

Poster: Policy-Guided Enforcement in Unmanned Aerial Vehicles

Sonali Rout

*Khosla School of Information Technology
Indian Institute of Technology Delhi
anz228063@iitd.ac.in*

Vireshwar Kumar

*Khosla School of Information Technology
Indian Institute of Technology Delhi
viresh@cse.iitd.ac.in*

Abstract

Unmanned Aerial Vehicles (UAVs) operating in civilian airspace are increasingly vulnerable to remote software attacks that can manipulate flight paths and violate restricted no fly zones. Existing geofencing mechanisms that assume trustworthiness of the onboard software, fail under attacks such as command injection, memory corruption, or compromise of the complete software stack, including the operating system. We propose a novel framework consisting of a multi layered attestation and policy enforcement mechanism for the UAV equipped with a trusted execution environment. Our framework establishes trust in the navigation system by ensuring that a given flight plan is strictly followed and protected against unauthorized modifications, thereby guaranteeing mission integrity, albeit with acceptable computational overhead.

1 Introduction

Unmanned Aerial Vehicles (UAVs) or drones are increasingly used in both warfare and civilian applications, making them prime targets for attackers that aim to disrupt their intended path or crash them to sabotage missions [1]. Typically, a drone relies on an onboard software stack (e.g., PX4/Autopilot) and a communication protocol (e.g., MAVLink) to connect with the ground control station for mission execution and No-Fly Zone (NFZ) enforcement. The drone is loaded with a flight plan consisting of mission waypoints [2] and is expected to follow them throughout the mission. However, this assumption breaks down when attackers remotely compromise the onboard software stack or communication channels [3] through malicious commands, and tamper with waypoints [4].

We further note that existing systems provide limited mechanisms to verify at runtime whether the drone is following its intended mission and avoiding restricted airspace. However, this is an essential security requirement if UAVs are to be deployed for commercial purposes (e.g., home delivery of goods). Therefore, we propose a novel framework based on a trusted execution environment (TEE) to validate mission integrity and enable trusted policy enforcement.

2 System and Threat Model

The UAV system consists of a flight controller (e.g., Pixhawk running ArduPilot) and a board equipped with TEE (e.g., NVIDIA Jetson AGX Orin with ARM TrustZone and OP-TEE support). It is also equipped with a module that supports communication with the ground control station (e.g., over the MAVLink protocol). We consider a remote attacker that can access the drone exclusively over a wireless link with no physical access. The attacker is capable of software-based attacks, e.g., command injection, code injection, control-flow hijacking attacks such as Return-Oriented Programming (ROP) and Jump-Oriented Programming (JOP), memory corruption, firmware attacks, and kernel-level compromise. The goal of the attacker is to cause the drone to violate declared No-Fly Zone (NFZ) boundaries. GPS spoofing, jamming, physical attacks, and side-channel attacks are considered out of scope, as these represent signal-layer and physical-layer threats, and require complementary defenses.

3 Proposed Framework

We propose a five-layer attestation and policy enforcement framework where each layer executes in order and addresses an attack path that the layer above cannot cover as detailed in the Figure 1. The layers are sealed inside ARM TrustZone and OP-TEE, making them secure from the compromised normal world. We have listed the framework below.

- L1: Mission Integrity Layer:** Secure hashing and verification of mission waypoints and NFZ policies to protect mission plans and geofence rules against unauthorized waypoint modification.
- L2: Communication Integrity Layer:** Verification of MAVLink command authenticity and integrity to secure the GCS-UAV communication channel against malicious command injection.
- L3: Kernel Attestation Layer:** Runtime integrity verification of kernel memory to protect the Linux kernel and

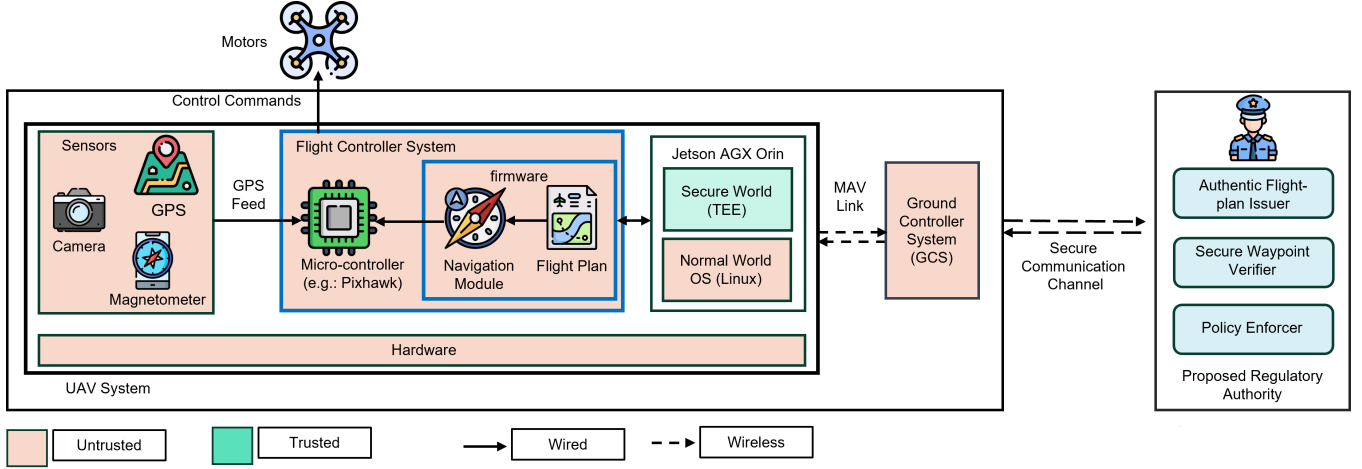


Figure 1: Overall structure and trust boundaries of the proposed framework which shows the clear isolation between trusted TEE layers and untrusted components, backed by the Regulatory Authority responsible for trusted policy enforcement.

privileged processes from kernel-level compromise.

L4: Firmware Attestation Layer: Validation of firmware and navigation stack to detect and prevent flight-controller firmware tampering.

L5: Runtime Enforcement Layer: Real-time policy enforcement against unsafe trajectories to protect navigation and control decisions from NFZ violation attempts.

Each layer within the framework is implemented as a Trusted Application (TA) running inside OP-TEE’s secure world. During boot, the secure world is initialized before the normal world, ensuring that all attestation is loaded into protected memory regions inaccessible to the normal world. The Mission Integrity Layer computes a SHA-256 hash of the uploaded flight plan and stores it in secure storage at mission initialization time. At runtime, the Runtime Enforcement Layer periodically re-hashes the active waypoint set and compares it against this stored reference. Any divergence triggers an immediate halt command issued directly to the Pixhawk over a dedicated secure UART channel that bypasses the compromised normal-world software stack.

The Communication Integrity Layer authenticates every inbound MAVLink message using HMAC-SHA256 with a key provisioned during a secure pairing procedure with the ground control station. Messages failing authentication are dropped and logged to tamper-evident secure storage for post-mission forensic analysis. The Kernel Attestation Layer leverages ARM TrustZone’s ability to trap SMC (Secure Monitor Call) instructions to periodically measure critical kernel memory regions, comparing live measurements against golden hashes established at boot. Similarly, the Firmware Attestation Layer validates the ArduPilot firmware image residing in the Pixhawk’s flash memory using a challenge-response protocol initiated from the secure world.

4 Conclusion

This work presents a design that provides layered protection spanning mission data, communication channels, kernel integrity, firmware integrity, and runtime navigation enforcement. Beyond improving cyber resilience, the architecture enables verifiable enforcement of safety constraints through hardware-backed trust anchors, meeting increasing regulatory requirements for policy compliance and enforcements.

Use of AI Policy

AI tools were used for text refinement and code assistance. All ideas, analyses, and conclusions are original to the authors.

References

- [1] B. Nassi, R. Bitton, R. Masuoka, A. Shabtai, and Y. Elovici, “SoK: Security and privacy in the age of commercial drones,” in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1434–1451.
- [2] Y. Yang, J. Khalife, J. J. Morales, and Z. M. Kassas, “UAV waypoint opportunistic navigation in GNSS-denied environments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 1, pp. 663–678, 2021.
- [3] S. Jeong, E. Park, K. U. Seo, J. D. Yoo, and H. K. Kim, “MUVIDS: False MAVLink injection attack detection in communication for unmanned vehicles,” in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2021, p. 25.
- [4] N. Yadav and V. Ganapathy, “Whole-program control-flow path attestation,” in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023, pp. 2680–2694.