pairing processes (**R1b**).

A possible human-in-the-loop-based solution to swarm pairing could be physically hard-coding keys to each drone from a ground control station before dispatching them. This would ensure energy efficiency and scalability, prevent an adversarial drone to eavesdrop on the keys, and allow drone additions to the swarm, satisfying **R1b**, **R2** and partially satisfying **R3**.

However, the key limitation of this approach is its vulnerability to stolen credentials. An adversary who steals a crashed/landed drone can learn the key and use it to eavesdrop on the communication, decrypt the messages sent before capturing the drone (if a forward-secure encryption scheme is not implemented), and inject fake messages to disrupt the swarm's operations. To address stolen credentials attacks, one may consider using secure hardware to store the cryptographic keys in drones. For example, STM32 microcontrollers are equipped with readout protection (RDP) to prevent stealing data from flash memory [25], [21]. Yet, open-source microcontroller systems (e.g., Pixhawk series [24]) do not activate RDP by default. Moreover, resourceful attackers can leverage side channels to extract the cryptographic keys [10], [11].

Research Directions: The main limitation of integrating human-in-the-loop-based pairing approaches into drone swarms is their vulnerability to stolen credential attacks. To address this, future research could investigate low-cost and energy-efficient secure hardware designs for key storage to prevent an adversary from extracting the keys of a captured drone. Such designs must be scalable to a large number of drones and resistant to physical side-channel attacks.

B. Context-based Pairing

Existing Methods. In context-based pairing methods, devices leverage their shared context (e.g., location, time, and system status) to derive cryptographic keys. In one line of work, co-located devices in an environment rely on on-board sensors to extract entropy from common events occurring in their surroundings and use it as evidence of co-presence to bootstrap key establishment protocols [7], [13]. For instance, two microphones in the same environment record the sound of a door-open event and may use it as evidence of co-presence to establish symmetric keys [20]. Recent works have also leveraged event timings of common events observed by heterogeneous devices as evidence to derive secure keys [7], [8], [13]. Another line of work has leveraged wireless localization techniques to authenticate all devices located within a specific physical range [9]. These methods rely on wireless antennas on devices to extract precise device locations and verify if the device is located within a predefined distance boundary.

Adapting to Drone Swarms. As drones employ a variety of sensors to continuously sense their physical environment, context-based pairing methods have the potential to enable secure pairing for swarms. Yet, unlike IoT environments, drones in a swarm are not restricted by a physical boundary (e.g., walls, doors) which makes defending against malicious drones in the swarm's surroundings challenging, hence violating **R1b**.

Although a physical boundary does not bind a swarm, drones observe similar environmental conditions (e.g., wind gusts, air pressure, birds flying nearby) during flight. Similar to the IoT device pairing methods, drones in a swarm can use the observed

environmental conditions over the course of a flight or operation as a shared secret to establish secure cryptographic keys.

However, leveraging context-based methods for pairing drones in a swarm involves several challenges. First, a swarm's flight patterns and the number of drones in it may cause member drones to fly far from each other, resulting in different environmental contexts observed by them. For instance, SocraticSwarm [14] and Sciadro [5] are decentralized swarm control algorithms where drones search different areas in the environment. Second, an adversarial drone flying near a swarm at any time instance may record a similar environmental context as the legitimate drones and attempt to use it for pairing. Third, to revoke the keys of crashed drones, the drones must continuously authenticate each other during flight through their shared context and evolve their cryptographic keys. Lastly, new drones joining a swarm need to provide evidence of shared environmental context to pair with the existing drones. However, the new drones do not have the same context accumulated in the swarm through time (violating **R3**).

Research Directions: For context-based pairing in swarms, future research could first identify the sensors that can provide a proof-of-co-presence for drones. The sensors used for pairing must (1) provide sufficient entropy (not predictable by attackers) and (2) acquire similar, compatible readings in all the drones within a certain distance (the swarm's operation area).

Continuous authentication and key evolvement schemes could be additionally devised to ensure the keys of crashed drones are revoked, and an adversary cannot use an adversarial drone to observe the same context.

Lastly, behavior-based authentication and attestation techniques could be developed to assess whether drones attempting to join the swarm are benign or malicious. To this aim, these techniques can leverage the physical and network behavior of a drone to fingerprint and authenticate its control software before allowing the drone to join the swarm.

C. Public Key Cryptography Based Techniques

Existing Methods. Key establishment is well-studied in public key cryptography (PKC) [6]. For instance, group key establishment schemes have been proposed to derive a single key between multiple devices for secure communication [31]. Yet, traditional PKC-based key establishment schemes require a public key infrastructure (e.g., certificate authorities) to ensure the validity of the public keys and prevent MitM attacks [1]. Integrating a public key infrastructure to drone swarms may not be feasible due to the transmission and verification overhead introduced by long certificate chains [15], violating **R2**.

To address the limitations of PKI, identity-based and certificateless key establishment schemes have been proposed [2], [28]. These schemes rely on a Trusted Third Party (TTP) to provide private/public key pairs to devices that are implicitly certified by the TTP. Thus, they enable devices that have received their key pairs from the TTP to derive shared keys. These schemes prevent MitM attacks since any malicious device that does not have a key pair from the TTP cannot join the key establishment protocol.

Adapting to Drone Swarms. Identity-based and certificateless key establishment schemes are especially suitable to provide secure pairing in drone swarms. This is because the ground control station (GCS) can serve as the TTP and provide private/public key pairs to the drones. Here, we note that even in the case of a decentralized swarm, a GCS is still used to provide configurations and firmware to the swarm's members. However, in this case, the swarm does not rely on the GCS during its missions. Many recent works have leveraged identity-based and certificateless cryptography to propose key establishment schemes for drones [12], [23], [27], [32]. These systems support drone additions (**R3**) and offer security against MitM attacks and malicious drones (**R1**).

However, these approaches have several limitations. First, they are vulnerable to stolen credentials, where an adversary who captures a crashed drone can extract its implicitly-certified private key and use it for communicating with other drones (partially violating **R1**). Second, PKC operations (e.g., modulo exponentiation and elliptic curve scalar multiplication) are usually computationally expensive (e.g., two-three magnitudes slower than symmetric key cryptography operations [23], [33]). Additionally, key establishment protocols require a large number of PKC operations with the increasing number of drones in a swarm. Although such expensive computations can be considered as a one-time effort to derive shared keys, it may still be necessary to execute them multiple times if a key update is necessary (e.g., when the member drones in the swarm change). This violates the energy efficiency and scalability (**R2**) requirement of drone swarms, especially when the swarm includes a large number of drones.

Research Directions: To address the limitations of existing PKC-based cryptographic schemes, future research could develop techniques to protect against stolen private keys. For this, one direction is secure hardware (detailed in Section III-A), however, it requires altering the drone hardware, which may be infeasible.

With PKC, another direction is identifying the crashed drones and maintaining a deny list to prevent such drones from further participating in the communication. One approach to identifying crashed drones could be implementing an energy-efficient liveness check mechanism. Here, frequent liveness checks may hurt energy efficiency, whereas less frequent checks may allow an adversary to capture the crashed drone.

Future research could also study lightweight and scalable cryptographic techniques to minimize the energy overhead of pairing on drones. Here, energy-efficient post-quantum schemes could also be developed due to the emerging threat of quantum computers against traditional PKC.

IV. Conclusion & Recommendations

Drone swarms enable autonomy in military, civilian and industrial applications. For effective coordination in a swarm, there is a need to design secure and energy-efficient swarm pairing systems that establish cryptographic keys for secure communication between drones. Our analysis highlights that the lack of physical protection, limited resources, and energy constraints of swarms present unique security and design challenges for swarm pairing. Unfortunately, existing human-in-the-loop, context-based and PKC-based pairing solutions fail

to address these challenges effectively. Thus, enabling secure communication for swarms requires developing new methods that can defend against traditional security threats (e.g., MitM attacks) as well as attacks specific to swarms (e.g., stolen credential attacks and adversarial drones).

Our study finds two common pitfalls of existing pairing techniques: these techniques cannot protect against stolen credentials and they fail to support drone additions and removals to the swarm. To address these, future research could (1) design secure hardware solutions for key storage, which are resilient against side-channel attacks [10], [11], (2) devise continuous authentication and key evolvement protocols, or (3) implement energy-efficient liveness checks to maintain revocation lists.

Another natural future research direction is combining different pairing techniques. In fact, different techniques provide different advantages in implementing a secure pairing system for drone swarms. For instance, designing a context-based pairing scheme and integrating it with human-in-the-loop-based or PKC-based techniques can provide the continuous authentication required to protect against stolen credentials.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their comments and suggestions. This work has been partially supported by the National Science Foundation (NSF) under grant CNS-2144645 and Office of Naval Research (ONR) under grant N00014-20-1-2128. Any findings, conclusions and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or ONR.

REFERENCES

[1] C. Adams and S. Lloyd, *Understanding public-key infrastructure: concepts, standards, and deployment considerations*. Sams Publishing, 1999.

[2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*, 2003.

[3] M. K. Chong, R. Mayrhofer, and H. Gellersen, "A survey of user interaction for spontaneous device association," *ACM Computing Surveys (CSUR)*, 2014.

[4] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, 2018.

[5] M. G. Cimino, M. Lega, M. Monaco, and G. Vaglini, "Adaptive exploration of a uavs swarm for distributed targets detection and tracking." in *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2019.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," in *IEEE Transactions on Information Theory*, 1976.

[7] H. Farrukh, M. O. Ozmen, F. K. Ors, and Z. B. Celik, "One key to rule them all: Secure group pairing for heterogeneous IoT devices," in *IEEE Symposium on Security and Privacy (SP)*, 2023.

[8] M. Fomichev, J. Hesse, L. Almon, T. Lippert, J. Han, and M. Hollick, "Fastzip: Faster and more secure zero-interaction pairing," in *International Conference on Mobile Systems, Applications, and Services*, 2021.

[9] E. Gaebel, N. Zhang, W. Lou, and Y. T. Hou, "Looks good to me: Authentication for augmented reality," in *International Workshop on Trustworthy Embedded Devices*, 2016.

[10] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "Ecdsa key extraction from mobile devices via nonintrusive physical side channels," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

[11] D. Genkin, A. Shamir, and E. Tromer, "Rsa key extraction via low-bandwidth acoustic cryptanalysis," in *Annual cryptology conference*, 2014.

[12] H. Guo, T. Liu, K.-S. Lui, C. Danilov, and K. Nahrstedt, "Secure broadcast protocol for unmanned aerial vehicle swarms," in *IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2020.

[13] J. Han, A. J. Chung, M. K. Sinha, M. Harishankar, S. Pan, H. Y. Noh, P. Zhang, and P. Tague, "Do you feel what i hear? enabling autonomous IoT device pairing using different sensor types," in *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[14] P. Henderson, M. Vertescher, D. Meger, and M. Coates, "Cost adaptation for robust decentralized swarm behaviour," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[15] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, "Towards viable certificate-based authentication for the internet of things," in *ACM Workshop on Hot Topics on Wireless Network Security and Privacy*, 2013.

[16] C. Jung, A. Ahad, Y. Jeon, and Y. Kwon, "Swarmflawfinder: Discovering and exploiting logic flaws of swarm algorithms," in *IEEE Symposium on Security and Privacy (S&P)*, 2022.

[17] X. Li, Q. Zeng, L. Luo, and T. Luo, "T2pair: Secure and usable pairing for heterogeneous IoT devices," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020.

[18] "MAVLink," https://mavlink.io/en/, 2022, [Online; accessed 20-Dec-2022].

[19] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," *IEEE Transactions on Mobile Computing*, 2009.

[20] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-based zero-interaction pairing and key evolution for advanced personal devices," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014.

[21] J. Obermaier and S. Tatschner, "Shedding too much light on a microcontroller's firmware protection," in *USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[22] "Openthread," 'https://openthread.io/', 2022, [Online; accessed 30-Jul-2022].

[23] M. O. Ozmen and A. A. Yavuz, "Dronecrypt-an efficient cryptographic framework for small aerial drones," in *IEEE Military Communications Conference (MILCOM)*, 2018.

[24] "Pixhawk Series," 'https://tinyurl.com/45hcfsc8', 2023, [Online; accessed 10-Jan-2023].

[25] "Readout Protection," 'https://tinyurl.com/m62dm6fe', 2023, [Online; accessed 10-Jan-2023].

[26] D. Schürmann and S. Sigg, "Secure communication based on ambient audio," in *IEEE Transactions on Mobile Computing*, 2011.

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

[28] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*, 1985.

[29] I. Sluganovic, M. Liskij, A. Derek, and I. Martinovic, "Tap-pair: Using spatial secrets for single-tap device pairing of augmented reality headsets," in *ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2020.

[30] I. Sluganovic, M. Serbec, A. Derek, and I. Martinovic, "Holopair: Securing shared augmented reality using microsoft hololens," in *Annual Computer Security Applications Conference (ACSAC)*, 2017.

[31] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-hellman key distribution extended to group communication," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 1996.

[32] J. Won, S.-H. Seo, and E. Bertino, "A secure communication protocol for drones and smart objects," in *ACM Symposium on Information, Computer and Communications Security*, 2015.

[33] M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, "Comprehensive study of symmetric key and asymmetric key encryption algorithms," in *IEEE International Conference on Engineering and Technology (ICET)*, 2017.