When Drones Meet Privately: Secure Coordination with t-PSI

Matteo Cornacchia
Sapienza University of Rome
Rome, Italy
matteo.cornacchia@uniroma1.it

Riccardo Lazzeretti
Sapienza University of Rome
Rome, Italy
riccardo.lazzeretti@uniroma1.it

Giulio Rigoni
Sapienza University of Rome
Rome, Italy
giulio.rigoni@uniroma1.it

Leonardo Ventura
Sapienza University of Rome
Rome, Italy
leonardo.ventura@uniroma1.it

Abstract

Autonomous multi-agent systems increasingly operate in environments where privacy, anonymity, and operational security are critical. In such settings, drones must often determine which events or locations have been independently confirmed by at least t participants, while revealing nothing about individual trajectories, identities, or exploration patterns. Threshold Private Set Intersection (t-PSI) naturally captures this requirement, but existing constructions rely on persistent identities, authenticated channels, or linkable communication, making them unsuitable for high-risk deployments where structural anonymity is essential. We introduce a fully anonymous t-PSI protocol tailored for distributed exploration tasks. Communication occurs in a You-Only-Speak-Once (YOSO) architecture: each participant sends exactly one anonymous, unlinkable message per round, ensuring confidentiality, integrity, and sender anonymity without authenticated channels. The protocol supports multi-agent exploration scenarios, such as drone-based reconnaissance, where drones must collaboratively identify zones explored by a minimum subset of units while preserving full operational privacy. We formally prove security against semi-honest participants and active man-in-the-middle adversaries interacting with the network. Our analysis shows that the protocol achieves threshold privacy, ciphertext unlinkability, sender anonymity, and robustness against message tampering. This work demonstrates that anonymous threshold-PSI can be realized efficiently in a YOSO framework, enabling privacy-preserving coordination in adversarial and strategically sensitive environments.

Keywords

Security and privacy → Mobile and wireless security; Security Protocols

1 Introduction

In recent years, the growing digitalization of public and private services has made the need for protocols that ensure strong privacy, anonymity, and data minimization increasingly evident. This requirement becomes even more pressing and fundamental in operational environments where autonomous systems, such as aerial drones, must explore or monitor areas that may contain critical infrastructures, sensitive facilities or assets of strategic relevance. In such contexts, the ability to coordinate distributed drones without revealing their precise objectives, trajectories, or collected data, is essential not only for privacy, but also for safety and resilience of sensitive information.

High-risk civilian deployment settings increasingly reflect this need. Modern unmanned systems are routinely employed in disaster -response operations, critical-infrastructure inspection, and humanitarian missions. Guidelines on the ethical and responsible use of UAVs in humanitarian settings are provided by the Humanitarian UAV Code of Conduct and Guidelines [47], the EASA U-Space regulatory framework for safe and coordinated drone operations [17], and the EU Drone Strategy 2.0 [16]. All of them emphasize that coordination mechanisms must preserve the confidentiality of operational data, especially when exposure could reveal mission objectives, inspection patterns, or the location of critical assets.

A representative scenario is the cooperative monitoring of critical infrastructures, such as power plants, ports, or data centers, where multiple drones must validate the status of specific zones. Revealing which drone visited which area, or which zones were explored by only a single drone, may expose sensitive operational patterns, highlight unmonitored corridors, or reveal the location of assets requiring protection. In such settings, information should be disclosed only when corroborated by at least the threshold number of independent agents, while all sub-threshold observations must remain hidden.

These settings introduce two research questions that must be addressed to enable secure and anonymous multi-agent coordination:

- *RQ1 - Anonymous and Privacy-Preserving Coordination.* How can autonomous drones coordinate distributed exploration tasks without revealing trajectories, identities, operational patterns, or sub-threshold information?
- *RQ2 - Private one-shot communication.* How do we efficiently establish a non-interactive communication protocol while maintaining full anonymity constraints, i.e., without relying on persistent identities or authenticated channels?

Among the cryptographic primitives that address these needs, Private Set Intersection (PSI) [20] has emerged as a central tool thanks to its privacy-by-design paradigm. PSI enables multiple parties, each holding a private set, to learn only the elements they

Proceedings on Privacy Enhancing Technologies 2026(4), 290–304

https://doi.org/10.56553/popets-2026-0121

have in common, without revealing any additional information about their respective inputs. Operating within a secure multi-party computation framework [21, 52], PSI has become a foundational primitive in numerous privacy-sensitive applications, including the secure sharing of indicators of compromise in cybersecurity [39], the deduplication of medical databases [3, 53], privacy-preserving audience measurement and advertising conversion [25, 32, 37], and secure record linkage in financial and governmental settings [10].

However, classical PSI is insufficient in our setting: it reveals all common elements, including those contributed by only a small number of drones, thereby violating RQ1. Sub-threshold events may correspond to rare observations, sensitive anomalies, or operational outliers that must remain hidden to avoid leaking mission-critical information. In contrast, threshold-PSI [33] reveals only elements that appear in at least t private sets, providing a more realistic and secure semantics for distributed exploration, anomaly validation, and collaborative sensing. Yet, existing t-PSI constructions typically assume persistent identities, authenticated channels, or linkable communication patterns, making them incompatible with the structural anonymity and one-shot communication requirements captured by RQ2.

We propose a surface-coverage algorithm that enables a group of autonomous drones to determine which regions of a discretized map remain unexplored, namely those that have not been validated by at least t drones, while simultaneously concealing the individual trajectories of each drone and ensuring both maximal communication anonymity and confidentiality of the transmitted data. To achieve this, our protocol encodes exploration events through Shamir secret sharing [44] and aggregates them via t-PSI mechanism, instantiated within a You-Only-Speak-Once (YOSO) [6] communication framework. YOSO is a communication framework designed to support anonymous MPC by preventing persistent identities and ensuring that actions cannot be linked across protocol runs. Each participant sends exactly one message using a fresh, unlinkable key, and no multi-round interaction is allowed. Authenticated or linkable channels are disallowed; only integrity mechanisms that do not reveal the sender are permitted. Participants therefore contribute exactly once in a non-linkable manner, without revealing trajectories, identities, or any information beyond the threshold output.

Based on these components, our work provides the following contributions:

- **A privacy-preserving t-PSI protocol.** We design a Shamir secret sharing-based protocol that achieves a threshold mechanism: multiple drones must collaboratively determine which zones have been explored by a minimum subset of size t, without revealing trajectories, identities, or individual exploration patterns.
- **A YOSO-style MPC architecture for t-PSI.** Our protocol operates in a You-Only-Speak-Once framework, ensuring one-shot, non-interactive communication and preventing correlation across rounds or protocol executions, a crucial property for anonymous t-PSI. This message protocol is therefore designed to support the one-shot anonymity constraints described in RQ2.
- **A practical design for fully anonymous multi-agent exploration.** We show how anonymous threshold-PSI naturally models scenarios with structural anonymity requirements. Participants contribute exactly once, with no persistent identities, no linkable communication patterns, and no authenticated channels tied to long-term keys. Multi-round message unlinkability allows us to satisfy the privacy requirements outlined in RQ1.

The remainder of the paper is structured as follows. Section 2 reviews the related work in this field, while Section 3 introduces the problem overview and the threat model we assume. Section 4 presents the preliminary building blocks and frameworks, including the Shamir secret sharing primitive. Section 5 presents the YOSO-based anonymous coordination protocol and shows how threshold-only information is revealed while preserving participant anonymity and trajectory privacy, thus addressing RQ1 and RQ2. In Section 6, we formally analyze the security of the protocol and prove anonymity, unlinkability, and threshold privacy under the YOSO communication model. Section 7 evaluates a practical implementation of the protocol, while Section 8 delves into ethical considerations. Lastly, Section 9 concludes the paper and outlines future works.

2 Related Works

The literature includes numerous works about threshold variants of private set intersection. Kissner and Song [28] introduce a general framework for PSI and t-PSI based on homomorphic encryption and polynomial representations of sets: each set is encoded as a polynomial whose roots correspond to the elements, and the threshold is enforced by homomorphically combining these polynomials and performing threshold decryption. Their approach follows a fundamentally different paradigm from Shamir-based secret sharing: instead of distributing input shares among participants, the protocol relies on a shared homomorphic encryption key and on algebraic manipulation of encrypted polynomials. This design naturally fits a model with identifiable participants and shared decryption authority, and does not aim to provide anonymity or unlinkability, which distinguishes it from our setting and goals.

On the other hand, Bay et al. [4] present a t-PSI protocol based on Paillier homomorphic encryption and threshold decryption, where participants submit encrypted sets that are homomorphically aggregated and decrypted only when a quorum of servers collaborates. The protocol realizes an asymmetric multi-party t-PSI but requires persistent identities and authenticated channels and provides neither anonymity nor YOSO-style properties; in contrast, our protocol implements an anonymous YOSO-style multi-party threshold-PSI based on Shamir secret sharing and per-round secret permutations.

Relatedly, Mahdavi et al. [33] present a private set intersection protocol for contact tracing based on Paillier homomorphic encryption and threshold decryption, in which users submit encrypted sets that are homomorphically aggregated and compared against the servers' set of infected tokens. The protocol realizes an asymmetric multi-party PSI for contact tracing but does not address the threshold-PSI over on the elements, nor does it provide anonymity or YOSO-style properties.

Chandran et al. [12] develop a highly efficient t-PSI protocol that combines structured hashing with garbled circuits to verify, within an MPC circuit, how many times each element appears. Their protocol optimizes communication but, on the other hand, requires an MPC infrastructure with fixed roles and authenticated channels; so it does not offer either contributor anonymity, nor YOSO-style mechanisms.

In a different vein, Hu et al. [23] present a delegated threshold-PSI protocol in which most of the computational load is offloaded to a server that assists in determining which elements appear in at least t sets, without learning the participants' full inputs. The model assumes persistent identities and authenticated channels and includes mechanisms for tracing malicious participants; however, it provides neither anonymity nor YOSO-style properties.

Beyond the works explicitly targeting threshold-PSI, it is also important to consider research on multi-party PSI protocols that focus on efficiency and robustness but still rely on traditional communication assumptions: Ben-Efraim et al. [5] introduce PSimple, a multiparty PSI protocol secure against malicious adversaries, which significantly improves computational and communication efficiency while maintaining a standard MPC model with persistent identities, fixed roles, and authenticated channels. Similarly, Rosulek and Trieu [41] propose a compact PSI construction optimized for small sets, reducing message sizes and computational overhead. However, these approaches remain grounded in classical MPC assumptions and do not address structural anonymity, unlinkability, or one-shot communication. They illustrate how the literature has largely focused on performance and adversarial robustness without exploring architectures where participants are anonymous, stateless, and allowed to speak only once.

A further body of literature relevant to our setting concerns privacy-preserving coordination in autonomous drones and multi-robot systems, where drones must collaborate without revealing sensitive information such as trajectories, positions or mission objectives. Brighente et al. [9] propose a privacy-preserving mechanism that enables drones to avoid critical infrastructures without disclosing their precise locations. On the other hand, Tedeschi et al. [45] develop a lightweight privacy-preserving proximity-discovery protocol for remotely controlled drones, while Sciancalepore and George [42] address privacy-preserving trajectory matching for autonomous UAVs. More broadly, Castelló Ferrer et al. [11] investigate secure and secret cooperation in robot swarms, and Majcherczyk et al. [34] study distributed data fusion in resource-constrained robotic collectives. Although these works highlight the importance of privacy in multi-agent exploration and coordination, none of them employ PSI or threshold-PSI, nor do they provide anonymity or adopt a YOSO-style communication model.

3 Problem Overview

The drone swarm operates over a *shared area of operation* that is partitioned into smaller labeled regions to enable systematic coordination and coverage tracking. In the considered scenario, this is a simple grid subdivision, and we may refer to each element as "coordinates", as each map section corresponds to one unique coordinate pair in the resulting grid. The high-level goal of this subdivision is to confirm the coverage of each area by multiple independent, low-trust parties.

Our protocol is designed to perform secure multi-party computation between three main types of parties: *participants*, *reconstructors*, and *key holders*.

The drones actively exploring the map are classified as participants. As the exploration progresses, each swarm member has the goal of maximizing individual coverage, while minimizing the time spent on areas already confirmed by other members. However, it must do so without disclosing or receiving information regarding the current location - or the location history - of other individual members. Furthermore, participants are assumed to have very limited computing power and cannot be expected to perform complex and power-consuming cryptographic calculations.

Reconstructors are network members which are tasked with receiving securely encoded transmissions from the participants, and calculating the resulting public information, that is the over-threshold private set intersection for each map coordinate. Due to the design of the protocol, reconstructors need to receive information from each member and broadcast to as many members as possible. They are also tasked with a number of calculations that are both computationally expensive and highly parallelizable. For these reasons, reconstructors are best modeled as servers on the ground, with one or multiple antennas for direct wireless communication with swarm members over the geographical area of the map.

Key holders are the trusted members responsible for computing the key pairs and secret shares. In order to participate in the online protocol, each participant and reconstructor must interface with the Key holder during the so-called offline initialization phase, described in Section 5.2. Thus, they are the only swarm member who has a high level of required trust. After the initial phase, the Key holder can flush all temporary and potentially sensitive cryptographic data, such as the original polynomials, as it is unnecessary for the operation of the online phases of the protocol, and essentially ceases to exist. Notice that the Key holder entity in our work is not a Dealer in the most traditional sense (i.e., in secret sharing primitive sense), as it operates entirely offline. Once its computation is completed, all intermediate values and temporary data are securely erased. This prevents any form of state-based leakage, since all rounds are computationally independent by design. Finally, bear in mind that online operations for each round are performed by the Reconstructor, which cannot access any privileged information and cannot extrapolate sensitive data during its operation.

To avoid collusion mechanisms that undermine the privacy goals of the protocol, each member of the described network can only be assigned one role. For instance, a Key holder cannot also be a Participant or a Reconstructor. While it is possible to securely distribute and parallelize some of the tasks performed by key holders and reconstructors, doing so introduces security challenges linked to the transmission of partially decrypted data, which are out of the scope of this work. Therefore, we will consider scenarios with one key holder and one reconstructor, at least from an outside view of the system.

In civilian multi-stakeholder deployments, autonomous drones operated by civil-protection units, environmental agencies, NGOs, or private operators often collaborate temporarily on shared monitoring or inspection tasks. In these settings, even partial leakage

of coverage patterns or movement trajectories can reveal sensitive operational information. Prior work in structural-health monitoring [18], remote-sensing validation [50], and humanitarian UAV operations [47] shows that single observations are frequently ambiguous or noisy, and that independent corroboration from multiple agents is required to reliably confirm anomalies, damage indicators, or environmental events. At the same time, revealing which specific drone contributed an observation, or that only one drone visited a given region, may expose organizational priorities, inspection routines, or unmonitored corridors. For these reasons, only the fact that a region has been validated by at least a threshold number of independent agents should be disclosed, while all sub-threshold information must remain strictly confidential.

Operational constraints in these domains also require threshold values to remain small. In real-time surveillance, infrastructure inspection, and environmental-monitoring missions, only a limited number of drones can observe the same area within the same time window due to mobility constraints, battery limitations, and restricted overlap in coverage [27, 35, 43]. Humanitarian UAV guidelines similarly emphasize timely coordination, rapid validation of critical observations, and the minimization of operational delays, consensus across, since delayed decision may compromise safety or mission effectiveness [47]. From a statistical perspective, a small number of independent observations is often sufficient to reach high confidence: for instance, three independent detections at an individual accuracy of 79% yield a combined accuracy of approximately 99%. While real-world correlations reduce this idealized gain, current research in computer vision supports the feasibility of small-threshold validation, with individual classification accuracies ranging from 51% for general-purpose object detectors [49] to over 80% for task-specific models [15, 26], and exceeding 95% in specialized wildfire-detection systems [7]. Across these applications, fixed threshold values up to $t = 7$ provide a practical balance between operational feasibility and confidence in the aggregated observations.

We now describe how the protocol enforces these privacy requirements during its operation.

From a technical standpoint, the protocol is designed so that participants share only the minimum information required for safe coordination. Each drone transmits a single anonymous message per round containing an encoded representation of the areas it has explored, indistinguishable from random data for all other areas. Crucially, the message does not reveal raw coordinates, trajectories, or incremental differences between rounds: participants include both newly explored regions and previously explored but still unconfirmed ones, preventing the Reconstructor from inferring movement patterns or correlating contributions across time. The Reconstructor can therefore only determine which regions have been independently confirmed by at least t agents, while all sub-threshold information-number of visits, contributor identities, and cross-round variations-remains hidden, answering RQ2.

These requirements mandate the use of a YOSO communication model: each message is encrypted with fresh, one-shot cryptographic material, carries no persistent identifier, and cannot be linked to messages from the same participant in previous rounds. This prevents correlation attacks and ensures structural anonymity even in the presence of a semi-honest Reconstructor, addressing RQ1. The protocol also remains robust under partial information: packet loss or replay attempts do not leak whether a region was unexplored or simply not transmitted, as integrity checks and share validation ensure that only fresh, valid contributions affect the threshold computation.

3.1 Threat Model

We consider adversarial scenarios consistent with the Semi-Honest model, where each member of the protocol may attempt to extract additional information from what is publicly available to them, but may not perform active attacks, i.e., craft malicious packets, disrupt the intended protocol flow, or perform denial of service attacks.

Adversarial Capabilities and Goals. We consider two main attacker scenarios. These scenarios vary significantly in attacker capabilities and goal, as they model potential attack surfaces in the network and endpoint layers, respectively.

In the Man-in-the-Middle scenario, the adversary can monitor the wireless communication channel between the participants and the reconstructor, and has the capability to interfere with normal network operation by intercepting, replaying, adding, and removing packets, but does not have access to the secret keys required to encrypt, decrypt, and sign the coordinate vectors sent from each participant to the reconstructor.

In the Semi-Honest attacker scenario, an attacker compromises the reconstructor, or up to $t - 1$ participants, and gains access to the member's personal secrets, such as its private exploration set or any secret keys. Their main goal is to extract the private information of other swarm members by manipulating the public information that each member receives from the rest of the swarm. However, a corrupted attacker cannot forge arbitrary coordinate vectors in the case of the participants, or the final PSI in the case of the reconstructor.

To prevent metadata-based linkability, we assume that the communication layer provides ephemeral, unlinkable channels, so that the reconstructor cannot access persistent identifiers (e.g., IP/MAC addresses) or timing patterns that would allow correlating multiple transmissions to the same participant. Such assumptions are standard in anonymous communication systems [13, 38, 48], and define the limits of the adversary's capabilities at the network layer.

Under these assumptions, a corrupted reconstructor should be unable to determine the location of individual swarm members, nor identify their communication between rounds. Conversely, corrupted participants must not be able to extract information on under-threshold coordinates or the location history of other participants. We will see in Section 6 that, given standard assumptions on the soundness of the chosen cryptographic primitives, these conditions are satisfied by our proposed protocol.

4 Preliminaries

In this section, we introduce the fundamental cryptographic primitives used to build our protocol and achieve the required security properties.

Key Derivation Function (KDF). We rely on a standard Key Derivation Function to derive pseudorandom and domain-separated keys from different secret inputs used in the protocol. In particular, we

instantiate the KDF with HKDF [30], built from HMAC [29] and SHA-256 [2]. The KDF is used both to expand the shared secret produced by the KEM into a symmetric encryption key bound to a public context string, and to derive round-specific MAC keys from the global master key K_G. A formal specification of the KDF interface and the properties we require is given in the protocol description below.

Authenticated Encryption with Associated Data (AEAD). For message protection, we rely on standard symmetric primitives: an EUF-CMA-secure MAC scheme (instantiated via HMAC [29]) and an AEAD scheme with standard IND-CCA and INT-CTXT guarantees, following the syntax of Rogaway [40]. In practice, we adopt widely deployed AEAD constructions such as ChaCha20-Poly1305 [36] and AES-GCM [1].

Key Encapsulation Mechanism (KEM). We rely on a standard IND-CCA secure Key Encapsulation Mechanism, instantiated with X25519 [31]. A sender encapsulates a fresh shared secret under the receiver's public key; this shared secret is then expanded via HKDF into the symmetric key used in our KEM+AEAD public-key encryption construction, described in Section 5.1.

Shamir Secret Sharing. Shamir secret sharing, introduced by Shamir [44], is a threshold secret-sharing scheme that allows any authorized set of parties to reconstruct a secret via Lagrange interpolation, while guaranteeing perfect privacy against any unauthorized set. More precisely, a (t, n)-Shamir secret-sharing scheme, where n is the number of parties and t is the reconstruction threshold, allows a dealer (in our protocol performed by the Key-Holder) to distribute a secret $M \in \mathbb{F}$ by sampling a random polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1},$$

of degree $t-1$, with coefficients $a_0, \ldots, a_{t-1} \in \mathbb{F}$ such that $f(0) = M$. Each party $i = 1, \ldots, n$ receives the share $\sigma_i = (x_i, f(x_i))$, where $x_1, \ldots, x_n$ are distinct non-zero field elements, and only sets of at least t parties can jointly reconstruct the secret by combining their shares via Lagrange interpolation, while any other set of fewer than t parties learns no information about M. For any subset $T \subseteq [n]$ with $|T| = t$, the secret is recovered as

$$M = f(0) = \sum_{i \in T} f(x_i) \prod_{\substack{k \in T \\ k \neq i}} \frac{-x_k}{x_i - x_k}.$$

Furthermore, this scheme guarantees perfect privacy: any set of fewer than t shares is consistent with every possible value of M, and therefore reveals no information about the secret.

YOSO framework. YOSO (You-Only-Speak-Once) is a communication framework designed to support anonymous MPC that forces participants to have no persistent identities, i.e., to not maintain long-term or stable identities within the system, preventing correlation across rounds or protocol executions. It uses ephemeral one-time and unlinkable keys for sending each message. It also performs one-shot communication: each participant sends exactly one message per protocol instance, with no interactive exchanges, no multi-round communication, in order to prevent the system from building behavioral or temporal profiles.

5 Proposed Exploration System

In this section, we formalize the YOSO-based t-PSI exploration protocol. We first introduce the notation, roles, and cryptographic assumptions shared across all phases (Section 5.1); next we describe the protocol itself (Section 5.2), structured into an offline initialization and an online mission phase.

In accordance with the YOSO communication model, the protocol operates without authenticated or linkable channels tied to persistent identities. Participants use only integrity mechanisms that do not reveal the sender, and each party contributes exactly one unlinkable message per protocol instance; this ensures that no long-term identity is maintained and that no information is disclosed beyond what becomes known once the threshold is met.

5.1 Notation and Assumptions

We first introduce the notation, roles, and cryptographic assumptions that are common to all components of the protocol; all subsequent phases of the protocol (initialization, online execution, and reconstruction) use this notation consistently. We assume that Shamir secret sharing is instantiated over a finite field $\mathbb{F}$ with $|\mathbb{F}| \geq 2^\lambda$, so that dummy values $\perp \in \mathbb{F}$ are computationally indistinguishable from real shares.

In order to improve readability, we begin by describing our protocol in three main parts; all three parts share a common notation, which we introduce below:

- λ security parameter;
- $n > 0$ participants P_i (excluding $\mathcal{KH}$ and $\mathcal{R}$);
- $t > 0$, reconstruction threshold;
- $\ell > 0$, number of secrets;
- $R_{\max} > 0$, number of rounds;
- i, index of parties from 1 to n;
- j, index of shares from 1 to ℓ;
- r, index of rounds from 1 to $R_{\max}$;
- if S is any set, we denote its complement S^c;
- $\sigma_i^{j,(r)}$ denote the i-th participant's share $(x_i, y_i)^{j,(r)}$ corresponding to the j-th polynomial $f_j^{(r)}$ at round r.

We assume a key-derivation function $\mathsf{KDF} : \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^L$, with $L \geq \lambda$, that satisfies:

- **Pseudorandomness:** For any efficient adversary not knowing the master key K_G, the output $\mathsf{KDF}(K_G, x)$ is computationally indistinguishable from a uniform string of length L.
- **Determinism:** For fixed K_G and input x, the output is always the same.
- **Public input:** The second input x (context, round index, etc.) may be public and need only be unique and canonically encoded.
- **Output length:** L is sufficient for the round keys (typically $L = \lambda$ bits).

The implementation relies on an HKDF (i.e., a KDF built on HMAC) to derive cryptographic keys from shared secret material in a secure and standardized manner. We also rely on a public-key encryption mechanism instantiated as a KEM+AEAD construction:

- A KEM scheme (KEM.KeyGen, KEM.Encaps, KEM.Decaps), internally built from a Diffie-Hellman-type key agreement;

- An AEAD scheme (AEAD.Enc, AEAD.Dec) with key space $\{0,1\}^L$, e.g. ChaCha20-Poly1305 or AES-GCM.

The KEM and the KDF are combined as follows: given ss and a public context string info, the encryption key is derived as $K = \mathsf{KDF}(ss, \mathsf{info})$.

5.2 Protocol

We now describe the protocol itself that conceptually relies on two layers deeply coupled together:

- A core layer implementing exploration via t-PSI based on Shamir secret sharing; and
- a YOSO communication layer providing anonymous, public-key-based encryption from participants P_i to the Reconstructor $\mathcal{R}$, realized via a KEM + AEAD construction.

The protocol execution is organized into two phases: (i) an offline initialization phase, where parameters and keys are prepared, and (ii) an online mission phase, where drones come online and submit their one-shot, unlinkable YOSO messages carrying the secret-shared information.

5.2.1 Offline Phase. The first part of the protocol consists of an offline phase performed by the key-holder, while all drones (both drones and the base station $\mathcal{R}$) are still on the ground at the base camp. We have divided this phase into two main algorithms.

Map and Key Setup. Described in Algorithm 1 is the stage where $\mathcal{KH}$ generates the polynomials required for (n,t)-secret sharing of the zones for all rounds; computes the shares for each secret; samples a master key uniformly at random, for the KDF and generates the key-pair of the reconstructor for the KEM.

Distribution. It is the second stage, described in Algorithm 2, where $\mathcal{KH}$ distributes to each participant: the key-derivation function KDF with its parameters, the master key previously sampled for the KDF, the Reconstructor's public key, an empty LastPSI set for storing the last PSI instance, and finally the shares. It is important to underline that this share distribution depends on a secret permutation $\pi^{(r)}$ selected by $\mathcal{KH}$ for each round, ensuring that the index i of the shares cannot be linked to the index i of the participants across rounds when $\mathcal{R}$ receives vector $E_i^{(r)}$ from participant P_i. Anyway, we abuse the notation by writing $\sigma_i^{j,(r)}$ instead of $\sigma_{\pi^{(r)}(i)}^{j,(r)}$; that is, participant P_i receives the share $\left(x_{\pi^{(r)}(i)}, f_j^{(r)}(x_{\pi^{(r)}(i)})\right)$ for each j and each round r.

5.2.2 Online Phase. After the offline initialization performed by the Key-Holder, we have now all drones ready to start the online mission divided into *Exploration* where participants collect explored zones and send them to $\mathcal{R}$; *Reconstruction*, where $\mathcal{R}$ computes the above threshold zones; and *Handover*, where Reconstructor $\mathcal{R}$ sends the private set intersection to all drones.

Exploration. In the exploration phase described in Algorithm 3, each drone P_i initializes an empty private set S_i in order to collect its explored zones and add them to this set. Then each party P_i computes the difference between S_i and LastPSI in order to build a frontier set to prevent differential attacks by the reconstructor $\mathcal{R}$ from linking a received vector $E_i^{(r)}$ to a specific index i in the Shamir secret sharing; next generates an ordered vector $E_i^{(r)}$ that in position j contains the share of the j-th zone or a dummy value sampled with the same distribution of a real share. Next, each participant P_i encrypt vector $E_i^{(r)}$ with the public key of Reconstructor, using the KEM+AEAD construction obtaining the encrypted vector $C_i^{(r)}$ using the derived symmetric key $K_i^{(r)}$ of the KEM scheme from a shared secret $ss_i^{(r)}$ and from additional information info_r, in our case $\mathsf{info}_r = r$; note that info_r contains no participant-specific identifier, preserving sender anonymity. Finally, we compute the MAC of the encrypted vector, in a anncrypt-then-MAC way, in order to guarantee integrity and authenticity. With a MAC-key derived through the common KDF from the common master key K_G, for each ciphertext $C_i^{(r)}$, we compute its respective $\mathsf{tag}_i^{(r)}$ and send the pair $(C_i^{(r)}, \mathsf{tag}_i^{(r)})$ to $\mathcal{R}$.

Algorithm 1: Offline Initialization — Map and Key Setup

We suppose the map $\mathcal{M}$ is divided into ℓ zones.

Map and Key Setup. Key-Holder $\mathcal{KH}$ does:

- For every round $r = 1, \ldots, R_{\max}$, generate ℓ polynomials $f_1^{(r)}, \ldots, f_\ell^{(r)}$ of degree $t-1$, for secret sharing the zones, imposing $f_j^{(r)}(0) = M_j$ for every $j = 1, \ldots, \ell$ and for all $r = 1, \ldots, R_{\max}$.
- Compute the shares $\sigma_i^{1,(r)}, \ldots, \sigma_i^{\ell,(r)}$ for all participants $i = 1, \ldots, n$ and for all rounds $r = 1, \ldots, R_{\max}$, corresponding to the polynomials $f_1^{(r)}, \ldots, f_\ell^{(r)}$.
- Sample a master key $K_G \leftarrow_{\$} \{0,1\}^\lambda$ uniformly at random.
- Run the KEM key generation for the Reconstructor:

$$(\mathsf{sk}_{\mathcal{R}}, \mathsf{pk}_{\mathcal{R}}) \leftarrow \mathsf{KEM.KeyGen}(1^\lambda).$$

Algorithm 2: Offline Initialization — Distribution

Distribution. For every i, $\mathcal{KH}$ distributes to participant P_i:

- Shares

$$\left\{\sigma_i^{1,(r)}, \ldots, \sigma_i^{\ell,(r)}\right\}_{r=1,\ldots,R_{\max}},$$

thus each participant P_i holds the following share matrix

$$\begin{bmatrix} \sigma_i^{1,(1)} & \cdots & \cdots & \sigma_i^{\ell,(1)} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_i^{1,(R_{\max})} & \cdots & \cdots & \sigma_i^{\ell,(R_{\max})} \end{bmatrix}$$

- The master key K_G.
- The key-derivation function KDF, with its parameters.
- The public key of the Reconstructor $\mathsf{pk}_{\mathcal{R}}$.
- A set LastPSI, initialized as empty and updated at the end of each round.

In addition, $\mathcal{KH}$ distributes to $\mathcal{R}$:

- The secret key $\mathsf{sk}_{\mathcal{R}}$ corresponding to $\mathsf{pk}_{\mathcal{R}}$.
- The master key K_G and the description of KDF, so that $\mathcal{R}$ can derive the same round keys as the participants.

Algorithm 3: Online Mission Phase — Exploration (Round r)

Exploration. For all $i = 1, \ldots, n$, participant P_i does:

- Construct an initially empty set $S_i = \emptyset$.
- Add the explored zones $\{M_j\}_{j \in J \subseteq [\ell]}$ to S_i.
- Compute
$$S_i - \mathsf{LastPSI} = S_i - \mathsf{PSI}^{(r-1)}.$$
- Generate the vector
$$E_i^{(r)} = \left(\widetilde{\sigma}_i^{1,(r)}, \ldots, \widetilde{\sigma}_i^{\ell,(r)}\right),$$
where
$$\widetilde{\sigma}_i^{j,(r)} = \begin{cases} \sigma_i^{j,(r)} & \text{if } P_i \text{ collected } M_j \text{ in } S_i - \mathsf{LastPSI} \\ \perp & \text{otherwise.} \end{cases}$$
- Encrypt $E_i^{(r)}$ under $\mathsf{pk}_{\mathcal{R}}$ using the KEM+AEAD construction:

(1) Run KEM encapsulation
$$(ct_{\mathsf{kem}}, ss_i^{(r)}) \leftarrow \mathsf{KEM.Encaps}(\mathsf{pk}_{\mathcal{R}}).$$
(2) Derive the symmetric encryption key
$$K_i^{(r)} = \mathsf{KDF}(ss_i^{(r)}, \mathsf{info}_r),$$
where info_r is a public context string depending only on the protocol identifier and the round index r.
(3) Sample uniformly at random a nonce $\mathsf{nonce}_i^{(r)}$ and compute
$$C_{\mathsf{sym},i}^{(r)} = \mathsf{AEAD.Enc}\left(K_i^{(r)}, \mathsf{nonce}_i^{(r)}, E_i^{(r)}, AD_i^{(r)}\right),$$
where $AD_i^{(r)}$ contains public metadata (e.g., the round index r).
(4) Set the ciphertext
$$C_i^{(r)} = \left(ct_{\mathsf{kem}}, \mathsf{nonce}_i^{(r)}, C_{\mathsf{sym},i}^{(r)}, AD_i^{(r)}\right).$$

- Derive the round MAC key $k_G^{(r)} = \mathsf{KDF}(K_G, r)$.
- Compute the tag
$$\mathsf{tag}_i^{(r)} = \mathsf{MAC}_{k_G^{(r)}}\left(C_i^{(r)} \,\|\, r\right).$$
- Send $(C_i^{(r)}, \mathsf{tag}_i^{(r)})$ to $\mathcal{R}$.

Algorithm 4: Online Mission Phase — Reconstruction (Round r)

Reconstruction. The Reconstructor $\mathcal{R}$:

- Anonymously receives $\{(C_i^{(r)}, \mathsf{tag}_i^{(r)})\}_{i \in I \subseteq [n]}$.
- Derives $k_G^{(r)} = \mathsf{KDF}(K_G, r)$.
- Verifies each tag $\mathsf{tag}_i^{(r)}$; otherwise discards the message.
- For each valid ciphertext, decrypts as follows:

(1) Parses
$$C_i^{(r)} = \left(ct_{\mathsf{kem}}, \mathsf{nonce}_i^{(r)}, C_{\mathsf{sym},i}^{(r)}, AD_i^{(r)}\right).$$
(2) Computes the shared secret
$$ss_i^{(r)} = \mathsf{KEM.Decaps}(\mathsf{sk}_{\mathcal{R}}, ct_{\mathsf{kem}}).$$
(3) Derives the symmetric key
$$K_i^{(r)} = \mathsf{KDF}(ss_i^{(r)}, \mathsf{info}_r).$$
(4) Recovers the plaintext vector
$$E_i^{(r)} = \mathsf{AEAD.Dec}\left(K_i^{(r)}, \mathsf{nonce}_i^{(r)}, C_{\mathsf{sym},i}^{(r)}, AD_i^{(r)}\right).$$

- Constructs the $(n \times \ell)$ matrix of received vectors
$$\begin{bmatrix} E_1^{(r)} \\ \vdots \\ E_n^{(r)} \end{bmatrix} = \begin{bmatrix} \widetilde{\sigma}_1^{1,(r)} & \cdots & \widetilde{\sigma}_1^{\ell,(r)} \\ \vdots & \ddots & \vdots \\ \widetilde{\sigma}_n^{1,(r)} & \cdots & \widetilde{\sigma}_n^{\ell,(r)} \end{bmatrix}.$$
- For each column j attempts to identify a subset of t- non dummy shares that interpolate to a polynomial of degree $t-1$, and recovers $M_j = f_j^{(r)}(0)$.
- If one such subset exists, add j to $\mathsf{PSI}^{(r)}$.

Algorithm 5: Online Mission Phase — Handover (Round r)

Handover. The Reconstructor $\mathcal{R}$:

- Broadcasts $\mathsf{PSI}^{(r)}$ to all P_i.

Each participant P_i:

- Saves $\mathsf{PSI}^{(r)}$ in $\mathsf{LastPSI}$.
- Computes $\left(\mathsf{PSI}^{(r)}\right)^{\mathsf{c}} \cap S_i^{\mathsf{c}}$.
- Computes $\left(\mathsf{PSI}^{(r)}\right)^{\mathsf{c}} \cap S_i$.

Reconstruction. After receiving all ciphertext–tag pairs for round r, the Reconstructor $\mathcal{R}$ performs the reconstruction phase described in Algorithm 4. Upon reception, $\mathcal{R}$ first derives the round MAC key $k_G^{(r)} = \mathsf{KDF}(K_G, r)$ and verifies the authenticity of each message; any pair $(C_i^{(r)}, \mathsf{tag}_i^{(r)})$ whose tag does not validate is discarded. For every valid ciphertext, $\mathcal{R}$ parses the KEM and AEAD components, computes the shared secret via the KEM decapsulation, derives the symmetric key $K_i^{(r)}$, and finally decrypts the AEAD ciphertext to recover the plaintext vector $E_i^{(r)}$.

Once all valid vectors have been recovered, $\mathcal{R}$ arranges them as the rows of an $(n \times \ell)$ matrix, where each column j collects all contributions for zone j. For each column, $\mathcal{R}$ attempts to identify a subset of t non-dummy entries that interpolate to a polynomial of degree $t-1$ over $\mathbb{F}$; if such a subset exists, the corresponding secret $M_j = f_j^{(r)}(0)$ is successfully reconstructed. Every index j for which a valid interpolation is found is added to the public set $\mathsf{PSI}^{(r)}$, which contains exactly the zones that have been explored by at least t participants in round r.

Handover phase. After completing the reconstruction step of round r, the Reconstructor broadcasts the set $\mathsf{PSI}^{(r)}$, which contains exactly the coordinates for which at least t valid shares were received.

The sequence $\mathsf{PSI}^{(1)}, \mathsf{PSI}^{(2)}, \ldots$ is monotone increasing: once a coordinate reaches the threshold in some round, it remains confirmed in all subsequent rounds. This property ensures that once a coordinate has reached the threshold, participants no longer contribute valid shares for it. The Reconstructor can immediately discard that column in subsequent rounds without performing a full interpolation attempt. More importantly, monotonicity prevents a semi-honest Reconstructor from performing differential attacks by comparing the evolution of the public output across rounds to infer which participants contributed new information.

Upon receiving $\mathsf{PSI}^{(r)}$, each participant stores it as LastPSI and uses it to update its local exploration strategy. The set of globally unexplored coordinates, those not in $\mathsf{PSI}^{(r)}$, is the only region where new information can still be contributed. Each participant identifies which of these coordinates has not yet been visited, i.e. have not yet reached the threshold, guiding its next-round movement toward areas that may still help the swarm reach consensus. These coordinates must be resubmitted in the next round to prevent differential leakage: without this step, the Reconstructor could observe which positions in the encrypted vectors transition from dummy to valid shares and infer newly explored coordinates for each participant. By ensuring that participants always include both newly explored and previously explored but unconfirmed coordinates, the protocol prevents the Reconstructor from distinguishing incremental contributions or linking ciphertext patterns across rounds.

However, such strategies are developed fully locally and independently, and collision risks rise in likelihood as the remaining unexplored areas shrink in number. To solve this, each drone must rely on short-term collision avoidance mechanisms, or integrate with other privacy-aware trajectory negotiation systems. Such systems are outside the scope of this research. We discuss this area further in Section 9.

5.3 Robustness Considerations

As mentioned in Section 3, each individual member can only cover one of the three roles. Collusion between multiple parties of different roles critically weakens the threshold-based share validation mechanism performed by the Reconstructor.

More specifically, access to the Key holder private data from another role would allow anyone to break threshold safety by generating $t-1$ sets of valid shares, reducing the validation problem to linear time. Collusion between the Reconstructor and k Participants would have a similar effect, reducing the required threshold to $t-k$.

The protocol also exhibits several operational properties that are relevant for understanding its behavior across rounds and under imperfect communication conditions.

In every iteration, each participant transmits both the newly explored shares and the subset of previously explored but still unconfirmed shares. This design allows the system to reuse computations from earlier rounds while preventing differential attacks: no participant ever reveals incremental changes in its private set, and the Reconstructor cannot infer sub-threshold contributions in previous rounds from confirmed coordinates in the current round.

In practical deployments, the Reconstructor may receive only a subset of the expected messages due to packet loss, intermittent connectivity, or temporary communication failures. The protocol remains correct under such conditions: reconstruction proceeds over the available shares, and any coordinate that does not reach the threshold is simply treated as under-threshold, without revealing whether missing contributions correspond to unexplored regions or to transmission issues. This prevents partial information from leaking operational details and preserves the privacy guarantees of the scheme. Furthermore, replayed or duplicated packets cannot affect the outcome, as MAC verification and share validation ensure that stale or malformed contributions are discarded. As a result, the system maintains robustness even in unreliable or adversarial communication environments.

5.4 Complexity Discussion

For each column j, the Reconstructor must identify a subset of at least t shares that lie on a polynomial of degree $t-1$. Since dummy shares are sampled uniformly at random and are computationally indistinguishable from real shares, this search is combinatorial in the worst case. More specifically, the reconstruction algorithm has a runtime complexity of

$$\tau_t \cdot L \cdot \binom{N}{t},$$

where τ_t is the time to validate a t-degree polynomial via Lagrange interpolation, L is the number of coordinates, and $\binom{N}{t}$ is the combination of t-sized subsets among N participants' shares for that coordinate.

However, in our target deployment $N \leq 1000$ and the number of honest contributions per column is limited; in practice, inconsistent subsets can be discarded early, and a simple incremental interpolation strategy suffices. We therefore treat the reconstruction step as efficient under our concrete parameter regime, while acknowledging its worst-case exponential nature.

6 Security Analysis

In this section we analyze and prove the security of the YOSO-based t-PSI exploration protocol under different adversarial models. We begin by introducing the common notation and the cryptographic assumptions (Section 6.1); then we consider semi-honest (honest-but-curious) adversaries, focusing on two corruption scenarios: a semi-honest Reconstructor and a semi-honest coalition of participants of size less than t (Section 6.2); and finally, we discuss and prove the resilience of the protocol against active, man-in-the-middle adversaries that may deviate from the prescribed behavior or tamper with the communication channels (Section 6.3).

6.1 Security Assumptions for Semi-Honest Adversary and MITM Attacks

We state the common security assumptions we require to our cryptographic functions, in order to prove security both in presence of Semi-Honest Adversary and a MITM adversary having capabilities described in Section 3.1. We assume the following standard cryptographic properties:

- Shamir secret sharing over $\mathbb{F}$ is information-theoretically secure: any set of fewer than t shares reveals no information about the secret.
- The KEM is IND-CCA secure.

- The AEAD scheme is IND-CCA and INT-CTXT secure.
- The KDF is a pseudorandom function keyed by K_G.
- The MAC is existentially unforgeable under chosen-message attacks under keys $k_G^{(r)} = \mathsf{KDF}(K_G, r)$.

6.2 Security Against Semi-Honest Adversaries

We now prove the security of our protocol against a semi-honest adversary $\mathcal{A}$, under the assumptions of Section 6.1; we show that such an adversary learns nothing beyond the protocol outputs and the public transcript. To do this, our proof follows the standard simulation-based paradigm: for each corruption scenario, we construct a PPT simulator that reproduces the adversary's view in an indistinguishable way, i.e. whose output distribution is computationally indistinguishable from the adversary's real view; therefore we obtain that no PPT adversary can distinguish whether it is interacting with a real execution of the protocol or with the simulator. In particular, since the simulator only uses the protocol outputs and the public transcript as input, this implies that $\mathcal{A}$ cannot learn anything beyond this information.

Theorem 6.1 (Semi-honest security of the YOSO Exploration Protocol)**.** *Under the assumptions of Section 6.1, the YOSO Exploration Protocol is secure in the semi-honest model against any polynomial-time adversary corrupting either:*

(1) *the Reconstructor $\mathcal{R}$ alone; or*
(2) *any subset of participants $\{P_i\}_{i\in S}$ with $|S| < t$.*

In particular, such an adversary learns nothing beyond:

- *the protocol outputs (the reconstructed zone secrets M_j for which at least t valid shares are received); and*
- *the public transcript (ciphertexts, tags, and broadcast messages).*

Proof. We construct simulators for each adversarial case and show that their simulated views are computationally indistinguishable from the real ones.

Case 1: Semi-honest Reconstructor. The real view of $\mathcal{R}$ consists of:

- its keys $(\mathsf{sk}_{\mathcal{R}}, K_G)$;
- the ciphertexts $(C_i^{(r)}, \mathsf{tag}_i^{(r)})$ received in each round;
- the decrypted vectors $E_i^{(r)}$; and
- the reconstructed secrets M_j.

We build a simulator $\mathsf{Sim}_{\mathcal{R}}$ as follows.

(1) **Simulating KEM+AEAD ciphertexts.** For each real ciphertext
$$C_i^{(r)} = (ct_{\mathsf{kem}}, \mathsf{nonce}_i^{(r)}, C_{\mathsf{sym},i}^{(r)}, AD_i^{(r)}),$$
the simulator samples:
- $ct_{\mathsf{kem}}^{\star}$ as a random public key (identical distribution to an ephemeral KEM key);
- $\mathsf{nonce}_i^{(r)\star}$ uniformly at random;
- $C_{\mathsf{sym},i}^{(r)\star}$ as a uniformly random string of the same length as $C_{\mathsf{sym},i}^{(r)}$;
- $AD_i^{(r)\star}$ either as the same public metadata prescribed by the protocol (e.g., the round index r), or as any value consistent with the allowed public information.

By IND-CCA security of the KEM+AEAD public-key encryption scheme, and since $AD_i^{(r)}$ contains only public metadata, this simulated distribution is indistinguishable from the real one, even to an adversary holding $\mathsf{sk}_{\mathcal{R}}$.

(2) **Simulating MAC tags.** Since $\mathsf{Sim}_{\mathcal{R}}$ knows K_G, it computes $k_G^{(r)} = \mathsf{KDF}(K_G, r)$ and outputs
$$\mathsf{tag}_i^{(r)\star} = \mathsf{MAC}_{k_G^{(r)}}(C_i^{(r)\star} \,\|\, r).$$
Thus the distribution of tags in the simulated view is identical to that in the real execution.

(3) **Simulating plaintext vectors $E_i^{(r)}$.** The simulator knows only which secrets M_j are reconstructed (the protocol output). For each zone j:
- If M_j is *not* reconstructed, the simulator samples $\widetilde{\sigma}_i^{j,(r)\star}$ uniformly from $\mathbb{F} \cup \{\perp\}$.
- If M_j *is* reconstructed, the simulator chooses any set of t shares interpolating to M_j and fills the remaining entries with random dummy values.

By the information-theoretic security of Shamir sharing and the fact that the permutation $\pi^{(r)}$ is secret, this distribution is identical to the real one from the perspective of $\mathcal{R}$.

(4) **Anonymity.** Because each ciphertext uses a fresh ephemeral key pk_e and the share indices are permuted by $\pi^{(r)}$, the Reconstructor cannot link any ciphertext to a specific participant. The simulator may therefore assign rows arbitrarily.

Thus, the simulated view is indistinguishable from the real one.

Case 2: Semi-honest coalition of participants with $|S| < t$. The adversary corrupts a subset $S \subset [n]$ of participants, with $|S| < t$. Its view consists of:

- the local state of the corrupted drones, including their shares $\{\sigma_i^{j,(r)}\}_{j,r}$ for all $i \in S$;
- the master key K_G and the description of KDF;
- the public key $\mathsf{pk}_{\mathcal{R}}$ of the Reconstructor;
- the ciphertexts and tags $(C_i^{(r)}, \mathsf{tag}_i^{(r)})$ that they generate and send to $\mathcal{R}$ in each round;
- the public broadcast messages (e.g., $\mathsf{PSI}^{(r)}$ and its complement) sent during the handover phase.

We construct a simulator Sim_S that, given the corrupted participants' inputs and the global protocol output (the reconstructed secrets M_j), produces a view that is indistinguishable from the real one.

(1) **Simulating Shamir shares.** For each zone j and round r:
- If M_j is not reconstructed in round r, the simulator samples a random polynomial $f_j^{(r)}$ of degree $t-1$ over $\mathbb{F}$ and assigns to each corrupted participant P_i (with $i \in S$) the share $\sigma_i^{j,(r)} = f_j^{(r)}(x_i)$. By the information-theoretic security of Shamir secret sharing, the joint distribution of fewer than t shares is independent of M_j, so this is consistent with any underlying secret.
- If M_j is reconstructed, the simulator samples a random polynomial $f_j^{(r)}$ of degree $t-1$ conditioned on $f_j^{(r)}(0) = M_j$ and again sets $\sigma_i^{j,(r)} = f_j^{(r)}(x_i)$ for all $i \in S$.

In both cases, the distribution of the corrupted participants' shares matches the one in a real execution.

(2) **Simulating ciphertexts and tags.** Given the simulated shares, the master key K_G, and the public key $\mathsf{pk}_{\mathcal{R}}$, the simulator can faithfully execute the exploration algorithm (Algorithm 3) on behalf of each corrupted participant P_i, for all rounds r:
- it constructs the sets S_i and the frontier $S_i - \mathsf{LastPSI}$, and builds the vectors $E_i^{(r)}$ exactly as prescribed;
- it runs the KEM+AEAD encryption procedure using $\mathsf{pk}_{\mathcal{R}}$, deriving $K_i^{(r)}$ via KDF and computing $C_i^{(r)}$ as in the real protocol;
- it derives the round MAC key $k_G^{(r)} = \mathsf{KDF}(K_G, r)$ and computes the tag $\mathsf{tag}_i^{(r)} = \mathsf{MAC}_{k_G^{(r)}}(C_i^{(r)} \,\|\, r)$.

Since Sim_S follows exactly the same algorithms as the corrupted parties, the distribution of $(C_i^{(r)}, \mathsf{tag}_i^{(r)})$ in the simulated view is identical to that in a real execution.

(3) **Simulating broadcast messages.** The broadcast messages in the handover phase (Algorithm 5) depend only on the reconstructed secrets M_j, i.e., on the protocol output. The simulator can therefore reproduce $\mathsf{PSI}^{(r)}$ and its complement exactly as in the real protocol, and the corrupted participants' local updates of LastPSI and their local computations $\left(\mathsf{PSI}^{(r)}\right)^{\mathsf{c}} \cap S_i^{\mathsf{c}}$ follow deterministically.

Thus, for any coalition of fewer than t participants, Sim_S can reconstruct their entire view from their inputs and the protocol output, and this view is identically distributed to that in a real execution. In particular, the coalition learns nothing about any unreconstructed secret M_j beyond what is implied by the public transcript and the reconstructed values.

Conclusion. In both adversarial cases, a simulator can reproduce the adversary's view using only the protocol output and public information. Therefore, the protocol is semi-honest secure. □

6.3 Security Against MITM Adversaries

We now discuss security against an active network adversary mounting man-in-the-middle (MITM) attacks on the communication between participants P_i and the Reconstructor $\mathcal{R}$. We assume that the adversary can arbitrarily intercept, drop, delay, reorder, and modify messages on the network, but does not compromise the long-term keys $(\mathsf{sk}_{\mathcal{R}}, K_G)$ nor the internal state of honest parties.

Theorem 6.2 (MITM robustness of the communication layer)**.** *Under the assumptions of Section 6.1, any polynomial-time man-in-the-middle adversary controlling the network cannot:*

(1) *learn any additional information about the plaintext vectors $E_i^{(r)}$ beyond what is implied by the protocol outputs; nor*
(2) *inject or modify messages so that they are accepted by $\mathcal{R}$ as valid ciphertexts for a given round r, except with negligible probability.*

Proof. We separate confidentiality and integrity/authenticity.

Confidentiality. Each message from P_i to $\mathcal{R}$ is encrypted using a fresh ephemeral KEM keypair and an AEAD scheme under a key derived from the shared secret ss via the KDF. By IND-CCA security of the KEM+AEAD public-key encryption scheme, even a fully active network adversary who does not know $\mathsf{sk}_{\mathcal{R}}$ cannot distinguish the ciphertexts from encryptions of random messages. Hence, the adversary learns no additional information about $E_i^{(r)}$ beyond what is revealed by the protocol output.

Integrity and authenticity. There are two layers of integrity:
- *AEAD integrity (INT-CTXT):* any modification of the ciphertext component $C_{\mathsf{sym},i}^{(r)}$ or of the associated data $AD_i^{(r)}$ will cause AEAD.Dec to fail, except with negligible probability. Thus, an adversary cannot alter the encrypted vector $E_i^{(r)}$ without being detected.
- *Round MAC under $k_G^{(r)}$:* even if the adversary replaces the entire ciphertext $C_i^{(r)}$ with an arbitrary value, it must also produce a valid tag $\mathsf{tag}_i^{(r)} = \mathsf{MAC}_{k_G^{(r)}}(C_i^{(r)} \,\|\, r)$ in order for $\mathcal{R}$ to accept the message. By EUF-CMA security of the MAC and the fact that $k_G^{(r)}$ is unknown to the adversary, the probability of forging such a tag is negligible.

Since the adversary only observes ciphertexts and tags, and does not know $\mathsf{sk}_{\mathcal{R}}$ nor the round keys derived from K_G, it cannot extract any information about $E_i^{(r)}$ beyond what is revealed by the protocol output. Therefore, any attempt to inject or modify messages so that they are accepted as valid by $\mathcal{R}$ succeeds only with negligible probability.

Conclusion. Combining confidentiality (IND-CCA KEM+AEAD) with integrity and authenticity (INT-CTXT AEAD + EUF-CMA MAC), we obtain robustness against man-in-the-middle attacks on the communication channel between P_i and $\mathcal{R}$ in the stated model. □

7 Evaluation

We implement the protocol and evaluate it systematically to support a comprehensive validation of both correctness and performance. The implementation is written in C++ and relies on cryptographic primitives provided by BoringSSL[22]. In particular, we analyze the scaling behavior of: (i) Key-Holder parameter generation, (ii) Participant row encryption, (iii) Reconstructor row decryption and reconstruction. These experiments are performed across controlled variations of core system parameters, including input set size, number of participants, and minimum reconstruction threshold, thus supporting a more complete and rigorous evaluation of the protocol's practical behavior. The full anonymized code for the test implementation is made available on Figshare[1].

Experimental Setup. We performed our tests on an MSI GF63 laptop, running an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz with 12 logical cores and 16 GB of RAM.

We have additionally tested all computations performed by the Participant role on a Raspberry Pi 400, with a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz and 4 GB of RAM.

We have timed a number of critical areas of the proposed protocol across various parameter values, demonstrating the existing bottlenecks and the performance scaling at different parameter configurations. These practical results align with the expected computational complexity of each step of the protocol.

[1] https://figshare.com/s/631e6018c63d13d1fb0c

Table 1: Memory overhead of current implementation for key data structures. Column 3 contains practical figures for $(N = 64, L = 1000, R = 10)$.

Individual share	s	$113B$
Participant row for round r	$L \cdot s$	$113KB$
Participant Packet	$MAC(Enc(L \cdot s) + AD)$	$113.1KB$
Participant Share Matrix	$R \cdot L \cdot s$	$1.13MB$
Reconstructor Share Matrix	$N \cdot L \cdot s$	$7.23MB$
Full Keyholder Matrix	$R \cdot N \cdot L \cdot s$	$72.3MB$
PSI	L bits	$125B$

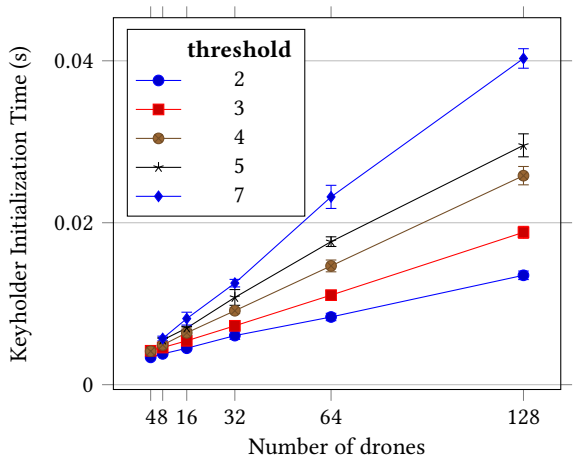

Figure 1: Performance comparison of KH parameter generation over number of participants.

7.1 Experiments

The first evaluated component of the protocol is the key-holder initialization and parameter generation procedure outlined in Algorithm 1. As expected, apart from generating fixed-size elements such as the Reconstructor key pair, the algorithm's cost is dominated by the construction of the share matrix for each participant. In practice, the key-holder generates a degree $t-1$ polynomial per coordinate, per round, and per participant. Therefore, the computational effort grows linearly with each of these parameters. We perform 100 KeyHolder initializations for $L = 1000$ coordinates, up to 128 participants N, and up to 7 threshold t. Figure 1 presents the average execution time, highlighting this linear relationship with respect to the number of participants. Standard deviation is lower than 10% of total runtime for all configurations tested, dropping below 5% in all configurations with 128 participants.

Next, we evaluate the Exploration phase row encryption and decryption procedures detailed in Algorithm 3. As indicated by the algorithm, each Participant independently encrypts its own vector (i.e., *row*) while the Reconstructor processes decryption on a per-row basis. Consequently, the computational cost scales solely with the size of the input, namely the maximum number of coordinates, as indicated in Table 1. Other protocol parameters, including the threshold value, do not affect the performance of these operations.

For *row encryption*, which is executed by the Participant, the experiments were conducted on a Raspberry Pi 400, representative of a typical *on-board companion computer* used in UAV applications. For all other tests where our model does not expect a constrained computing model we utilize the MSI GF63 laptop. In both cases, we perform a statistical analysis over 100 runs of the individual Participant process for encryption, and Reconstructor process for decryption. The corresponding performance results are presented in Figure 2. The standard deviation on these tests was also consistently below 10% of average runtime.

Finally, we tested end-of-round matrix reconstruction, as described in Algorithm 4. This is the most significant bottleneck of the algorithm, as the Reconstructor is required to test $\binom{N}{t}$ subsets of potential shares for each coordinate. This formulation grows exponentially with the threshold t, which must therefore be chosen carefully depending on the computing capabilities of the Reconstructor in a practical setting.

Figure 3 contains a log-scale graph detailing reconstruction time per individual coordinate. We perform the average over 1000 coordinates for parametrizations where the resulting processing time is feasible. For extreme cases, we extrapolate the average cost per subset verification τ_t and compare our predicted result with the per-coordinate computing cost on a smaller number of coordinates over 24 hours of runtime. In the case of $N = 128, t = 7$, for instance, the average worst-case runtime (i.e., the average cost to evaluate every possible subset combination of a coordinate) in our testing setup for each coordinate is 14959.3 seconds (slightly over 4 hours), but each coordinate processed allows us to collect $\binom{128}{7} = 94525795200$ timing samples for subset verification. In all cases tested, our standard deviation is within 5% of the average value reported, and under 2% on all tests above 32 participants.

7.2 Results and Discussion

To assess the scalability of the proposed system we quantify: (i) the storage requirements associated with each operational role, (ii) the bandwidth, and (iii) computing time.

Storage Constraints: Table 1 contains the detailed formulas to determine the main storage constraints of each role, plus an additional column demonstrating their values in our current implementation for $N = 64, L = 1000, R = 10$ as our reference parametrization, where N is the number of Participants, L is the number of coordinates, and R is the number of rounds. These parameters were selected to reflect realistic large-scale deployment scenarios while preserving computational feasibility.

In our implementation, each role's local storage is dominated by the corresponding share matrix, with less than 1 kB taken by other fixed-size elements such as keypairs. Consequently, the estimated storage at our chosen parameters is 1.13 MB, 72.3 MB and 7.23 MB for Participants, KeyHolders, and Reconstructors respectively.

The scaling factors described in the second column of Table 1 may be used to adjust the storage, bandwidth and computing time estimates corresponding to each role depending on practical application requirements. As we will discuss further in following paragraphs, the effects of fixed overhead factors is very limited compared to the scaling factors of these parameters. This is especially relevant for computing time measurements.

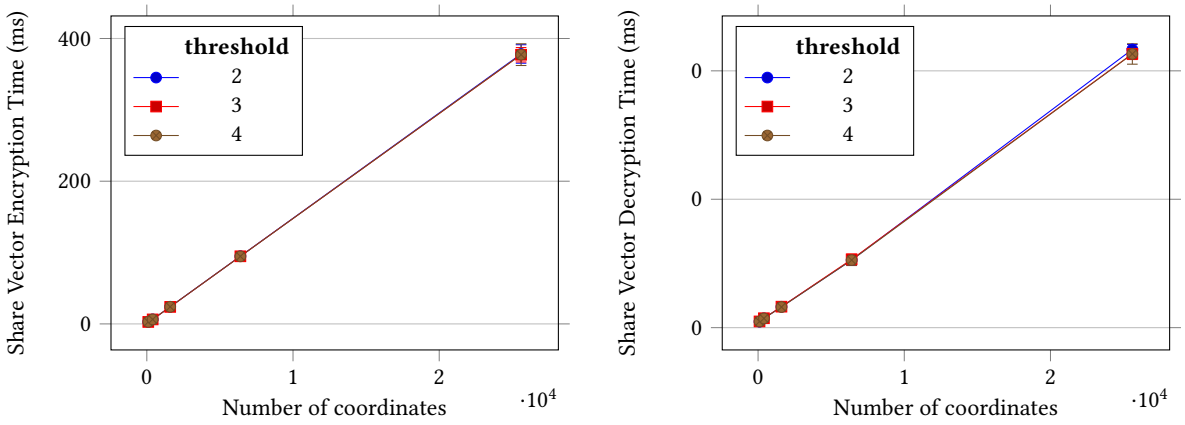

Figure 2: Performance comparison of participant's encryption of one share vector and the corresponding reconstructor's decryption over number of coordinates.

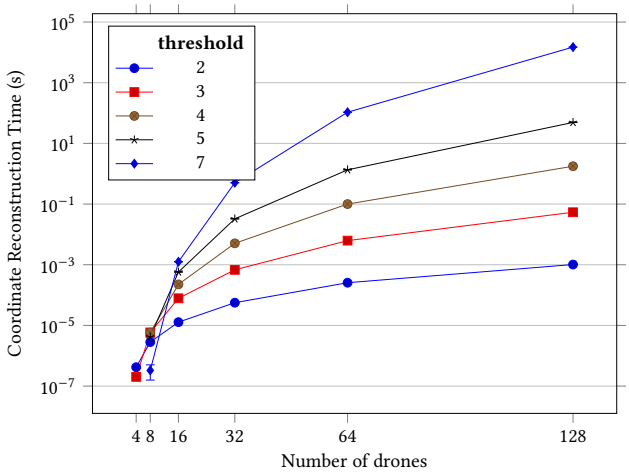

Figure 3: Performance comparison of Reconstructor matrix reconstruction over number of participants.

Bandwidth: Bandwidth-wise, the overhead added by each participant's encryption of the current round's potential shares is minimal at 100 bytes. With our standard parameters, each drone transmits a single message of 113.1 kB each round. Based on existing literature and technical documents, we estimate that on a simple on-board computer such as a Raspberry Pi 4, encryption of this payload via the chosen AES-GCM implementation would require 135-145 mJ [14] , while transmission over 802.11ac WiFi at a typical 10 $MBps$ data rate would require roughly 30-50 mJ [24]. On real hardware in a practical scenario, these energy figures may vary significantly depending on a combination of electronic and cyber-physical factors such as available antennas, radio communication standards, data rate, high-power states and transmission power, network congestion, obstacles, and other environmental factors. For instance, the Cortex A53 CPU available on the Raspberry Pi 3 models, while somewhat similar in terms of throughput [46], is significantly less power efficient in GCM encryption, consuming up to 94% extra energy per kB [14]. Also note that the versions of ARM Cortex A53 and A72 installed on the Raspberry Pi 3 and 4 respectively do not offer hardware extensions to accelerate AES encryption and decryption.

On the Reconstructor side, we receive N packets, totaling roughly 7.23 MB, and transmit a single packet containing the current PSI. Energy measurements for the Reconstructor are not relevant to this discussion as it is not considered a constrained device, and are difficult to estimate regardless because of the significant incentive to parallelize and batch its workload as we discuss later.

Computing Time: The results of the timing tests described in Section 7.1 are available in Figures 1, 2, and 3. We rely on the scaling factors in Table 1 in order to focus on the most critical parametrization for each timing test. More specifically, we focus on threshold values and number of participants for initialization and reconstruction, while for encryption and decryption of share vectors we focus on number of coordinates, as payload size is solely dependent on that parameter.

Our experiments confirm the effectiveness of excluding the constrained hardware sections of our system from the most computationally intensive tasks. In our tests, even at the largest map size considered (25600 coordinates), encryption is resolved in the order of the tenths of second on our Raspberry Pi 400 test environment.

In terms of overall computing time, the clear bottleneck of the entire protocol is the reconstruction algorithm. The protocol initialization process may encounter scaling issues when processing large maps for a large number of parties, but those same parameters are significantly more problematic when considering the effective scaling of matrix reconstruction. However, the most critical parameter when considering reconstruction is the minimum threshold value t.

While the time to process t-degree polynomials grows slightly with t, from roughly 1.3 μs at threshold 2 to to 1.7 μs at threshold 7, the most impactful component of the complexity formula discussed in Section 5.4 remains the $\binom{N}{t}$ combinations which form the potential subsets to be tested.

Threshold values should be set depending on the application-specific required level of redundancy for each region, acceptable

wait times, number of participants, and available computing power. On our testing hardware, at $t = 5$ and $N = 64$, we can guarantee worst-case performance of 1.12 seconds per coordinate evaluation. However, similar performance can be achieved at $t = 7$ over 32 participants, or at $t = 4$ over 128 participants.

Although these scaling limitations are substantial, in practice their impact is reduced through several optimizations that preserve the protocol's cryptographic soundness.

First, computations for individual coordinates are fully independent. This independence enables parallel execution of all algorithms, with the exception of the vector encryption and decryption primitives. The most pronounced performance gains arise in the initialization and reconstruction phases, where per-coordinate parallelization can significantly reduce overall runtime.

Second, during reconstruction, each candidate subset can be evaluated independently. As soon as one subset succeeds, processing for that coordinate can terminate immediately. This short-circuit behavior yields considerable speedups, particularly when the reconstruction workload is partitioned into batches and distributed across multiple computational nodes or endpoints.

Finally, multi-round executions introduce additional optimization opportunities. As $PSI^{r-1} \subset PSI^r$, at round r the Reconstructor can skip calculations for any coordinate $j \in PSI^{r-1}$ entirely. Each participant is already serving dummies for those coordinates, in order to protect themselves more effectively from differential attacks such as the one discussed in the handover paragraph in Section 5.2.

8 Ethical Considerations

This work is conceived for civilian, multi-stakeholder contexts such as disaster response, environmental monitoring, and cooperative inspection. Although privacy-preserving coordination mechanisms can, in principle, be applied to more sensitive domains (e.g, surveillance, crowd sensing, or military operations), our protocol is not designed or intended to deliver operational advantages in adversarial scenarios. Its contribution lies in improving the efficiency and robustness of swarm-coordination tasks, without providing additional capabilities or intelligence over the underlying data.

A recognised limitation arises from the architectural requirements of the protocol. The low-trust cooperative setting of our proposed protocol is by default not aligned with the organization standards of cooperative defense missions, being our protocol secure in the semi-honest model. These assumptions are incompatible with adversarial or contested environments.

Moreover, the protocol does not provide meaningful benefits to external adversaries or trespassers. Its privacy guarantees are targeted at preventing insider inference attacks among cooperating entities, not at enabling covert or unauthorized mapping. The system is therefore aligned with civilian cooperation settings where transparency, accountability, and controlled data sharing are essential.

9 Conclusion and Future Works

This work introduces a fully anonymous threshold-PSI protocol tailored for distributed exploration tasks in adversarial and strategically sensitive environments. By combining Shamir secret sharing, a YOSO communication architecture, and anonymous authenticated encryption, our solution achieves threshold correctness without persistent identities, linkable communication patterns, or authenticated channels tied to long-term keys. The protocol supports multi-agent scenarios, such as drone-based reconnaissance, where drones must collaboratively determine which zones have been explored by at least t participants while preserving operational privacy. We provide formal security guarantees against semi-honest participants and active man-in-the-middle adversaries, showing that the protocol ensures threshold privacy, sender anonymity, ciphertext unlinkability, and robustness of the communication layer. The resulting design demonstrates that anonymous threshold-PSI can be realized efficiently within a YOSO framework, enabling privacy-preserving coordination even in high-risk settings where structural anonymity is essential.

Future Works. A promising direction for future research concerns the resilience of the proposed framework against cyber-physical attacks. In particular, external sensing infrastructures such as radar systems, RF localization techniques, and acoustic sensors, may infer drone positions or movement patterns independently of the communication-layer protections considered in this work. Moreover, regulatory frameworks in several countries require drones to continuously broadcast their position information [19], similarly to ADS-B transponders in civil aviation. Recent studies on location-privacy mechanisms aim to mitigate these risks while remaining compliant with legal requirements and enabling trespasser detection [8, 51]. Addressing such threats will require the integration of cross-layer countermeasures that combine secure communications, mobility obfuscation, and physical stealth techniques. Although a detailed analysis of these attack vectors is beyond the scope of the present study, these aspects are currently being investigated as part of our ongoing research.

Another relevant research direction involves the integration of advanced collision-detection and avoidance mechanisms into the proposed framework. The current work assumes standard navigation and built-in safety systems, without explicitly addressing complex collision scenarios that may arise in highly dynamic or dense operational environments. Large-scale deployments, in particular, may require predictive and cooperative avoidance strategies capable of coping with communication delays and rapidly changing trajectories. The design and evaluation of such advanced collision-management solutions remain outside the scope of this work, but they represent an active area of ongoing investigation.

Finally, future work will also focus on evaluating system performance under limited-connectivity conditions, where reconstructors must operate with partial information, as well as enabling participants to independently perform reconstruction operations. These developments would move the framework toward a fully distributed architecture, supporting autonomous and decentralized local exploration by each participant.

Acknowledgments

The authors used generative AI-based tools to revise the text, improve flow and correct any typos, grammatical errors, and awkward phrasing.

This work was supported by Italian National Cybersecurity Agency (Agenzia per la Cybersicurezza Nazionale) under the programme for promotion of XL cycle PhD research in cybersecurity – CUP B83C24005650005, B83C24005670005, project "DANTE" under the European Union's Digital Europe Programme (DIGITAL) 2021–2023, grant agreement no 101093913, and project "SERICS" (PE00000014) under the NRRP MUR program funded by the EU-NGEU. The views expressed are those of the author only, not of the funding agencies.

References

[1] 2007. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.* Technical Report SP 800-38D. NIST. https://csrc.nist.gov/publications/detail/sp/800-38d/final

[2] 2015. *Secure Hash Standard (SHS).* Technical Report FIPS PUB 180-4. NIST. https://csrc.nist.gov/publications/detail/fips/180/4/final

[3] Allon Adir, Ehud Aharoni, Nir Drucker, Eyal Kushnir, Ramy Masalha, Michael Mirkin, and Omri Soceanu. 2022. Privacy-preserving record linkage using local sensitive hash and private set intersection. In *International Conference on Applied Cryptography and Network Security*. Springer, 398–424.

[4] Aslı Bay, Zekeriya Erkin, Jaap-Henk Hoepman, Simona Samardjiska, and Jelle Vos. 2021. Practical multi-party private set intersection protocols. *IEEE Transactions on Information Forensics and Security* 17 (2021), 1–15.

[5] Aner Ben-Efraim, Olga Nissenbaum, Eran Omri, and Anat Paskin-Cherniavsky. 2022. PSimple: Practical multiparty maliciously-secure private set intersection. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. ACM, 1098–1112.

[6] Dan Boneh, Sam Eskandarian, Lukas Hanzlik, Zahra Jafargholi, Hooman Malvai, Ilya Mironov, Tal Rabin, Leonid Reyzin, Pratyush Vasudevan, and Daniel Wichs. 2021. You-Only-Speak-Once: Secure MPC with Stateless Parties. In *Advances in Cryptology – CRYPTO 2021*. Springer, 337–366.

[7] Abdelmalek Bouguettaya, Hafed Zarzour, Amine Mohammed Taberkit, and Ahmed Kechida. 2022. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. *Signal Processing* 190 (2022), 108309.

[8] Alessandro Brighente, Mauro Conti, and Savio Sciancalepore. 2022. Hide and seek: privacy-preserving and FAA-compliant drones location tracing. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*. 1–11.

[9] Alessandro Brighente, Mauro Conti, Savio Sciancalepore, and Harshul Vaishnav. 2025. Strangers Sets: Preserving Drones' Location Privacy while Avoiding Violations of Critical Infrastructures. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*. ACM, 1560–1569.

[10] Giuseppe Bruno, Diana Nicoletti, Monica Scannapieco, and Diego Zardetto. 2018. Privacy Preserving Set Intersection.. In *Proceedings of the Ninth Irving Fisher Conference: Bank of International Settlements, Basel*. 30–31.

[11] Eduardo Castelló Ferrer, Thomas Hardjono, Alex Pentland, and Marco Dorigo. 2021. Secure and secret cooperation in robot swarms. *Science Robotics* 6, 56 (2021), eabf1538.

[12] Nishanth Chandran, Nishka Dasgupta, Divya Gupta, Sai Lakshmi Bhavana Obbattu, Sruthi Sekar, and Akash Shah. 2021. Efficient linear multiparty PSI and extensions to circuit/quorum PSI. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1182–1204.

[13] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The second-generation onion router. In *USENIX Security*.

[14] Gene Dupré. 2024. Energy efficiency in AES encryption on ARM Cortex CPUs: Comparative analysis across modes of operation, data sizes, and key lengths.

[15] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. 2015. Multimodal deep learning for robust RGB-D object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 681–687.

[16] European Commission. 2022. A Drone Strategy 2.0 for a Smart and Sustainable Unmanned Aircraft Ecosystem in Europe. Commission Staff Working Document. COM(2022) 652 final.

[17] European Union Aviation Safety Agency. 2024. Easy Access Rules for U-space. EASA Online Publication. https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-u-space Consolidated version of Regulations (EU) 2021/664, 2021/665 and 2021/666.

[18] Charles R. Farrar and Keith Worden. 2012. *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons, Chichester, West Sussex, United Kingdom.

[19] Federal Aviation Administration. 2021. Remote Identification of Unmanned Aircraft. Federal Register, 86 FR 4390, 14 CFR Parts 1, 11, 47, 48, 89, 91, and 107. https://www.federalregister.gov/documents/2021/01/15/2020-28948/remote-identification-of-unmanned-aircraft

[20] Michael J Freedman, Kobbi Nissim, and Benny Pinkas. 2004. Efficient private matching and set intersection. In *International conference on the theory and applications of cryptographic techniques*. Springer, 1–19.

[21] Oded Goldreich and Ronen Vainish. 1987. How to Solve any Protocol Problem - An Efficiency Improvement. In *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology (CRYPTO '87)*. Springer-Verlag, 73–86.

[22] Google Inc. 2026. BoringSSL. https://boringssl.googlesource.com/boringssl/. Accessed: 2026-02-27.

[23] Jingwei Hu, Zhiqi Liu, and Cong Zuo. 2025. Delegated multi-party private set intersection from secret sharing. *Cryptology ePrint Archive* (2025).

[24] Infineon Technologies AG. 2021. *CYW43455 Single-Chip 5G WiFi IEEE 802.11n/ac MAC/Baseband/Radio with Integrated Bluetooth 5.0*. Infineon Technologies AG. https://www.infineon.com/assets/row/public/documents/30/49/infineon-cyw43455-datasheet-en.pdf Document Number: 002-15051 Rev. *O.

[25] Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. 2020. On deploying secure computing: Private intersection-sum-with-cardinality. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 370–389.

[26] Shu-Jun Ji, Qing-Hua Ling, and Fei Han. 2023. An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information. *Computers and Electrical Engineering* 105 (2023), 108490.

[27] Yong-il Jo, Seonah Lee, and Kyong Hoon Kim. 2020. Overlap avoidance of mobility models for multi-UAVs reconnaissance. *Applied Sciences* 10, 11 (2020), 4051.

[28] Lea Kissner and Dawn Song. 2004. *Private and threshold set-intersection*. Technical Report.

[29] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. 1997. HMAC: Keyed-Hashing for Message Authentication. RFC 2104. https://www.rfc-editor.org/rfc/rfc2104

[30] Hugo Krawczyk and Pasi Eronen. 2010. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869. https://www.rfc-editor.org/rfc/rfc5869

[31] Adam Langley, Mike Hamburg, and Sean Turner. 2016. Elliptic Curves for Security. RFC 7748. https://www.rfc-editor.org/rfc/rfc7748

[32] Tancrede Lepoint, Sarvar Patel, Mariana Raykova, Karn Seth, and Ni Trieu. 2021. Private join and compute from PIR with default. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 605–634.

[33] Rasoul Akhavan Mahdavi, Thomas Humphries, Bailey Kacsmar, Simeon Krastnikov, Nils Lukas, John A Premkumar, Masoumeh Shafieinejad, Simon Oya, Florian Kerschbaum, and Erik-Oliver Blass. 2020. Practical over-threshold multi-party private set intersection. In *Proceedings of the 36th Annual Computer Security Applications Conference*. 772–783.

[34] Nathalie Majcherczyk, Daniel Jeswin Nallathambi, Tim Antonelli, and Carlo Pinciroli. 2021. Distributed Data Storage and Fusion for Collective Perception in Resource-Limited Mobile Robot Swarms. *IEEE Robotics and Automation Letters* 6, 3 (2021), 5549–5556.

[35] Saurabh Mishra, Samuel Rodriguez, Marco Morales, and Nancy M Amato. 2016. Battery-constrained coverage. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 695–700.

[36] Yoav Nir and Adam Langley. 2018. ChaCha20 and Poly1305 for IETF Protocols. RFC 8439. https://www.rfc-editor.org/rfc/rfc8439

[37] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. 2015. Phasing: Private set intersection using permutation-based hashing. In *24th USENIX Security Symposium (USENIX Security 15)*. 515–530.

[38] Ania M Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. 2017. The loopix anonymity system. In *26th usenix security symposium (usenix security 17)*. 1199–1216.

[39] Davy Preuveneers and Wouter Joosen. 2022. Privacy-preserving polyglot sharing and analysis of confidential cyber threat intelligence. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*. 1–11.

[40] Phillip Rogaway. 2002. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (Washington, DC, USA) *(CCS '02)*. Association for Computing Machinery, New York, NY, USA, 98–107. https://doi.org/10.1145/586110.586125

[41] Mike Rosulek and Ni Trieu. 2021. Compact and malicious private set intersection for small sets. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1166–1181.

[42] Savio Sciancalepore and Dominik Roy George. 2022. Privacy-preserving trajectory matching on autonomous unmanned aerial vehicles. In *Proceedings of the 38th Annual Computer Security Applications Conference*. ACM, 1–12.

[43] Hazim Shakhatreh, Abdallah Khreishah, Jacob Chakareski, Haythem Bany Salameh, and Issa Khalil. 2016. On the continuous coverage problem for a swarm of UAVs. In *2016 IEEE 37th Sarnoff Symposium*. IEEE, 130–135.

[44] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.

[45] Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. 2023. Lightweight privacy-preserving proximity discovery for remotely-controlled drones. In *Proceedings of the 39th Annual Computer Security Applications Conference*. ACM, 178–189.

[46] Thierry Lelégard. 2026. *AES Benchmarks*. https://github.com/lelegard/aesbench Accessed: 2026-05-21.

[47] UAViators Humanitarian UAV Network. 2016. Humanitarian UAV Code of Conduct and Guidelines. Online report. Developed in consultation with UN OCHA, WFP, UNHCR, ICRC and other humanitarian organizations.
[48] Jelle Van Den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. 2015. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*. 137–152.
[49] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024. Yolov10: Real-time end-to-end object detection. *Advances in neural information processing systems* 37 (2024), 107984–108011.
[50] Michael A Wulder, Jeffrey G Masek, Warren B Cohen, Thomas R Loveland, and Curtis E Woodcock. 2012. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sensing of Environment* 122 (2012), 2–10.
[51] Aiting Yao, Shantanu Pal, Xuejun Li, Zheng Zhang, Chengzu Dong, Frank Jiang, and Xiao Liu. 2024. A privacy-preserving location data collection framework for intelligent systems in edge computing. *Ad Hoc Networks* 161 (2024), 103532.
[52] Andrew C Yao. 1982. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 160–164.
[53] Kassaye Yitbarek Yigzaw, Antonis Michalas, and Johan Gustav Bellika. 2017. Secure and scalable deduplication of horizontally partitioned health data for privacy-preserving distributed statistical computation. *BMC medical informatics and decision making* 17, 1 (2017), 1.