Distributed Coordination and Security in Autonomous Drone Swarms: A Literature Survey

John Brittain
Dept. of Electrical and Computer Engineering
Iowa State University
Ames, Iowa
jdbritt@iastate.edu

May 3, 2026

***Abstract*—Autonomous drone swarms are one of the most compelling real-world applications of distributed systems principles, requiring groups of independent computing nodes to reach consensus, tolerate failures, and communicate under adversarial conditions, all in real time. This survey reviews ten recent works that cover swarm coordination architectures, communication overhead, security threats, and emerging defenses. Three major themes emerge: (1) decentralized and hybrid control architectures offer the best scalability and fault tolerance but increase algorithmic complexity; (2) distributed machine learning techniques, including federated learning and multi-agent reinforcement learning, enable adaptive coordination that static protocols cannot match; and (3) security remains the most open challenge, with no standardized framework and large gaps between current countermeasures and realistic threat models. Key unsolved problems include the gap between small-scale simulation results and large operational deployments, the trade-off between cryptographic overhead and real-time coordination latency, and the lack of shared benchmarks that allow fair comparison between research groups.**

***Index Terms*—drone swarms, UAV coordination, distributed systems, swarm security, federated learning, multi-agent systems**

I. Introduction

Unmanned aerial vehicles (UAVs), commonly called drones, have become practical tools in many fields. A single drone can inspect a structure, photograph a crop field, or deliver a package. However, when many drones operate together as a coordinated *swarm* , their collective capability grows far beyond what any single unit can do. Swarms of hundreds or thousands of drones can simultaneously cover large areas, maintain complex three-dimensional formations, and divide tasks among themselves dynamically [1], [4].

From a computer engineering perspective, a drone swarm is a distributed system. Each drone has its own processor, sensors, memory, and wireless radio, and no single node holds a global view of the system. Drones must coordinate by passing messages across a wireless mesh network, which is exactly the scenario covered in a course on distributed systems. The classic problems appear immediately: how do nodes reach agreement when messages can be delayed or dropped? How does the system keep running when nodes fail? How does the system defend itself when the network is open to attack?

Two challenges dominate the literature. The first is *coordination*: getting all drones to continuously agree on positions, velocities, task assignments, and mission state even as individual nodes drop out and network topology changes. The second is *security*: protecting the wireless infrastructure from attackers who may spoof GPS signals, jam links, or inject malicious drones into the swarm.

This survey reviews ten recent works that address these challenges, covering technical mechanisms and broader policy context. Section II gives background on applications. Section III analyzes coordination architectures. Section IV covers communication and distributed learning. Section V examines security. Section VI critiques gaps in the field. Section VII concludes.

II. Swarm Applications and Motivation

Understanding real-world deployment scenarios helps clarify what constraints coordination and security solutions must satisfy.

Spinko provides a useful industry-level overview of current swarm deployments [1]. Agriculture is a primary application: swarms coordinate to monitor crop health, apply pesticides with precision, and cover large fields faster than any single unit. Humanitarian demining deploys swarms to scan wide areas for buried ordnance more safely than ground teams. Logistics applications coordinate multi-drone deliveries across distributed waypoints. Emergency response scenarios such as wildfire assessment, terrain mapping, and search-and-rescue let a swarm cover large regions simultaneously while aggregating sensor data from each node [8].

Military applications have driven major investment. A 2023 U.S. Government Accountability Office (GAO) report on drone swarm technologies notes that both civilian and defense sectors are actively exploring swarm deployments [4]. The core appeal for defense use is distributing risk across many low-cost units, making a swarm harder to neutralize than a single expensive platform. Alqudsi and Makaraci note that military and commercial requirements diverge sharply: commercial swarms operate in relatively benign, GPS-available environments, while military swarms must function in GPS-denied, jammed, and actively adversarial conditions [7].

These use cases impose distinct requirements on the underlying distributed systems design and motivate why both coordination and security must be treated as first-class engineering concerns rather than afterthoughts.

III. Coordination Architectures

A. Centralized, Decentralized, and Hybrid Control

The most fundamental design question for a drone swarm is where control authority lives. In a *centralized* architecture, a ground station or designated leader drone makes all decisions and broadcasts commands to followers. This is straightforward to implement and can compute globally optimal plans, but the controller is a single point of failure, and as swarm size grows, the communication load at the leader grows linearly, becoming a bottleneck [1].

In a *decentralized* architecture, each drone makes autonomous decisions based on local sensor data and messages from nearby neighbors. No global controller exists; coherent behavior emerges from consistent local rules applied by every node. This mirrors the design of many large-scale distributed systems and is inherently fault-tolerant, since no single node is critical. The tradeoff is increased algorithmic complexity: achieving global objectives from local rules is harder to design and harder to verify [8].

Hybrid leader-follower hierarchies combine both approaches. A small set of designated leader nodes manages high-level coordination, while follower nodes execute simpler local rules. Adoni et al. design, implement, and flight-test a self-organizing UAV swarm based on the leader-follower paradigm, demonstrating that hierarchical organization reduces the consensus overhead faced by each node while preserving meaningful fault tolerance at the follower level [10]. Alqudsi and Makaraci review the literature and conclude that hybrid architectures suit most operational scenarios because they balance scalability, simplicity, and resilience [7].

B. Consensus and Formation Control

Within any architecture, drones must continuously solve a form of consensus: all nodes must agree on a shared reference frame so that formation geometry is maintained as positions change and messages arrive out of order.

Ling et al. model and simulate cooperative planning and perception algorithms for distributed UAV swarms, analyzing how the structure of the communication graph affects coordination performance [2]. Their results show that the algebraic connectivity of the network, specifically the second eigenvalue of the graph Laplacian, determines how quickly consensus converges. Sparse networks converge slowly; dense networks consume more bandwidth. This fundamental tradeoff between connectivity and communication cost appears throughout the swarm literature and closely parallels similar results in distributed sensor networks and multi-robot systems.

Formation control must also handle collision avoidance. Chen et al. address this with safe reinforcement learning, training drones to plan collision-free paths while encoding safety constraints directly into the reward function [9]. Their approach outperforms classical potential-field methods, which suffer from local minima that can trap drones in deadlocked configurations. This illustrates a broader trend: learning-based methods are beginning to outperform hand-engineered algorithms for coordination tasks that are difficult to specify analytically. Pommeranz et al. reinforce this point by proposing a modular ROS 2 and PX4-Autopilot architecture for heterogeneous UAV swarms, showing that componentized software design enables scalable formation flight and leader-follower coordination across diverse hardware configurations [3].

IV. Communication and Distributed Learning

A. Wireless Network Challenges

Swarm communication relies on mobile ad hoc networks (MANETs): self-organizing wireless networks with no fixed infrastructure. As drones move, topology changes constantly, requiring adaptive routing. Standard MANET protocols were designed for slower, ground-based networks and perform poorly in three-dimensional, high-mobility swarm environments [1].

Overhead scales with swarm size. Broadcasting position updates to all nodes costs $O(n)$ messages per cycle, making full state synchronization impractical for large swarms. Gossip-based protocols limit overhead by propagating updates to a random subset of neighbors, achieving eventual consistency at lower cost. Hierarchical clustering (grouping drones into clusters with elected leaders that relay inter-cluster state) further reduces network load at the cost of some coordination latency. Reed highlights that real-time data sharing and bandwidth management remain among the most significant practical obstacles to large-scale autonomous swarm operation [8].

B. Distributed Machine Learning

An important recent development is using distributed machine learning to extend or replace classical coordination protocols. Ding et al. provide a comprehensive 2023 survey of these techniques applied to UAV swarms [5], covering federated learning, multi-agent reinforcement learning (MARL), distributed inference, and split learning.

Federated learning lets each drone train a local model on its own sensor data and share only model updates with neighbors, not raw data streams. This reduces bandwidth and keeps sensitive data on-device, which matters when drones are collecting imagery or location data. The shared model can improve object detection, channel estimation, or coordination policy quality across the whole swarm.

MARL trains drones to discover cooperative policies through reward signals, adapting to dynamic environments without hand-programmed coordination rules. The main challenge is non-stationarity: each drone's effective environment changes as its neighbors also update their policies, which can destabilize joint training. Centralized training with decentralized execution (CTDE) addresses this by coordinating learning offline in simulation, then deploying independent policies to each drone's hardware. Ding et al. also identify semantic communication as a promising direction: transmitting only

task-relevant abstractions rather than raw sensor data dramatically compresses bandwidth requirements for perception-heavy tasks [5].

Chen et al.'s safe RL path-planning work connects directly to this theme [9]: by learning collision-avoidance policies rather than programming them explicitly, each drone's onboard model adapts to the density and behavior of its neighbors in ways that fixed rules cannot.

V. Security in Drone Swarms

A. Attack Surface

Drone swarms inherit all the security vulnerabilities of wireless distributed systems, plus some that are specific to their autonomous, physically-moving nature.

The threat landscape for drone swarms spans several categories [7], [8]. *GPS spoofing* transmits fake satellite signals that cause drones to compute incorrect positions. Since most coordination depends on accurate localization, a successful spoof can redirect an entire swarm without any visible sign of intrusion. Alqudsi and Makaraci identify GPS spoofing as one of the most significant unresolved vulnerabilities facing operational swarm systems [7].

Jamming degrades or cuts wireless links, preventing consensus and potentially fragmenting a swarm into isolated subgroups that can no longer coordinate. Adversaries may combine jamming with spoofing: disabling communication while simultaneously misdirecting surviving nodes. Reed observes that these electronic threats are fundamentally coupled to coordination failures, since disrupting the communication channel also breaks the consensus mechanism the swarm depends on [8].

Malicious node injection is the drone-swarm analog of a Byzantine fault. An adversary introduces a rogue drone that sends inconsistent messages to different neighbors, undermining consensus. Classical Byzantine fault-tolerant (BFT) protocols tolerate up to $\lfloor (n-1)/3 \rfloor$ such nodes, but these protocols impose communication overhead that conflicts with real-time swarm requirements [6], [8].

B. Defense Mechanisms

Several approaches have been proposed to address these threats. Mershad proposes PROACT, a blockchain-based trust protocol for the Internet of Drones that uses parallel multi-miner consensus to confirm transactions securely while minimizing latency and energy overhead [6]. Simulation results show PROACT reduces transaction confirmation time and per-drone energy consumption compared to sequential blockchain approaches, making distributed trust management feasible on resource-constrained UAV hardware. Because no single node controls trust arbitration, PROACT is well-suited to decentralized swarm deployments where a central trust authority would itself be a single point of failure.

Alqudsi and Makaraci survey cryptographic authentication approaches for swarm environments [7], noting that standard public-key infrastructure protocols introduce latency that conflicts with real-time coordination. Lightweight shared-key schemes with periodic re-keying can balance security and performance, but key distribution in a dynamic swarm (where membership changes as drones enter, leave, or fail) remains a difficult open problem.

Intrusion detection systems for swarms analyze collective behavior to flag compromised nodes: if one drone's reported velocity, position, or communication pattern diverges from what its neighbors observe, it is marked for isolation. Critically, these detection systems must themselves operate without a trusted central authority, requiring distributed anomaly detection that can reach consensus on which nodes to exclude. This is effectively a consensus problem nested inside the coordination problem [6], [8].

C. Policy and Governance

Alqudsi and Makaraci observe that policy and standards have not kept pace with technical capabilities [7]. No common security evaluation framework exists for swarm systems, security requirements are not standardized across programs, and there is no established mechanism for enforcing a baseline security posture across development efforts. The GAO's 2023 report on drone swarm technologies similarly highlights that governance frameworks must evolve as deployments grow, noting that technical progress alone is insufficient without coordinated oversight structures [4]. Organizational coordination is ultimately as important as technical coordination.

VI. Critiques and Open Challenges

A. The Simulation-to-Reality Gap

The most consistent gap across the surveyed literature is the limited scale of experimental evaluation. Most coordination and security research is validated in simulation with tens to a few hundred drones [2], [9]. Real-world deployments, however, involve far more nodes, and physical environments introduce channel fading, hardware variability, and adversarial conditions that simulation cannot fully replicate. Chen et al. explicitly acknowledge that transferring their reinforcement learning policy from simulation to hardware remains unsolved [9]. Until solutions are demonstrated at operational scale in real environments, claims of readiness should be treated with caution.

B. Security vs. Coordination Latency

Security mechanisms impose direct costs on coordination performance. Cryptographic authentication adds per-message overhead; BFT consensus requires extra communication rounds; intrusion detection consumes CPU cycles on resource-constrained embedded processors. While PROACT demonstrates that distributed trust mechanisms can be kept lightweight enough for embedded drone hardware [6], the interaction between security cost and coordination latency across a range of swarm sizes and mission profiles has not been comprehensively characterized. This tradeoff is most acute for military swarms that must coordinate in real time while under active electronic attack.

C. Lack of Shared Benchmarks

There is no common benchmark or evaluation framework for drone swarm research. Each group defines its own metrics, swarm sizes, communication models, and attacker models, making cross-paper comparison difficult [7]. Alqudsi and Makaraci identify this absence of standardized evaluation criteria as one of the field's most significant structural gaps, preventing researchers from fairly assessing competing approaches. Establishing community-standard benchmarks analogous to those that advanced the distributed database and network security research communities would accelerate progress significantly.

D. Heterogeneous Swarms

Most surveyed works assume homogeneous swarms in which all drones have identical hardware and capabilities [2], [9]. Operational deployments increasingly involve drones with different sensor payloads, compute budgets, and flight characteristics. The leader-follower architecture in Adoni et al. assumes all nodes can take on either role [10], which does not hold when hardware differs. Pommeranz et al. are a notable exception, explicitly targeting heterogeneous swarms and showing that a modular software architecture can accommodate diverse UAV types without redesigning the coordination layer [3]. Even so, Ding et al. identify hardware heterogeneity as a key open problem for distributed learning in UAV swarms [5], and it equally complicates security policy enforcement across nodes with different compute capabilities.

VII. Conclusion

This literature survey has examined distributed coordination and security in autonomous drone swarms through ten recent works spanning industry analysis, peer-reviewed research, and government policy. Three findings stand out.

First, decentralized and hybrid control architectures are the right long-term direction, and distributed machine learning is expanding what adaptive coordination is achievable. The combination of leader-follower organization, consensus-based formation control, and federated or reinforcement-learned policies represents the current state of the art.

Second, security remains the most open and most critical challenge. The attack surface is wide, countermeasures are not standardized, and evaluation under realistic adversarial conditions is rare. No swarm today can reliably detect and recover from coordinated GPS spoofing combined with malicious node injection.

Third, the gap between what has been demonstrated in simulation and what operational deployments require is the field's most pressing practical problem. Bridging it will require validated real-world experiments, shared benchmarks, and closer collaboration between the research community and operational stakeholders.

As drone swarms grow larger and more autonomous, the distributed systems problems they embody (consensus, fault tolerance, Byzantine resilience, and scalable communication) become both more important and harder to solve. The field is maturing rapidly, but significant work remains before large-scale autonomous swarms can be deployed with confidence in security-critical environments.

References

[1] V. Spinko, "Drone Swarms: How They Actually Work and What Industries Should Care," *The Robot Report*, Aug. 2025. [Online]. Available: https://www.therobotreport.com/drone-swarms-how-they-actually-work-and-what-industries-should-care/

[2] H. Ling, H. Luo, H. Chen, L. Bai, T. Zhu, and Y. Wang, "Modelling and Simulation of Distributed UAV Swarm Cooperative Planning and Perception," *International Journal of Aerospace Engineering*, 2021, doi: 10.1155/2021/9977262.

[3] R. Pommeranz, K. Tebbe, R. Heynicke, and G. Scholl, "A Modular and Scalable System Architecture for Heterogeneous UAV Swarms Using ROS 2 and PX4-Autopilot," arXiv:2510.27327, Oct. 2025. [Online]. Available: https://arxiv.org/abs/2510.27327

[4] U.S. Government Accountability Office, "Science and Technology Spotlight: Drone Swarm Technologies," GAO-23-106930, Washington, D.C., 2023. [Online]. Available: https://www.gao.gov/products/gao-23-106930

[5] Y. Ding, Z. Yang, Q.-V. Pham, Z. Zhang, and M. Shikh-Bahaei, "Distributed Machine Learning for UAV Swarms: Computing, Sensing, and Semantics," arXiv:2301.00912, Jan. 2023. [Online]. Available: https://arxiv.org/abs/2301.00912

[6] K. Mershad, "PROACT: Parallel Multi-Miner Proof of Accumulated Trust Protocol for Internet of Drones," arXiv:2206.06670, Jun. 2022. [Online]. Available: https://arxiv.org/abs/2206.06670

[7] Y. Alqudsi and M. Makaraci, "UAV Swarms: Research, Challenges, and Future Directions," *Journal of Engineering and Applied Science*, 2025, doi: 10.1186/s44147-025-00582-3.

[8] W. Reed, "Distributed Algorithms for Autonomous Drone Swarms: Enabling Coordinated Surveillance," *Sensor Networks*, Aug. 2024. [Online]. Available: https://sensor-networks.org/distributed-algorithms-for-autonomous-drone-swarms-enabling-coordinated-surveillanc

[9] H. Chen, D. Huang, C. Wang, L. Ding, L. Song, and H. Liu, "Collision-Free Path Planning for Multiple Drones Based on Safe Reinforcement Learning," *Drones*, vol. 8, no. 9, p. 481, 2024, doi: 10.3390/drones8090481.

[10] W. Y. H. Adoni *et al.*, "Intelligent Swarm: Concept, Design and Validation of Self-Organized UAVs Based on Leader-Followers Paradigm for Autonomous Mission Planning," *Drones*, vol. 8, no. 10, p. 575, 2024, doi: 10.3390/drones8100575.