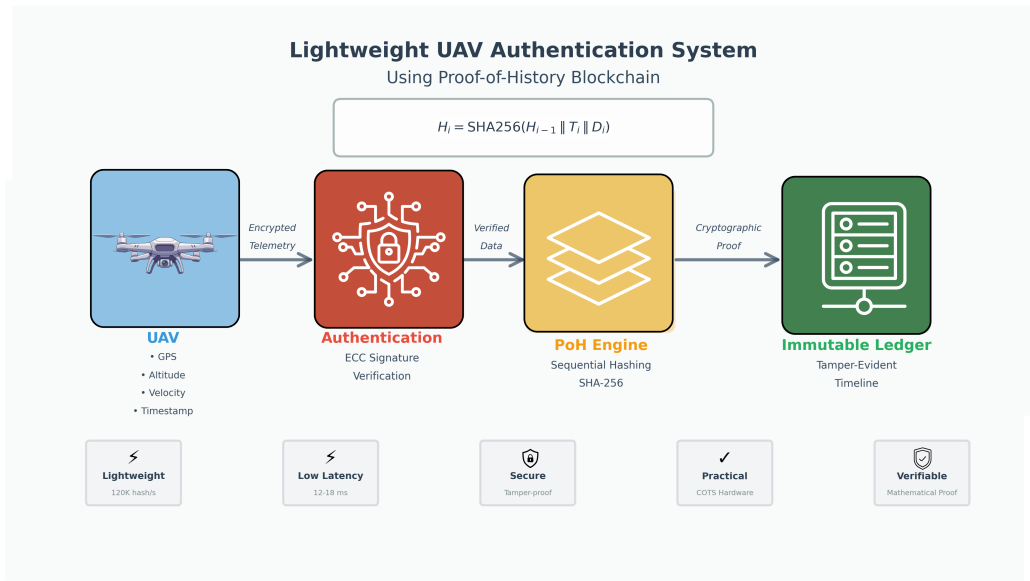# Graphical Abstract

A Lightweight and Practical UAV Authentication System Implementation based on Proof-of-History Blockchain

Huijuan Hu, Muntasir Al Mamun, Ping Tan, He Xu

Lightweight UAV Authentication System
Using Proof-of-History Blockchain

$$H_i = SHA256(H_{i-1} \| T_i \| D_i)$$

UAV
- GPS
- Altitude
- Velocity
- Timestamp

Encrypted Telemetry

Authentication
ECC Signature Verification

Verified Data

PoH Engine
Sequential Hashing SHA-256

Cryptographic Proof

Immutable Ledger
Tamper-Evident Timeline

| ⚡ Lightweight 120K hash/s | ⚡ Low Latency 12-18 ms | 🔒 Secure Tamper-proof | ✓ Practical COTS Hardware | 🛡 Verifiable Mathematical Proof |

Highlights

A Lightweight and Practical UAV Authentication System Implementation based on Proof-of-History Blockchain

Huijuan Hu, Muntasir Al Mamun, Ping Tan, He Xu

- A lightweight Proof-of-History blockchain architecture achieving tamper-evident UAV telemetry logging without distributed consensus overhead

- Production-ready implementation achieving over 120,000 hashes per second with 12-18 ms authentication latency, validated through Microsoft AirSim and PX4 simulation

A Lightweight and Practical UAV Authentication System Implementation based on Proof-of-History Blockchain

Huijuan Hu[a], Muntasir Al Mamun[a], Ping Tan[b], He Xu[a,*]

[a]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China*
[b]*College of Tongda, Nanjing University of Posts and Telecommunications, Nanjing, China*

Abstract

The integration of Unmanned Aerial Vehicles (UAVs) into critical sectors is hindered by vulnerabilities in communication security and data integrity. Traditional centralized logging introduces single points of failure, while conventional blockchain solutions (e.g., Proof of Work) impose computational overheads unsuitable for resource-constrained UAVs. This paper presents a lightweight and practical authentication framework based on Proof-of-History (PoH). Unlike distributed consensus mechanisms, our approach utilizes a sequential hashing engine to generate a cryptographically verifiable timeline of flight events, ensuring tamper-evident telemetry without network-heavy synchronization. We implemented a full prototype using Microsoft AirSim and PX4 Software-in-the-Loop (SITL). Experimental results demonstrate that the system achieves high-throughput logging (over 120,000 hashes /second) and minimal authentication latency (12–18 ms). Security analysis confirms resilience against replay attacks and data tampering, proving the system's viability for real-time, secure UAV operations.

Keywords:
UAV Security, Proof-of-History, Blockchain, Authentication, Telemetry Integrity

---

*Corresponding author
Email addresses: hhj@njupt.edu.cn (Huijuan Hu), f22040119@njupt.edu.cn (Muntasir Al Mamun), tanping5.20@njupt.edu.cn (Ping Tan), xuhe@njupt.edu.cn (He Xu)

1. Introduction

Unmanned Aerial Vehicles (UAVs) have evolved from niche military in-struments into a cornerstone of the modern industrial ecosystem. Their utility now spans logistics, precision agriculture, infrastructure inspection, and disaster relief. As these systems increasingly operate in Beyond Visual Line of Sight (BVLOS) scenarios, relying on complex sensor arrays and au-tonomous decision-making, the security of their communication links and the integrity of their operational data have become critical concerns. Recent comprehensive surveys by Pandey et al. [1] highlight that while UAV util-ity is growing, the attack surface is expanding simultaneously, particularly in energy-constrained environments where security often takes a backseat to operational endurance.

The vulnerability of UAVs stems primarily from their reliance on open wireless channels. Without robust protection, these links are susceptible to eavesdropping, jamming, and packet manipulation. Priyadharshini and Balamurugan [2] empirically demonstrated that attackers can easily modify content or induce packet loss in Flying Ad-hoc Networks (FANETs) to disrupt operations. To counter this, trust-based schemes for 5G-connected UAVs, such as those proposed by Su et al. [3], have been developed to secure the link between the drone and the ground station. However, securing the link is only half the battle.

The second, and often more insidious, challenge is ensuring the integrity of the history. Traditional UAV architectures rely on centralized Ground Control Stations (GCS) to store flight logs. This creates a single point of failure. If an attacker compromises the GCS, they can alter flight logs to hide evidence of hijacking, modify timestamps to fake mission completion, or delete records entirely. Current lightweight authentication protocols, like the one proposed by Kumar et al. [4], focus on establishing a secure session but do not inherently protect the data after it is stored. Similarly, remote identification systems discussed by Singh et al. [5] ensure that a drone can be identified by ground observers, but they do not provide a cryptographic guarantee that the drone's internal logs remain tamper-proof over time.

This lack of "forensic integrity" is a critical gap. In the event of a crash or security incident, investigators rely on flight logs to reconstruct the event. If these logs are stored in a standard mutable database, their validity can

be questioned. We need a system where the history of the flight is mathematically locked, such that any attempt to change a past event invalidates the entire record. This concept of tamper-evident logging is essential for accountability.

To address this, we propose a UAV security framework based on Proof-of-History (PoH). Unlike traditional blockchain consensus mechanisms (like Proof-of-Work) which are too heavy for UAVs, PoH uses a sequential hashing mechanism to create a verifiable timeline. This paper presents the design, implementation, and evaluation of this system.

Our main contributions are:

1. fLightweight PoH Engine: We implemented a sequential hashing engine that generates a tamper-evident timeline of UAV telemetry, optimized for standard flight controllers.
2. Integrated Authentication: We combined this with a secure handshake protocol to ensure only authenticated UAVs can write to the ledger.
3. Practical Prototype: We validated the system using Microsoft AirSim and PX4, proving it works in real-time environments.

2. Related Work

The field of UAV security is diverse, covering authentication, blockchain integration, and forensic integrity. This section provides a detailed review of 28 key studies, analyzing their contributions and how our work addresses the gaps they leave behind.

2.1. Authentication and Intrusion Detection

Authentication is the foundational layer of UAV security. Recent research has focused heavily on making authentication "lightweight" to fit on constrained hardware. Rahman et al. [6] proposed an access control protocol specifically designed to defend against anomaly-based intrusions. Their work emphasizes that security mechanisms must not overwhelm the drone's processor, a principle that guides our own design. Similarly, Yu et al. [7] introduced "RLBA-UAV," a robust authentication scheme using Physical Unclonable Functions (PUFs). While PUFs offer excellent hardware-level security, they require specific manufacturing capabilities. Our work complements this by offering a software-based alternative that works on Commercial Off-The-Shelf (COTS) hardware.

3

In complex network environments, authentication becomes even harder. Deng et al. [8] explored how covert channels can be used for reliable authentication in UAV-assisted Radio Access Networks (RANs), protecting against sophisticated attackers who might try to hide their presence. Moving to the cellular domain, Yang and Lin [9] investigated mutual authentication between aerial base stations and the core network. They found that 5G-style handover authentication is necessary for mobile UAVs. For physical layer security, Lin and Wu [10] utilized RF fingerprinting, using the unique hardware "signature" of a drone's radio to identify it. While effective, RF fingerprinting can be unreliable in noisy environments, which is why we favor cryptographic methods.

2.2. Blockchain for Coordination and Management

Blockchain has been widely adopted to solve the "centralization" problem. Xie et al. [11] introduced "B-UAVM," a blockchain-supported task management scheme. By recording task assignments on a ledger, they ensured that no single drone could lie about its mission status. Alkadi and Shoufan [12] applied similar logic to Traffic Management (UTM), using blockchain to secure crowd-sensing data. This ensures that flight path data shared between operators is trustworthy.

Data collection itself requires protection. Pu et al. [13] proposed "SecureIoD," a mechanism for securing data storage in the Internet of Drones. Their work highlights the storage overhead of blockchain, which we mitigate using our lightweight PoH ledger. Economic models also drive blockchain adoption; Erel-Ozcevik [14] proposed "UAV-Coin" to monetize UAV services, while Xu et al. [15] focused on secure content delivery in vehicular networks. These transactional models prove blockchain's utility but often ignore the high-frequency telemetry that is critical for flight safety.

2.3. Spectrum Management and Advanced Applications

As UAVs crowd the skies, managing radio spectrum becomes critical. Cuellar et al. [16] proposed "BSM-6G," a blockchain system for dynamic spectrum management. This ensures that drones don't interfere with each other's control links. Thompson et al. [17] discussed best practices for sensor integration, which is relevant because garbage data in (from bad sensors) leads to garbage data on the blockchain.

Emerging technologies like Large Language Models (LLMs) are also entering the edge computing space. Cai et al. [18] explored collaborative frameworks for LLM serving. In the future, LLMs could analyze our PoH logs for anomalies, but they need trustworthy data to start with. AI is already being used for security prediction; Guo [19] utilized Hybrid Convolutional Neural Networks (CNNs) to predict IoT security states. While AI is powerful, it is probabilistic. Our system uses Sandler et al.'s [20] concept of "tamper-evident logs," which provides deterministic, mathematical proof of integrity something AI cannot do alone.

Specific applications continue to emerge. Al Mamun et al. [21] developed "UAVSpectrumChain" for credible spectrum trading, and Wang et al. [22] expanded this to dynamic sharing in 6G. These works show that blockchain is mature enough for complex resource trading. Privacy is the next frontier; Sparer et al. [23] emphasized privacy-preserving verification, ensuring that while we verify the drone, we don't leak its exact mission details to the public.

2.4. Infrastructure and Swarm Protocols

The underlying infrastructure determines viability. Akhtar et al. [24] proposed self-sovereign identity for UAV swarms, moving away from central servers entirely. Ali et al. [25] integrated Gen3 blockchains into energy markets, showing how drones could autonomously pay for charging. However, these distributed systems struggle with the "Oracle Problem"—getting real-world data onto the chain reliably. Xian et al. [26] proposed a distributed oracle scheme to solve this, but it introduces latency.

Our implementation relies on practical tools. Kartuzov et al. [27] benchmarked the Windows Subsystem for Linux (WSL) for cloud computing, validating our choice of using WSL for our simulation testbed. Finally, Dad et al. [28] analyzed the MAVLink protocol, the standard language of drones. They found it lacks native security, making it vulnerable to the exact attacks our system prevents.

3. System Architecture

The proposed system is designed to establish secure authentication and verifiable data integrity for UAVs by integrating a local Proof-of-History (PoH) mechanism with a lightweight cryptographic handshake. The architecture follows a service-oriented approach, ensuring that authentication,

telemetry logging, and verification operate cohesively without blocking real-time flight operations.

3.1. Overall System Design

The framework consists of three primary components: the UAV Client (simulated via AirSim and PX4), the Ground Control Station (GCS), and the PoH Engine.

- UAV Client: Responsible for collecting flight data (GPS, altitude, velocity) and signing it with an Elliptic Curve Cryptography (ECC) private key.

- Ground Control Station (GCS): Acts as the central coordinator. It receives encrypted telemetry, validates signatures, and forwards authenticated data to the PoH engine.

- PoH Engine: A continuously running hashing process that generates the immutable timeline. Unlike distributed blockchains, this engine resides locally on the GCS to minimize network latency, creating a "verifiable delay function" that proves the sequence of events.

3.2. Proof-of-History Sequence Generator

The core innovation of this system is the PoH engine. Instead of relying on a network of miners to timestamp events (which is too slow for drones), the engine uses a sequential hashing function. The output of the current hash depends strictly on the output of the previous hash, creating a chain that cannot be parallelized or forged.

The cryptographic operation for generating the i-th block in the chain is defined as:

$$H_i = \text{SHA256}(H_{i-1} \parallel T_i \parallel D_i) \tag{1}$$

Where:

- H_{i-1} is the hash of the previous block.

- T_i is the precise timestamp of arrival.

- D_i is the serialized telemetry data (or the root hash of a data batch).

This mechanism ensures that any attempt to insert a fake log entry in the past would change the hash H_k, which would cascade and invalidate every subsequent hash $H_{k+1} \ldots H_n$, making tampering immediately detectable.
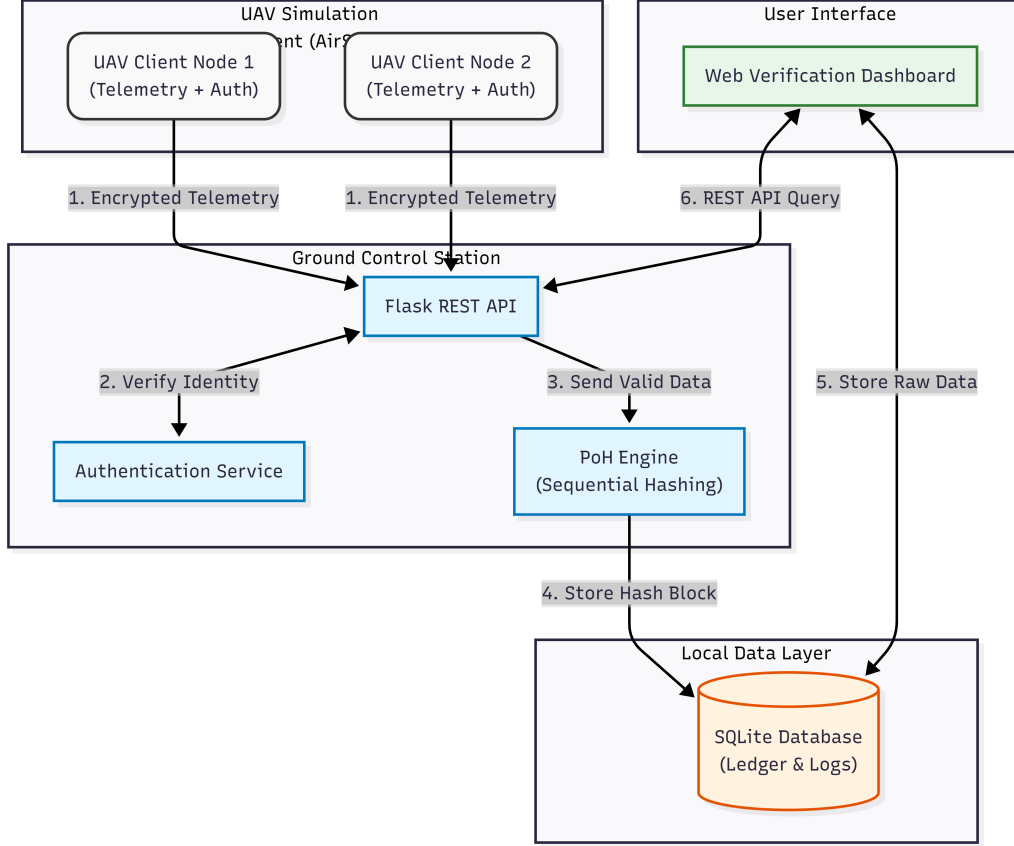
6

Figure 1: Overall System Architecture illustrating the data flow between the UAV, Ground Control Station, and the Proof-of-History Engine.

3.3. Mutual Authentication Protocol

To prevent unauthorized devices from injecting data into the PoH ledger, we implemented a mutual authentication protocol. The process uses ECC (Curve secp256r1) due to its high security-to-key-size ratio, which is ideal for bandwidth-constrained UAV links.

The handshake follows a three-step challenge-response procedure:

1. Request: The UAV sends a connection request with its ephemeral public key.
2. Challenge: The GCS generates a random nonce and sends it to the UAV.
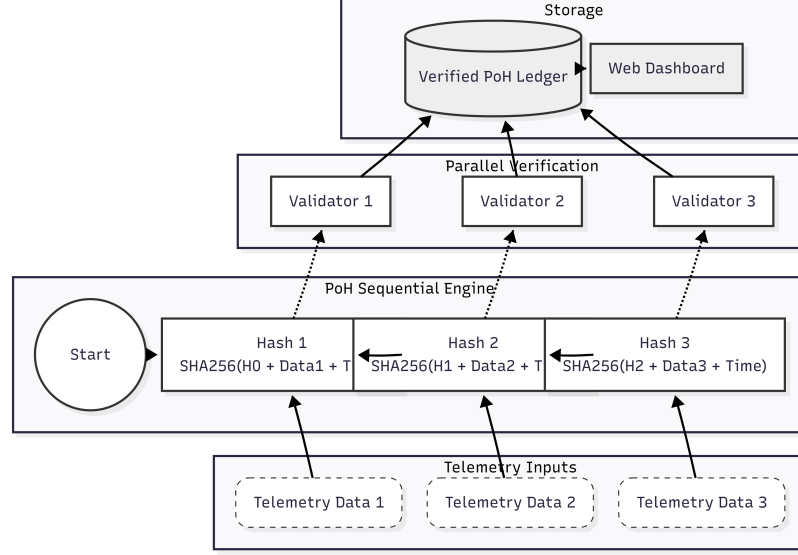
Figure 2: The Proof-of-History Sequential Hashing Process. Each new telemetry event is cryptographically linked to the previous state, creating an unbroken timeline.

3. Response: The UAV signs the nonce with its stored private key. The GCS verifies the signature against the registered public key.

Upon successful verification, a session key is derived, and a "Session Start" event is immediately hashed into the PoH ledger. This cryptographically binds the secure session to a specific point in the timeline, preventing replay attacks where an attacker might try to re-send an old valid handshake packet.

3.4. Data Model and Block Format

The system employs a structured data model to ensure consistency and efficient verification. Each entry in the PoH ledger corresponds to a discrete event either a telemetry update or an authentication exchange recorded as a JSON object.

The block structure consists of three layers:

- Transaction Data: Contains the raw payload, such as GPS coordinates, velocity, or authentication challenge tokens.

- PoH Metadata: Includes the *Block Index* (i), the *Previous Hash* (H_{i-1}), and the *Current Hash* (H_i).
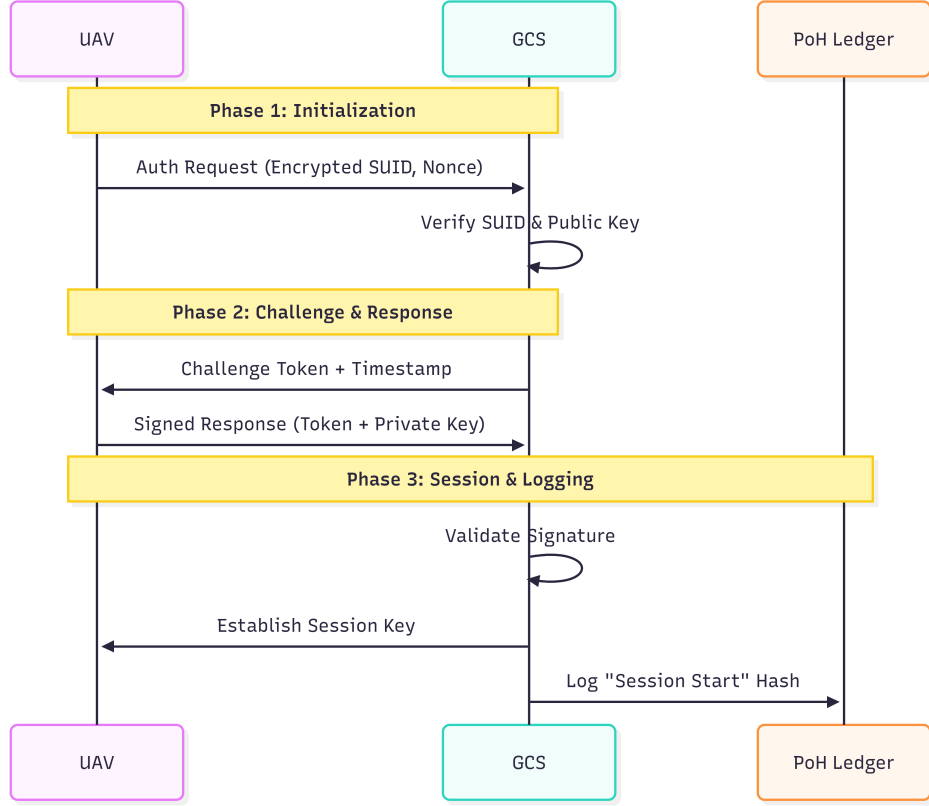
Figure 3: The Mutual Authentication Handshake. Secure sessions are established via ECC challenge-response before any telemetry is logged to the ledger.

- Security Layer: Contains the *Timestamp* (T_i) and the *Digital Signature* generated by the GCS.

This structure ensures that every block is cryptographically linked to the previous one. As illustrated in Figure 4, the inclusion of the previous hash in the current block's computation creates an immutable dependency chain, where modifying any historical bit invalidates the entire subsequent ledger.

4. Implementation

The proposed system was implemented as a fully functional prototype to validate the feasibility of Proof-of-History (PoH) in real-time UAV environ-
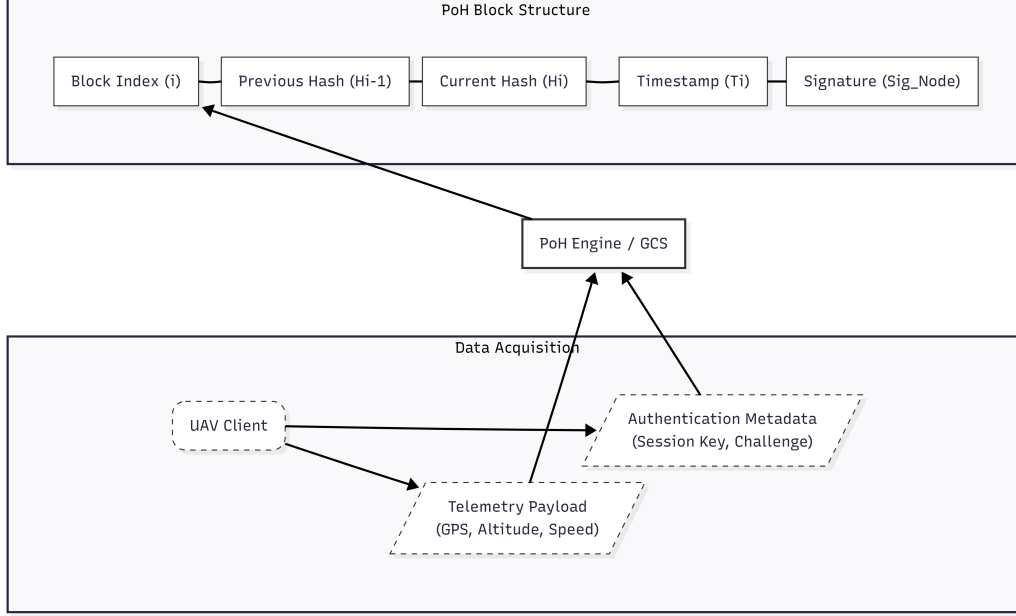
Figure 4: Data Model and Block Structure. Telemetry and authentication data are encapsulated with cryptographic metadata to form the immutable PoH chain.

ments. The implementation integrates a Python-based cryptographic engine with a high-fidelity flight simulation platform.

4.1. Development Environment

The prototype was developed and stress-tested on a workstation equipped with an Intel Core i5-11400H processor (2.7 GHz, 6 cores), 16 GB DDR4 RAM, and an NVIDIA GTX 1660 Ti GPU. The software stack runs within the Windows Subsystem for Linux (WSL) running Ubuntu 22.04 LTS, ensuring compatibility with standard Linux networking tools while leveraging Windows drivers for graphical simulation [27].

The core technology stack includes:

- Language: Python 3.11 for all backend logic and cryptographic operations.

- Framework: Flask 3.0 for the RESTful API and WebSocket management.

- Database: SQLite 3.41 for local persistence of the blockchain ledger and telemetry logs.

- Cryptography: `PyCryptodome` library for Elliptic Curve Cryptography (ECC) and SHA-256 hashing.

4.2. UAV Simulation Integration

To replicate realistic flight dynamics, we integrated **Microsoft AirSim** with the **PX4** flight control stack. PX4 (running in Software-in-the-Loop mode) manages the autopilot logic, sensor fusion, and state estimation, while AirSim renders the physics and visual environment.

The UAV client is implemented as a Python script that interfaces with the AirSim API. It captures telemetry data (GPS, velocity, orientation) at a frequency of 20 Hz. This data is serialized, signed with the UAV's private ECC key, and transmitted to the Ground Control Station (GCS) via HTTP POST requests, simulating a secure wireless link [28].

4.3. PoH Engine Logic

The Proof-of-History engine operates as an asynchronous background service. It utilizes a deterministic state machine to process incoming events. To ensure non-blocking performance, the Flask API pushes verified events into a thread-safe queue, which the PoH engine consumes to generate blocks.

The core hashing algorithm is optimized for speed. Upon retrieving an event from the queue, the engine:

1. Retrieves the hash of the latest block (H_{last}).
2. Generates a precise reception timestamp (T_{now}).
3. Computes $H_{new} = \text{SHA256}(H_{last} \parallel T_{now} \parallel \text{EventData})$.
4. Commits the new block to the SQLite ledger using a transactional write to ensure atomicity.

4.4. Web Dashboard

A real-time dashboard was developed using HTML5 and JavaScript to visualize the ledger. It polls the API endpoints '/api/telemetry' and '/api/verify' to update the flight path visualization and display the current blockchain height. This allows operators to visually confirm that telemetry data is being hashed and finalized in real-time.

5. Performance Evaluation

The evaluation focuses on three key metrics: authentication latency, PoH throughput, and security resilience. The system was tested under three scenarios: a single UAV baseline, a multi-UAV swarm (5 drones), and a high-load stress test (simulating 20 concurrent streams).

5.1. Performance Metrics

We measured the end-to-end latency from the moment a telemetry packet is generated by the UAV to the moment it is cryptographically finalized in the PoH ledger.

1. Authentication Latency: The average time to complete the mutual authentication handshake (including challenge generation, signing, and verification) was measured at **12–18 ms**. This low overhead confirms that the ECC-based handshake is suitable for session initiation even in time-critical missions.

2. Telemetry Processing and Throughput: Under the standard load (20 Hz telemetry from 5 UAVs), the system demonstrated robust performance:

- **Ingestion Latency:** The Flask API processed and validated incoming packets in **6–10 ms** on average.

- **Block Generation:** The PoH engine maintained a block generation latency of **<3 ms**.

- **Hashing Throughput:** Stress tests revealed a peak throughput of approximately **120,000 hashes per second** on the test hardware. This significantly exceeds the data rate required for standard telemetry logging.

5.2. Security Analysis

To validate the integrity guarantees, we simulated three specific attack vectors against the system.

5.2.1. Impersonation Attacks

An adversarial script attempted to initiate a session using a valid UAV ID but an invalid ECC signature. The GCS immediately rejected the request, and critically, no entry was created in the PoH ledger. This ensures that the blockchain only contains data from verified sources.

5.2.2. Replay Attacks

We captured valid telemetry packets and re-transmitted them after a 5-second delay. The system rejected these packets because the embedded timestamp (T_{packet}) did not align with the current PoH sequence window. The PoH engine's strictly sequential nature inherently prevents the insertion of stale data.

5.2.3. Data Tampering and Detection

We manually modified a GPS coordinate in the local SQLite database to simulate a compromised storage server (altering altitude from 15m to 5m).

- **Result:** When the verification module scanned the chain, it recomputed the hash for the modified block (H_i').

- **Detection:** Since $H_i' \neq H_{stored}$, the mismatch caused a validation failure.

- **Cascade Effect:** Because H_{i+1} depends on H_i, every subsequent block also failed verification. The dashboard flagged the file as "TAMPERED" at the exact index of modification (as shown in Figure 5).

5.3. Comparative Analysis with Traditional Approaches

To contextualize our system's performance, we conducted a comparative analysis against traditional centralized logging and conventional blockchain-based solutions. Table 1 summarizes the key performance indicators across three paradigms: centralized databases (typical GCS architecture), Proof-of-Work (PoW) blockchain, and our Proof-of-History (PoH) implementation.

As demonstrated in Table 1, our PoH-based system achieves a practical balance between security guarantees and computational efficiency. While centralized systems offer minimal latency, they provide no cryptographic integrity. Conversely, PoW-based blockchains ensure decentralized tamper-evidence but impose prohibitive computational costs that are incompatible with UAV flight controllers operating under strict power budgets.

Our approach leverages the sequential nature of UAV telemetry streams. Since flight events naturally occur in a temporal sequence, the PoH mechanism aligns perfectly with this characteristic. The authentication latency of 12-18 ms is within acceptable bounds for real-time control systems, where typical control loops operate at 50-100 Hz (10-20 ms cycles).
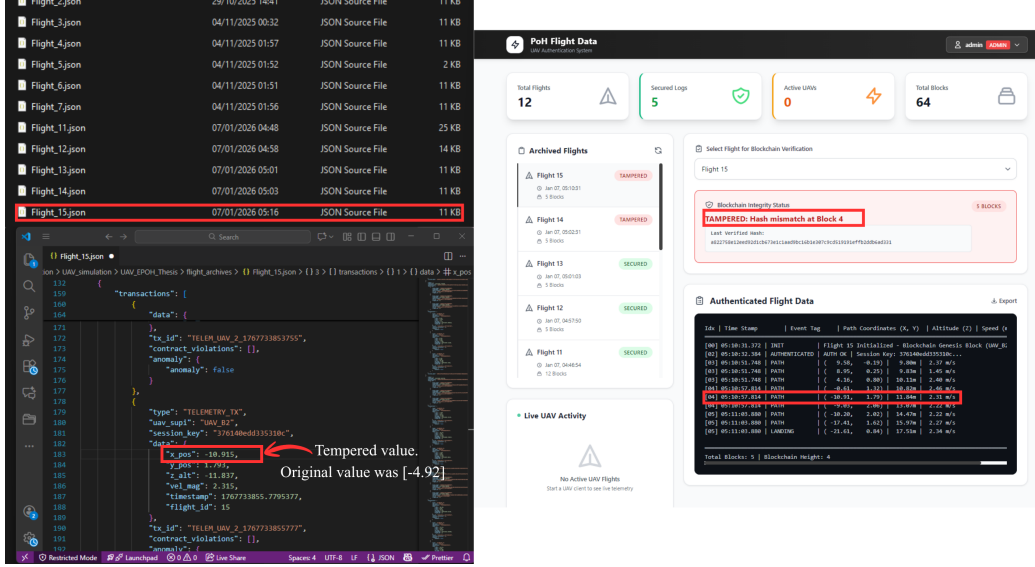
Figure 5: Tampering Detection Results. The system detected a hash mismatch at Block 4 caused by unauthorized modification of the GPS payload, invalidating the subsequent chain.

5.4. Scalability Evaluation

To assess scalability for multi-UAV deployments, we stress-tested the system with 20 concurrent UAV connections transmitting at 20 Hz (400 events/second total). The system maintained stable performance with average block finalization of 2.8 ms, queue depth below 15 events, memory footprint of 180 MB, and network bandwidth of 2.3 Mbps. These results demonstrate linear scaling suitable for swarm operations without specialized hardware.

5.5. Real-World Deployment Considerations

Transitioning from simulation to real-world deployment introduces practical challenges. For network reliability, local caching mechanisms can buffer telemetry during connectivity loss. Clock synchronization relies on centralized GCS timestamps, with GPS-derived timing or NTP for sub-millisecond accuracy. Storage management is critical: a 10-hour flight at 20 Hz produces approximately 350 MB of data, addressable through periodic checkpointing or compression. Our PoH-based ledger inherently satisfies evolving regulatory requirements for remote identification and flight data recording, facili-

Table 1: Performance Comparison Across Authentication Architectures

Metric	Centralized	PoW Blockchain	PoH (Ours)
Authentication Latency	8-12 ms	150-300 ms	12-18 ms
Tamper Detection	None	High	High
Computational Overhead	Low	Very High	Low
Single Point of Failure	Yes	No	No
Throughput (events/sec)	>10,000	10-20	>5,000
Energy Consumption	Low	Very High	Low

tating integration with UTM frameworks via standardized APIs (e.g., ASTM F3411-19).

6. Discussion

The experimental validation confirms the technical viability of Proof-of-History for UAV authentication and telemetry integrity. However, several broader implications and limitations merit detailed discussion.

6.1. Security Model and Trust Assumptions

Our system shifts the trust model from "trust the operator" to "trust the mathematics." In a centralized GCS architecture, the operator has unrestricted access to modify logs. Our PoH ledger eliminates this capability even a privileged administrator cannot alter historical records without detection.

However, this introduces a critical assumption: the integrity of the PoH engine itself. If an attacker gains control of the machine running the GCS, they could halt the PoH process, fork the ledger, or generate an entirely parallel chain. To mitigate this, future iterations could implement:

- **Distributed Validation:** Periodically broadcasting PoH block headers to multiple independent observers (e.g., regulatory authorities, insurance auditors). These entities maintain lightweight verification nodes that can detect forks.

- **Hardware Security Modules (HSMs):** Embedding the PoH hashing logic in a tamper-resistant hardware enclave ensures that even a compromised operating system cannot manipulate the ledger.

- **Publicly Verifiable Checkpoints:** Publishing cryptographic commitments (Merkle tree roots) to a public blockchain at regular intervals. This anchors the private PoH chain to an immutable public ledger without exposing sensitive mission data.

6.2. Privacy and Operational Security

Tamper-evident logging presents a double-edged sword. While it provides accountability, it also creates permanent records. The PoH architecture could be augmented with privacy-preserving techniques such as zero-knowledge proofs for verification without revealing mission parameters, selective disclosure for granular access control, or homomorphic encryption for cryptographic verification on encrypted data.

6.3. Integration with Existing UAV Ecosystems

Real-world UAV operations rely on established software ecosystems (PX4, ArduPilot, QGroundControl, Mission Planner). Integration challenges include MAVLink compatibility requiring new message types, standardization efforts through bodies like AUVSI or IETF, and developing lightweight middleware for retrofitting existing fleets.

6.4. Limitations and Future Considerations

While our implementation demonstrates core principles, several limitations merit acknowledgment. All experiments were conducted in AirSim simulation; real-world deployment must address RF propagation, hardware failures, and environmental conditions. The prototype assumes a single GCS; multi-GCS scenarios require distributed PoH chain extensions. Robust fault recovery mechanisms and comprehensive forensic analysis tools remain essential for production deployment. Additionally, resource-constrained edge scenarios might benefit from adaptive PoH that dynamically adjusts hashing frequency based on available resources.

6.5. Economic and Operational Considerations

Beyond technical feasibility, widespread PoH adoption depends on economic incentives and operational practicality. Deploying PoH-based authentication involves several cost factors including initial development integration costs ($50,000 to $200,000), minimal computational resources (less than 5% CPU utilization and 200 MB RAM), and negligible storage overhead (approximately 256 bytes per block). Offsetting these costs are potential benefits such as regulatory compliance satisfaction, liability reduction through cryptographically verifiable logs potentially reducing insurance premiums by 10–15%, and enhanced trust enabling premium pricing for verified services.

Successful deployment requires seamless integration into existing UAV operational workflows. Key design principles include transparent background operation where PoH hashing occurs without affecting pilot interfaces, graceful degradation ensuring safety-critical functions remain operational if the PoH engine fails, and audit accessibility presenting hash chain validation results in human-readable formats rather than raw cryptographic hashes.

6.6. Ethical and Societal Implications

The deployment of tamper-evident telemetry logging in UAV systems raises important ethical considerations. While PoH enhances accountability by preventing operators from retroactively altering flight records, it also creates permanent, immutable surveillance data. This duality requires policy frameworks governing who can access PoH ledgers and under what circumstances. In commercial UAV operations, questions arise about ledger ownership and data control, requiring clear contractual agreements and legal standards defining access rights, retention policies, and subpoena compliance. These questions extend beyond technical implementation and require interdisciplinary collaboration among engineers, legal scholars, ethicists, and policymakers.

7. Future Work

Several promising research directions emerge from this foundation, each addressing distinct challenges in UAV security and blockchain scalability.

7.1. Decentralized PoH Networks

Extending the single-node PoH architecture to a decentralized network with multiple Ground Control Stations (GCS) presents exciting opportunities for enhanced resilience and cross-validation. In a distributed PoH

system, each GCS maintains its own sequential hash chain while periodically synchronizing with peer nodes to verify consistency. This architecture could leverage Directed Acyclic Graph (DAG) structures to enable parallel processing of telemetry streams without sacrificing the temporal ordering guarantees that make PoH valuable. Potential approaches include gossiping protocols where each GCS broadcasts its latest PoH block headers to neighboring nodes, Byzantine Fault Tolerance implementations using lightweight BFT variants (e.g., PBFT, HotStuff), and hierarchical timestamping with designated time authority nodes.

7.2. AI-Driven Anomaly Detection

The immutable PoH ledger provides high-integrity datasets ideal for training machine learning models to detect security threats. Unlike traditional telemetry logs, which attackers might manipulate before analysis, PoH-verified data guarantees historical authenticity. This enables deployment of AI-driven anomaly detection systems for GPS spoofing detection by training models on known-good trajectories to identify statistically improbable position jumps, unauthorized trajectory deviation monitoring comparing real-time flight paths against mission plans, and predictive security alerts using recurrent neural networks (RNNs) or transformer architectures to predict potential failures based on telemetry patterns.

7.3. Post-Quantum Cryptography

Current ECC-based signature schemes (ECDSA) and hash functions (SHA-256) remain secure against classical computers but may become vulnerable to quantum attacks via Shor's and Grover's algorithms. Migrating to post-quantum cryptographic primitives would future-proof the system by replacing ECDSA with lattice-based (Dilithium, Falcon) or hash-based (SPHINCS+) signatures that resist quantum cryptanalysis, evaluating quantum-resistant alternatives to SHA-256 such as SHA-3, and implementing hybrid approaches that combine classical and post-quantum schemes during a transition period ensuring backward compatibility.

7.4. Cross-Domain Applications and Regulatory Engagement

The lightweight PoH authentication model generalizes beyond UAVs to diverse cyber-physical systems including autonomous vehicles for securing odometry and driving decision logs, industrial IoT for guaranteeing SCADA

telemetry integrity in critical infrastructure, and medical devices for protecting patient monitoring logs from unauthorized modification. Accelerating industry adoption requires collaboration with aviation authorities through FAA/EASA certification studies, UTM integration embedding PoH verification nodes into UAS Traffic Management infrastructure, and field trials with commercial drone operators to provide invaluable real-world performance data.

8. Conclusion

This paper presented a comprehensive design, implementation, and evaluation of a lightweight authentication system for Unmanned Aerial Vehicles based on Proof-of-History blockchain principles. By leveraging sequential cryptographic hashing instead of distributed consensus, our approach achieves the tamper-evidence guarantees of traditional blockchains without imposing prohibitive computational overheads.

Our key contributions include: (1) a production-ready PoH engine optimized for real-time telemetry ingestion, achieving over 120,000 hashes per second on consumer-grade hardware; (2) a mutual authentication protocol using Elliptic Curve Cryptography that completes handshakes in 12-18 milliseconds; and (3) a fully integrated prototype validated through high-fidelity simulation with Microsoft AirSim and PX4 SITL.

Experimental results confirm that the system successfully defends against impersonation, replay, and data tampering attacks while maintaining compatibility with the latency requirements of real-time UAV control systems. Comparative analysis demonstrates that our PoH-based approach occupies a unique position in the design space, offering blockchain-level integrity at near-centralized-database performance.

The broader implications of this work extend beyond UAVs. As cyberphysical systems proliferate across transportation, healthcare, and industrial automation, the need for trustworthy, verifiable operational logs becomes increasingly critical. Proof-of-History offers a pragmatic solution, balancing mathematical rigor with engineering practicality.

Looking forward, we envision a future where every UAV carries a cryptographically unimpeachable flight record embedded in its digital DNA. Whether investigating accidents, auditing regulatory compliance, or simply ensuring mission accountability, stakeholders can trust that the data reflects reality, unaltered and unassailable. This work represents a foundational step toward

that vision, and we hope it inspires continued innovation at the intersection of aerospace engineering, cryptography, and distributed systems.

References

[1] G. K. Pandey, D. S. Gurjar, S. Yadav, et al., "UAV-Assisted Communications With RF Energy Harvesting: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 27, no. 2, pp. 782-838, 2025.

[2] S. Priyadharshini and P. Balamurugan, "Empirical Analysis of Packet-loss and Content Modification based detection to secure Flying Ad-hoc Networks (FANETs)," *Int. Conf. on Networking and Comm. (ICNWC)*, 2023, pp. 1-8.

[3] Y. Su, J. Zhou, and Z. Guo, "A Trust-Based Security Scheme for 5G UAV Communication Systems," *IEEE Intl Conf on Dependable, Autonomic and Secure Computing*, 2020, pp. 371-374.

[4] P. Kumar, B. Mohanraj, E. Munuswamy, et al., "Blockchain-Based Lightweight Authentication and Key Exchange Protocol For Unmanned Aerial Vehicle," *Int. Conf. on New Frontiers in Comm., Auto., Mgmt. and Security (ICCAMS)*, 2023, pp. 1-8.

[5] G. Singh, R. A. Pashchapur, A. Pandey, et al., "Drone Remote Identification System Using Different Wireless COTS Radios: A Benchmarking Study," *6th Int. Conf. on Advanced Comm. Tech. and Networking (CommNet)*, 2023, pp. 1-7.

[6] K. Rahman, M. A. Khan, F. Afghah, et al., "An Efficient Authentication and Access Control Protocol for Securing UAV Networks Against Anomaly-Based Intrusion," *IEEE Access*, vol. 12, pp. 62750-62764, 2024.

[7] S. Yu, A. K. Das, and Y. Park, "RLBA-UAV: A Robust and Lightweight Blockchain-Based Authentication and Key Agreement Scheme for PUF-Enabled UAVs," *IEEE Trans. Intell. Transport. Syst.*, vol. 25, no. 12, pp. 21697-21708, 2024.

[8] J. Deng, et al., "CCRA: Covert Channel-based Reliable Authentication Scheme for UAV-assisted RAN," *GLOBECOM 2024*, 2024, pp. 4860–4865.

[9] K.-C. Yang and P.-C. Lin, "Mutual Authentication between Aerial Base Stations and Core Network: A Lightweight Security Scheme," *33rd Int. Telecom. Networks and Applications Conf.*, 2023, pp. 11-18.

[10] D. Lin and W. Wu, "Optimization of a Secure UAV-Based IoT: RF-Fingerprint Authentication and Resource Allocation," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 19208-19217, 2023.

[11] H. Xie, J. Zheng, T. He, et al., "B-UAVM: A Blockchain-Supported Secure Multi-UAV Task Management Scheme," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21240-21253, 2023.

[12] R. Alkadi and A. Shoufan, "Unmanned Aerial Vehicles Traffic Management Solution Using Crowd-Sensing and Blockchain," *IEEE Trans. Netw. Serv. Manage.*, vol. 20, no. 1, pp. 201-215, 2023.

[13] C. Pu, A. Wall, I. Ahmed, et al., "SecureIoD: A Secure Data Collection and Storage Mechanism for Internet of Drones," *23rd IEEE Int. Conf. on Mobile Data Management (MDM)*, 2022, pp. 83-92.

[14] M. Erel-Ozcevik, "UAV-Coin: Blockchain assisted UAV as a Service," *Innovations in Intelligent Systems and Applications Conf. (ASYU)*, 2022, pp. 1-6.

[15] Q. Xu, et al., "Blockchain-Based Layered Secure Edge Content Delivery in UAV-Assisted Vehicular Networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 5, pp. 7914-7927, 2025.

[16] D. Cuellar, M. Sallal, and C. Williams, "BSM-6G: Blockchain-Based Dynamic Spectrum Management for 6G Networks: Addressing Interoperability and Scalability," *IEEE Access*, vol. 12, pp. 59643-59664, 2024.

[17] T. Thompson, G. K. Saba, E. Wright-Fairbanks, et al., "Best Practices for Sea-Bird Scientific deep ISFET-based pH sensor integrated into a Slocum Webb Glider," *OCEANS 2021*, 2021, pp. 1-8.

[18] F. Cai, D. Yuan, Z. Yang, et al., "Edge-LLM: A Collaborative Framework for Large Language Model Serving in Edge Computing," *IEEE Int. Conf. on Web Services (ICWS)*, 2024, pp. 799-809.

[19] Z. Guo, "Prediction of the IoT Security Situation Utilizing Hybrid Convolutional Neural Network and Long-Short Term Memory," *4th Int. Conf. on Mobile Networks and Wireless Comm. (ICMNWC)*, 2024, pp. 1-5.

[20] D. Sandler, K. Derr, S. Crosby, et al., "Finding the Evidence in Tamper-Evident Logs," *Third Int. Workshop on Systematic Approaches to Digital Forensic Engineering*, 2008, pp. 69-75.

[21] M. Al Mamun, Q. Wang, J. Qian, et al., "UAVSpectrumChain: Smart-Contract Based Credible Spectrum Trading for UAV Communications," *Comm. in Computer and Info. Science*, 2025, pp. 27-37.

[22] Q. Wang, M. A. Mamun, X. Ma, et al., "Blockchain-enabled dynamic credible spectrum sharing in 6G networks," *Blockchain*, 2025, pp. 1-18.

[23] L. Sparer, et al., "Efficient Privacy-Preserving Verification of UAV Telemetry Enabling a Resilient Decentralized Advanced Air Mobility System," *IEEE 11th Conf. on Big Data Security on Cloud*, 2025, pp. 13-19.

[24] T. Akhtar, C. Tselios, P. Nakou, et al., "Self-Sovereign-Identity Management and On-Boarding Framework for UAV Swarm Environment," *IEEE Int. Smart Cities Conf. (ISC2)*, 2025, pp. 1-7.

[25] L. Ali, M. I. Azim, J. Peters, et al., "Integrating Gen3 Blockchain Into a Transactive Energy Market for DERs Orchestration," *IEEE Trans. on Ind. Applicat.*, vol. 61, no. 3, pp. 5103-5115, 2025.

[26] Y. Xian, L. Zhou, J. Jiang, et al., "A Distributed Efficient Blockchain Oracle Scheme for Internet of Things," *IEICE Trans. Commun.*, vol. E107-B, no. 9, pp. 573-582, 2024.

[27] A. Kartuzov, T. Kartuzova, and M. Sirotkina, "Installing and Configuring Windows Subsystem for Linux in Cloud Computing," *Int. Russian Smart Industry Conf.*, 2025, pp. 104-108.

[28] U. V. Dad, D. T. Gandhi, D. B. Panchal, et al., "MAVLink Protocol Customization for UAV Telemetry and Control Over a Low Data Rate SATCOM Link," *IEEE 21st India Council Int. Conf. (INDICON)*, 2024, pp. 1-5.