Quantum Safe Drone Security - Authenticated Key Exchange and Privacy Preserving Intrusion Detection for Low Altitude Intelligent Transportation Systems

Quantum Safe Drone Security - Authenticated Key Exchange and Privacy Preserving Intrusion Detection for Low Altitude Intelligent Transportation Systems*

Shafiq Ahmed[a,*], Mohammad Hossein Anisi[b], Ayesha Iqbal[c], Abhirami Suresh[d], Mohammed Amoon[e] and Kadambri Agarwal[f]

[a]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
[b]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
[c]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
[d]School of Physics, Engineering and Computer Science, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, United Kingdom
[e]Department of Computer Science and Engineering, Applied Studies College, King Saud University, P.O. Box 28095, Riyadh, 11437, Saudi Arabia
[f]Department of Computer Science and Engineering, ABES Engineering College, , Ghaziabad, 201009, India

ARTICLE INFO

Keywords:
Post Quantum Cryptography
Low Altitude Intelligent Transportation System
ML-KEM
ML-DSA
Authenticated Key Exchange
Federated Learning
Differential Privacy
AAV Security

ABSTRACT

Low altitude intelligent transportation systems (LITS) coordinate Autonomous Aerial Vehicles (AAVs) across shared airspace, but their wireless drone to ground links remain exposed to Global Positioning System (GPS) spoofing, radio frequency (RF) jamming, command injection, and future quantum attacks. Existing Internet of Drones authentication schemes usually rely on elliptic curve or symmetric primitives, while National Institute of Standards and Technology (NIST) transition guidance targets deprecation of quantum vulnerable public key mechanisms beginning after 2030, they also rarely connect authentication with privacy aware intrusion detection. We propose Post Quantum Low Altitude Intelligent Transportation Security (PQ-LITS), a multi layer framework that combines post quantum mutual authentication with federated telemetry anomaly detection. The authentication layer uses the Module Lattice Based Key Encapsulation Mechanism (ML-KEM-768, Federal Information Processing Standards (FIPS) 203) and the Module Lattice Based Digital Signature Algorithm (ML-DSA-65, FIPS 204) to establish a session key in three messages, with 0.82 milliseconds (ms) latency over 1,000 runs. The detection layer trains autoencoders across three ground control stations with non independent and identically distributed (non IID) attacks using Federated Averaging (FedAvg) and Gaussian differential privacy ($\sigma = 0.01$). On 15,000 synthetic telemetry records, PQ-LITS reaches F1 = 0.7549, 95.7% of the centralised upper bound, while reducing communication by 58.2%. We give a security proof in the Quantum Random Oracle Model, reducing session security to Module Learning With Errors (Module LWE) and Module Short Integer Solution (Module SIS), and release the dataset, code, and benchmarks.

1. Introduction

The low-altitude economy (LAE) is no longer a speculative concept. Over the past three years, it has matured into a tangible ecosystem where autonomous aerial vehicles(AAV) carry out package delivery, traffic surveillance, agricultural monitoring, and even passenger transport in the form of electric vertical takeoff and landing (eVTOL) air taxis [1, 2]. At the centre of this expansion sits the Low Altitude Intelligent Transportation System (LITS), a networked infrastructure that coordinates fleets of drones, ground control stations (GCSs), and federation servers within shared airspace below roughly 1 000 metres. Chinese provincial governments have already begun issuing commercial beyond visual line of sight (BVLOS) licences, and industry analysts project the global AAV market to surpass USD 54 billion by 2030 [3]. The pace is real, and so are the security risks that come with it.

Every AAV in a LITS deployment exchanges telemetry, command, and video data with its assigned GCS over wireless links that are, by their very nature, exposed to eavesdropping, replay, and injection attacks [4]. A spoofed GPS signal can redirect a delivery drone into restricted airspace. A jammed radio link can sever the control path mid flight. A forged command packet can override autopilot waypoints entirely. These are not hypothetical cases, the academic literature documents each of them in experimental settings [3], and military operators have dealt with GPS spoofing in contested environments for years. These examples point to a basic requirement: a LITS deployment must authenticate AAVs and GCSs before a single telemetry byte is transmitted.

Meeting this requirement with today's IoD protocols is difficult. Most authentication and key establishment (AKE) protocols proposed for the Internet of Drones (IoD) rely on classical assumptions such as hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP) for ECC based

*The authors extend their appreciation to Ongoing Research Funding Program (ORF-2026-968), King Saud University Riyadh, Saudi Arabia.
*Corresponding author: Shafiq Ahmed
s.ahmed@essex.ac.uk (S. Ahmed); m.anisi@essex.ac.uk (M.H. Anisi); ayeshaiqbal0197@gmail.com (A. Iqbal); abhiramisuresh16@gmail.com (A. Suresh); mamoon@ksu.edu.sa (M. Amoon); kadambri.agarwal@abes.ac.in (K. Agarwal)
ORCID(s): 0000-0001-9599-1887 (S. Ahmed); 0000-0001-8414-2708 (M.H. Anisi); 0009-0009-4774-8378 (A. Iqbal); 0009-0002-5460-2667 (A. Suresh); 0000-0002-3800-0933 (K. Agarwal)

schemes [5, 6], the intractability of integer factorisation for RSA or the hardness of the Ring Learning with Errors (R-LWE) problem for a handful of lattice based proposals [7, 8]. The trouble is that Shor's algorithm [9] can solve both ECDLP and integer factorisation in polynomial time on a sufficiently large quantum computer. NIST recognised this threat formally in 2024 when it published IR 8547 [10], which sets a deprecation date of 2030 for RSA, ECDSA, EdDSA, DH, and ECDH, followed by a complete ban after 2035. The window is short, especially for systems like LITS that are being designed and deployed right now. Harvest now decrypt later attacks compound the urgency, an adversary who records encrypted drone telemetry today can decrypt it the moment a cryptographically relevant quantum computer (CRQC) becomes available.

NIST finalised its first three post quantum cryptography (PQC) standards in August 2024, ML-KEM (FIPS 203) [11] for key encapsulation and ML-DSA (FIPS 204) [12] for digital signatures and both grounded in the Module Learning with Errors (MLWE) assumption along with SLH-DSA (FIPS 205) based on hash functions. These standards are production ready. The question is no longer whether post quantum cryptography works but how to integrate it into the particular constraints of low altitude airspace without crippling communication latency or overwhelming the companion computers that sit alongside Pixhawk flight controllers on commercial drones.

Beyond authentication, LITS must also detect malicious or abnormal telemetry after a secure channel has been established. Once the communication channel is secured, the GCS still needs to detect whether the telemetry flowing over it has been tampered with at the sensor level or whether the AAV is behaving anomalously due to a cyberattack. Traditional rule based intrusion detection struggles with the heterogeneous non stationary nature of drone telemetry. Machine learning offers a more adaptive path, but centralised training requires all raw telemetry to be pooled at a single server, which creates a single point of failure and a privacy liability. Federated learning (FL) sidesteps this by keeping raw data local to each GCS while still producing a shared model [13]. The catch is that FL itself introduces new attack surfaces such as a curious aggregation server can infer private training samples from uploaded gradients [14]. Differential privacy (DP) provides a formal mechanism to bound this leakage [15], but it has rarely been integrated with authenticated key exchange in the IoD context.

We introduce **PQ-LITS**, a two layer security architecture for low altitude intelligent transportation systems that addresses both the authentication gap and the anomaly detection gap simultaneously. Our contributions are threefold. *First*, we present a post quantum mutual authentication and key agreement protocol built on ML-KEM-768 (FIPS 203, NIST Level 3) and ML-DSA-65 (FIPS 204, NIST Level 3), with certificate based identity verification, dual key encapsulation for perfect forward secrecy, and a cryptographic zone handover mechanism. The protocol produces a shared session key in three messages with a measured latency of 0.82 ms on commodity hardware, which is 50.5% faster than the state of the art (SOTA) protocols [16] under the reported benchmark setup. *Second*, we design a federated autoencoder based anomaly detection system operating across geographically distributed GCS clients with non IID attack distributions using FedAvg aggregation with Gaussian differential privacy ($\sigma = 0.01$). The system detects GPS spoofing, RF jamming, and command injection attacks while reducing communication overhead by 58.2% compared to centralised training and achieving 95.7% of the centralised F1 score upper bound. *Third*, we provide a complete working implementation benchmarked on real hardware with the cryptographic protocol implemented using liboqs (FIPS 203/204 final) and measured over 1 000 iterations, and the federated pipeline built in PyTorch with synthetic yet realistic non IID telemetry across three GCS sectors.

The rest of this paper is organised as follows. Section 2 surveys related work. Section 3 introduces the mathematical foundations, system model, and threat model. Section 4 presents the PQ-LITS protocol across six phases. Section 5 provides informal and formal security analysis. Section 6 reports benchmarks alongside federated learning results. Section 7 concludes this manuscript.

2. Related Work

This section reviews prior efforts at the intersection of drone authentication and anomaly detection. Table 2 summarises the surveyed protocols along ten security and functionality features.

2.1. Symmetric and Hash Based AKE for IoD

Early IoD authentication protocols lean on lightweight symmetric primitives. Zhang et al. [17] propose a hash only key agreement mechanism but subsequent cryptanalysis by Chang et al. [18] reveals replay and impersonation vulnerabilities. Akram et al. [19] integrate AES with fuzzy extractors for three factor authentication in drone assisted vehicular networks, yet Park et al. [20] later demonstrate stolen verifier weaknesses. Hussain et al. [21], building on Wazid et al. [22], design a user authentication scheme for IoD enabled smart cities that still cannot resist stolen verifier and tampering attacks. Tanveer et al. [23] incorporate Chebyshev chaotic maps with AES and hash functions, achieving forward secrecy through the CDMP hard problem, though the scheme offers no defense against tampering and exposes the drone to cloning.

These symmetric solutions are fast (typically below 3 ms [16]), but none withstand quantum adversaries, and mostly lack mutual authentication.

2.2. Public Key and Lattice Based AKE for IoD

ECC based protocols form the backbone of public key IoD authentication. Nikooghadam et al. [6] rely on CDH based forward secrecy but later analysis [16] exposes secret leakage and stolen verifier weaknesses. Hussain et al. [5] add

three factor authentication on top of ECC without achieving mutual authentication or identity anonymity. Berini et al. [24] push toward smaller key sizes with Hyperelliptic Curve Cryptography, though a subtle binding flaw in their public key management breaks the intended cryptographic relationship [16].

Two recent protocols attempt lattice based quantum resistance. Mishra et al. [7] and Chaudhary and Lee [8] both use Ring LWE, but R-LWE has known weak instances for some parameter choices [25, 26] and was not selected by NIST for standardisation. Neither addresses mutual authentication between all participants or stolen verifier resilience.

The only published IoD protocol that uses a NIST standardised post quantum KEM is QSAKE by Shamshad et al. [16]. QSAKE employs Kyber-512 for encapsulation, a ring oscillator PUF with BCH error correction for drone fingerprinting, and AES-GCM for verifier table protection. It achieves 1.65 ms total computation cost and satisfies eight security features. Still, it operates at NIST Security Level 1, does not use digital signatures for certificate based identity binding, lacks telemetry anomaly detection, and does not support zone handover for multi sector operations.

2.3. PUF Based Authentication

PUFs offer hardware level defense against drone cloning and memory extraction. Yu et al. [27] achieve cloning resistance but fail to protect against stolen verifier attacks. Sharma et al. [28] leave verifier data unprotected. Irshad et al. [29] design SPAKE-DC for 5G drone communication but omit PUF error correction risking authentication failures from noisy hardware responses. Tanveer et al. [30] partially address this with AEGIS authenticated encryption though their scheme stores registration parameters unencrypted. Mahmood et al. [31] combine chaotic mapping with PUF obtaining forward secrecy via the CDMP hard problem but again struggle with PUF noise and stolen verifier protection.

PUF based approaches contribute device capture resistance that purely software based protocols cannot match. Our protocol achieves the same property through a different mechanism, XOR masking the private decapsulation key with a hardware bound secret η_i, so that an adversary who dumps drone memory obtains $\alpha_i = dk_i \oplus H(\eta_i \| UID_i \| \texttt{0x01})$ rather than dk_i itself.

2.4. Anomaly Detection and Federated Learning for AAVs

A separate body of work tackles intrusion detection on drone networks using machine learning. Ceviz et al. [3] survey attack taxonomies and IDS approaches for AAVs and FANETs noting that most systems assume centralised data collection which creates both a privacy liability and a single point of failure. The SP-IoUAV model by Ntizikira et al. [32] combines federated learning with differential privacy and secure multi party computation for IoD intrusion detection using a CNN-LSTM architecture. Their work comes closest to our privacy preserving anomaly detection layer but it does not address authenticated key exchange. The FL operates over an assumed secure channel without specifying how that channel is established. Ceviz et al. [33] propose a federated IDS for FANETs that demonstrates how FL narrows the performance gap with centralised detection consistent with our own finding that the federated model reaches 95.7% of the centralised F1 upper bound. The Mamba KAN Liquid hybrid model [34] achieves detection rates above 95% with only 47.3 ms inference latency confirming that lightweight models can run on companion computer class hardware. None of these detection focused works, however, consider how the model training process itself should be protected cryptographically, or how the AAV to GCS channel is authenticated before telemetry flows.

To make the novelty claim experimentally traceable, we use two comparison tracks. For the authentication layer, QSAKE [16] is the closest and strongest comparator because it is the only surveyed IoD protocol with a NIST standardised post quantum KEM; Section 6.2.3 gives the head-to-head computation comparison and reports 0.815 ms for PQ-LITS against 1.649 ms for QSAKE under the reported benchmark setup. For the intrusion detection layer, no public work appears to combine the same federated, non-IID, multi-GCS telemetry setting with post quantum AKE, so we compare against the most similar FL/IDS studies [32, 33, 34] and then report local-only, federated, and centralised baselines in Section 6.4. The review also includes directly relevant studies Chaudhary and Lee's dynamic identity quantum safe IoD protocol [8] and Sharma et al.'s PUF based IoD authentication scheme [28]. We use them for positioning because neither combines NIST-final ML-KEM/ML-DSA AKE with federated IDS. Three gaps emerge from the surveyed literature. No existing IoD authentication protocol uses the NIST finalised ML-KEM and ML-DSA at Security Level 3 with certificate based mutual authentication and dual key encapsulation. No prior work integrates post quantum key exchange with federated anomaly detection and differential privacy as a unified framework and none of the surveyed protocols support cryptographic zone handover for multi sector LITS deployments. Table 2 makes these gaps visible at a glance.

3. Preliminaries

Before presenting the protocol itself, we lay out the mathematical ground it stands on. This section walks through the lattice hard problems, the NIST post quantum primitives, and the federated learning setup that together form the two layers of PQ-LITS. It wraps up with the system model and the threat model that frame our security claims.

3.1. Notation

Table 1 collects the symbols and conventions used throughout this paper. Calligraphic letters ($\mathcal{TA}$, $\mathcal{GCS}_k$, $\mathcal{AAV}_i$) refer to system entities, while arrows $\Rightarrow$ and $\rightarrow$ distinguish secure channels from insecure ones.

Table 1
Summary of Notations

Notation	Description	Notation	Description
$\mathcal{TA}$	Trusted authority (offline registration)	$\mathcal{FS}$	Federation server for model aggregation
$\mathcal{GCS}_k$	Ground control station of sector k	$\mathcal{AAV}_i$	autonomous aerial vehicle i
UID_i	Unique identity of $\mathcal{AAV}_i$	GID_k	Unique identity of $\mathcal{GCS}_k$
ZID_k	Airspace zone identifier of sector k	msk	Master secret key of $\mathcal{TA}$
(ek_i, dk_i)	ML-KEM encaps./decaps. key pair	(sk_i^s, pk_i^s)	ML-DSA signing/verification key pair
KEM.KG$(\cdot)$	ML-KEM-768 key generation (FIPS 203)	KEM.Enc(ek)	ML-KEM-768 encapsulation
KEM.Dec(dk, ct)	ML-KEM-768 decapsulation	DSA.KG$(\cdot)$	ML-DSA-65 key generation (FIPS 204)
DSA.Sign(sk, m)	ML-DSA-65 signature on m	DSA.Vfy(pk, m, σ)	ML-DSA-65 verification
$H(\cdot)$	SHA3-256 hash function (256-bit output)	HKDF$(ikm, info)$	HKDF-SHA3-256 key derivation (256-bit output)
AEnc$_K(m)$	AES-256-GCM encryption	ADec$_K(c)$	AES-256-GCM decryption
N_i, N_k	256 bit cryptographic nonces	ts	Current timestamp
SK	Established session key	$Cert_i$	Certificate of entity i from $\mathcal{TA}$
$\mathcal{W}_k$	Local model weights at $\mathcal{GCS}_k$	$\tilde{\mathcal{W}}_k$	DP noised weights ($\mathcal{W}_k + \mathcal{N}(0, \sigma^2\mathbf{I})$)
η_i	Hardware secret of $\mathcal{AAV}_i$	ΔT	Max allowable timestamp skew
$\Rightarrow$ / $\rightarrow$	Secure / insecure channel	$\|$ / $\oplus$	Concatenation / XOR
$\mathcal{EVE}$	Quantum adversary	λ	Security parameter ($\lambda = 128$, NIST Level 3)

3.2. Lattice Based Hard Problems

The post quantum security guarantees of PQ-LITS rest on two lattice problems. Both have been studied extensively over the past decade and as far as the community can tell, appear to resist quantum polynomial time adversaries.

Module Learning with Errors (MLWE). Consider the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n+1)$, a matrix $\mathbf{A} \xleftarrow{\$} R_q^{k\times k}$, a secret vector $\mathbf{s} \xleftarrow{\$} R_q^k$, and a small error $\mathbf{e} \leftarrow \chi_\eta^k$ drawn from a centred binomial distribution with parameter η. The MLWE problem asks whether an adversary can distinguish the pair $(\mathbf{A},\ \mathbf{b} = \mathbf{As} + \mathbf{e})$ from a uniformly random pair $(\mathbf{A},\ \mathbf{u})$. Formally, the advantage is

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{MLWE}} = |\Pr[\mathcal{A}(\mathbf{A}, \mathbf{As}+\mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}) = 1]| \quad (1)$$

and we assume it remains negligible in the security parameter λ. Current lattice estimation tools suggest that the best quantum attacks on the ML-KEM-768 parameters ($n=256$, $k = 3$, $q = 3329$, $\eta = 2$) likely require on the order of 2^{170} operations [26, 35]. That said, this figure may shift as estimation methodologies mature.

Module Short Integer Solution (MSIS). Given a uniformly random $\mathbf{A} \xleftarrow{\$} R_q^{k\times\ell}$, the MSIS problem asks to find a nonzero short vector $\mathbf{z} \in R_q^\ell$ satisfying

$$\mathbf{A}\cdot\mathbf{z} = \mathbf{0} \pmod q \quad \text{with} \quad \|\mathbf{z}\| \le \beta \quad (2)$$

In the module setting, the hardness of MSIS reduces to worst case approximate shortest vector problems [36]. ML-DSA-65 ties its signature unforgeability directly to this assumption [37].

3.3. NIST Post Quantum Primitives

We now describe the three NIST standardised cryptographic families used in our construction. Readers already comfortable with FIPS 203 and 204 may easily understand these concepts, but the parameter sizes are worth noting because they feed directly into the communication cost analysis of Section 6.

ML-KEM-768 (FIPS 203). ML-KEM is a lattice based key encapsulation mechanism that achieves IND CCA2 security under the MLWE assumption [11]. The key generation algorithm KEM.KG produces an encapsulation key ek and a decapsulation key dk from a public matrix $\mathbf{A}$ and secret/error vectors sampled from the centred binomial distribution. Encapsulation via KEM.Enc(ek) outputs a ciphertext together with a 32 byte shared secret derived through fresh randomness hashed internally. Decapsulation via KEM.Dec(dk, ct) recovers the shared secret by re encrypting and comparing. When the comparison fails a pseudorandom value is returned instead. This implicit rejection mechanism is what the Fujisaki Okamoto transform [38] uses to prevent chosen ciphertext attacks. At NIST Level 3, the encapsulation key occupies 1184 bytes, the ciphertext 1088 bytes, and the shared secret 32 bytes.

ML-DSA-65 (FIPS 204). ML-DSA is a lattice based digital signature scheme whose EUF CMA security reduces jointly to MLWE and MSIS [12]. Key generation produces a verification key and a signing key from a public matrix and bounded secret vectors. The signing procedure itself deserves brief mention because it is not straightforward. It runs a rejection sampling loop where a masking vector hides the secret key behind random noise. If the resulting signature does not satisfy certain norm bounds the algorithm discards the attempt and restarts with fresh randomness. On average, ML-DSA-65 needs about 4.25 iterations before it produces a valid signature [37]. This repeated sampling is what makes signing the most expensive individual operation in our benchmarks, at 0.138 ms on the test platform (see

Table 2
Comparative Synthesis of IoD Authentication and Anomaly Detection Protocols

Ref.	Year	Applied Primitives	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8	SF9	SF10	Eff. ↑
Symmetric Cryptographic Solutions													
[17]	2020	Hash	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	High
[19]	2021	AES, Fuzzy Ext., Hash	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Low
[21]	2022	AES, Hash	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
[23]	2023	Chaotic Map, AES, Hash	⊙	⊙	⊙	⊙	⊕	⊙	⊙	⊙	⊙	⊙	Mod.
Public Key Infrastructure (PKI) Solutions													
[6]	2021	ECC, Hash	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
[5]	2021	ECC, Hash	⊙	⊙	⊙	⊙	⊕	⊙	⊙	⊙	⊙	⊙	Low
[24]	2023	Hyper-ECC, Hash	⊙	⊖	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Low
[7]	2023	R-LWE, Hash	⊙	⊖	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
[8]	2024	R-LWE, Hash	⊙	⊖	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
PUF-Based Solutions													
[27]	2022	PUF, Fuzzy Ext., Hash	⊕	⊙	⊕	⊙	⊙	⊕	⊙	⊙	⊙	⊙	Low
[28]	2023	PUF, Hash	⊕	⊙	⊕	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
[29]	2023	PUF, Hash	⊙	⊙	⊕	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Low
[30]	2024	PUF, AEAD, Hash	⊙	⊙	⊕	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
[31]	2024	Chaotic Map, PUF, Hash	⊙	⊙	⊕	⊙	⊙	⊙	⊙	⊙	⊙	⊙	Mod.
Post-Quantum Solutions													
[16]	2025	Kyber-512, RO-PUF, BCH-ECC, AES-GCM	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊙	⊙	High
Federated Anomaly Detection													
[32]	2023	FL, DP, SMPC, CNN-LSTM	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊕	⊕	Mod.
[33]	2025	FL, DNN	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊕	⊙	Mod.
PQ-LITS	**2026**	**ML-KEM-768, ML-DSA-65, AES-256-GCM, HKDF, FL, DP**	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	**High**

SF1: Mutual Auth. SF2: PQ Resilience (NIST) SF3: Device Capture Resist. SF4: Replay Protect. SF5: Forward Secrecy
SF6: Impersonation Resist. SF7: Zone Binding SF8: Stolen Verifier Resil. SF9: Telemetry Privacy (FL) SF10: Gradient Privacy (DP)
⊕: Covered ⊖: Partial ⊙: Not available Eff.: Computational efficiency rating based on reported computation cost

Section 6). The verification key is 1952 bytes, the signing key 4032 bytes, and each signature 3309 bytes.

Symmetric Primitives. Three well understood symmetric tools complete the construction. SHA3-256, built on the Keccak sponge as specified in FIPS 202 [39], provides collision resistance at 2^{128} even against Grover accelerated search. HKDF-SHA3-256 following RFC 5869 [40] derives session keys from partially weak input material through an extract then expand paradigm. AES-256-GCM, per SP 800-38D [41] handles authenticated encryption of telemetry payloads and key confirmation messages retaining a comfortable 128 bit quantum security margin.

3.4. Federated Learning with Differential Privacy

This subsection introduces the machine learning components that form the second layer of PQ-LITS, namely the federated training pipeline and the differential privacy mechanism layered on top of it.

Federated Learning (FL) allows K GCS clients to collaboratively train a shared anomaly detection model without ever pooling their raw telemetry [13]. In each communication round r, the federation server broadcasts the current global weights $\mathcal{W}_G^{(r-1)}$. Every client initialises its local model with those weights trains for E epochs on private data $\mathcal{D}_k$, and ships the updated parameters back. The server then computes a weighted average and the cycle repeats for R rounds. What actually travels over the network is remarkably small. In our setup, it amounts to 443 float32 parameters, totalling just 1.73 KB per model snapshot.

$$\mathcal{W}_G^{(r)} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{\sum_{j=1}^{K} |\mathcal{D}_j|} \cdot \mathcal{W}_k^{(r)} \tag{3}$$

Each GCS trains a lightweight autoencoder with architecture $12 \to 10 \to 6 \to 3 \to 6 \to 10 \to 12$ on normal flight telemetry only. The training objective is the mean squared reconstruction error

$$\mathcal{L}_k = \frac{1}{|\mathcal{D}_k|} \sum_{\mathbf{x} \in \mathcal{D}_k} \|\mathbf{x} - \mathcal{M}_k(\mathbf{x})\|_2^2 \tag{4}$$

At inference time, a telemetry record $\mathbf{x}^*$ is flagged as anomalous whenever $\|\mathbf{x}^* - \mathcal{M}_k(\mathbf{x}^*)\|_2^2$ exceeds a threshold θ optimised on a held out validation set. Attacks like GPS spoofing and RF jamming distort certain features in ways the autoencoder was never trained to reconstruct, which is precisely what pushes the error above θ.

FL keeps raw data local but it is not entirely free of privacy risk. Gradient inversion attacks have demonstrated that a curious server can, in certain settings, reconstruct training samples from uploaded model updates [14]. We mitigate this by adding calibrated Gaussian noise before each upload

$$\tilde{\mathcal{W}}_k^{(r)} = \mathcal{W}_k^{(r)} + \mathcal{N}(0,\ \sigma_{DP}^2 \cdot \mathbf{I}) \tag{5}$$

where the noise scale σ_{DP} is calibrated to satisfy (ε, δ) differential privacy under the Gaussian mechanism [15]. More concretely the mechanism requires $\sigma_{DP} \geq \Delta_2 \sqrt{2\ln(1.25/\delta)}\,/\,\varepsilon$, where Δ_2 is the ℓ_2 sensitivity bounded by gradient clipping. For the reported experiments, we set $\delta = 10^{-5}$, clipping bound $C = \Delta_2 = 1$, and $\sigma_{DP} = 0.01$; under the standard single-release Gaussian bound, this corresponds to $\varepsilon \approx 484.5$, so the current setting should be read as a lightweight empirical privacy perturbation rather than a tight privacy budget. Taken together, the AES-256-GCM encrypted channel and the DP perturbation make it unlikely that either a passive eavesdropper or a curious federation

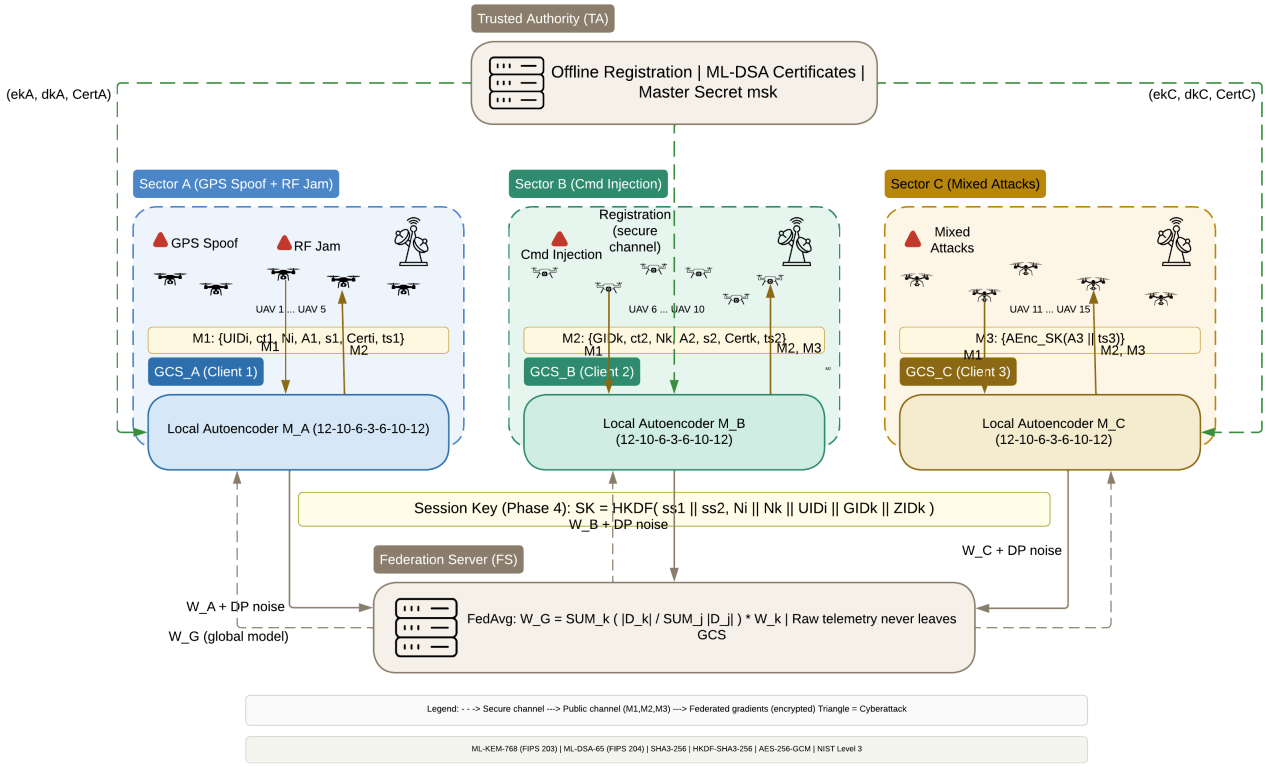

Figure 1: System architecture of the proposed PQ-LITS framework.

server could recover private flight trajectories from the model exchange.

3.5. System Model

The PQ-LITS architecture spans three low altitude airspace sectors and involves four types of entities as illustrated in Fig. 1.

Trusted Authority ($\mathcal{TA}$). This is an offline entity responsible for generating post quantum key pairs and issuing ML-DSA certificates during one time registration. It plays no role in real time authentication or federated training, much like a Certificate Authority in classical PKI but with lattice based credentials in place of RSA or ECDSA.

Autonomous Aerial Vehicles ($\mathcal{AAV}_i$). These are resource constrained platforms equipped with GPS, inertial sensors, an RF transceiver, and a companion computer (Raspberry Pi 5 class) running the PQ-LITS cryptographic stack. Fifteen AAVs are spread across three sectors, five per sector. Private keys are stored XOR-masked with a hardware secret η_i rather than in plaintext, which is intended to offer a degree of protection if the device is physically captured.

Ground Control Stations ($\mathcal{GCS}_k$). Three stations are deployed, one per airspace sector. Each station authenticates incoming AAVs, collects encrypted telemetry, trains a local anomaly detection model, and participates in federated aggregation rounds with the central server.

Federation Server ($\mathcal{FS}$). This entity aggregates DP noised model weights from all three GCS clients using FedAvg. It never sees raw telemetry. Communication between each GCS and the $\mathcal{FS}$ is protected by a session key established through the same Phase 4 mutual authentication protocol used for AAV to GCS links.

We assume registration between the $\mathcal{TA}$ and all other entities occurs over a physically secure channel. Everything after that point, AAV-to-GCS and GCS-to-FS, travels over open wireless or internet links that an adversary can freely observe, intercept, and tamper with.

3.6. Threat Model

We adopt the Dolev-Yao adversary model [42], extended to include quantum computational capabilities. This appears to be the appropriate fit for a LITS environment where communication is predominantly wireless and adversaries may eventually gain access to quantum hardware.

Adversary Capabilities. The adversary $\mathcal{EVE}$ is modelled as a probabilistic polynomial time quantum attacker. $\mathcal{EVE}$ can eavesdrop on all messages exchanged among $\mathcal{AAV}_i$, $\mathcal{GCS}_k$, and $\mathcal{FS}$ over public channels. $\mathcal{EVE}$ may also intercept, modify, replay, or inject arbitrary messages at will. Beyond passive observation, $\mathcal{EVE}$ is permitted to compromise the long term keys of at most one party, though not those of the $\mathcal{TA}$. Physical capture of a AAV and subsequent extraction of stored credentials from memory is also within scope. On the quantum front, $\mathcal{EVE}$ may run Shor's algorithm against any classical public key primitive and apply Grover's algorithm for brute-force key search. Finally, $\mathcal{EVE}$

can mount cyberphysical attacks against AAV telemetry, including GPS spoofing, RF jamming, and command injection.

Trust Assumptions. The $\mathcal{TA}$ is fully trusted and operates in a secure offline environment. The $\mathcal{FS}$, by contrast, is honest but curious. It follows the aggregation protocol faithfully but may attempt to learn private information from the model updates it receives. Hardware level secrets η_i in AAV secure elements are assumed to resist physical extraction under standard tamper resistance assumptions.

Security Goals. Given the capabilities above, the protocol targets nine properties. First, it must provide mutual authentication between $\mathcal{AAV}_i$ and $\mathcal{GCS}_k$ before any data flows (G1). Session keys must be computationally indistinguishable from random to any external observer (G2). Fresh KEM encapsulations in every session are meant to guarantee perfect forward secrecy (G3), while timestamps and nonces handle replay resistance (G4). Zone binding (G5) prevents an adversary from reusing a session key across different airspace sectors. Post quantum security at NIST Level 3 (G6) is the baseline cryptographic target. On the data side, raw telemetry must never leave the GCS boundary (G7), and (ε, δ)-differential privacy on uploaded model weights ensures gradient privacy (G8). Finally, XOR masked key storage tied to the hardware secret η_i is designed to resist device capture attacks (G9).

4. Proposed PQ-LITS Protocol

This section presents the complete PQ-LITS protocol across six phases, beginning with a high level design rationale before specifying each phase in detail.

4.1. Design Rationale

Two guiding principles shaped the protocol. The first is separation of concerns. Authentication and key transport are handled entirely by NIST standardised post quantum primitives (ML-KEM-768 for encapsulation, ML-DSA-65 for signatures), while telemetry integrity is the job of a federated autoencoder operating over the authenticated channel. Neither layer depends on the implementation details of the other, which means the cryptographic handshake can be upgraded independently of the anomaly detector and the other way around.

The second principle is that no single point of compromise should unravel the entire system. The session key SK is derived from two independently generated shared secrets (ss_1 from the AAV direction, ss_2 from the GCS direction), so compromising one KEM instance alone is not enough. Long term signing keys serve only authentication and never key transport, preserving forward secrecy. On the machine learning side, raw telemetry never crosses the GCS boundary, and even the model weights are DP noised before transmission. An honest but curious federation server should learn nothing useful about individual flight records.

4.2. Phase 1: System Initialisation

Phase 1 runs exactly once, in a secure offline environment before any AAV or GCS is deployed. The $\mathcal{TA}$ selects the security parameter $\lambda = 128$ (NIST Level 3) generates its own ML-DSA-65 signing key pair $(sk^s_{\mathcal{TA}},\ pk^s_{\mathcal{TA}}) \leftarrow \mathsf{DSA.KG}(1^\lambda)$, and fixes SHA3-256 as the system hash function. It then samples a master secret $msk \xleftarrow{\$} \{0,1\}^{256}$ for use in registration tokens, defines the zone identifiers $\{ZID_k\}_{k=1}^K$ mapping each $\mathcal{GCS}_k$ to its sector, and publishes the public parameters $params = \{pk^s_{\mathcal{TA}}, H, \mathsf{HKDF}, \lambda, \{ZID_k\}_{k=1}^K\}$. The master secret and the $\mathcal{TA}$'s signing key are stored in tamper resistant hardware and never exposed to any network interface. All intermediate values are erased from volatile memory.

4.3. Phase 2: AAV Registration

Before deployment, each $\mathcal{AAV}_i$ registers with the $\mathcal{TA}$ through a secure out of band channel. This registration phase binds the AAV's physical identity to its post quantum key material and its assigned airspace sector.

The AAV retrieves its hardware embedded secret η_i (a device specific fuse value burned during manufacturing, functionally comparable to a PUF response) and computes a masked identity token $HUID_i = H(UID_i \| \eta_i)$. This token is sent to the $\mathcal{TA}$ alongside the plaintext UID_i. The $\mathcal{TA}$ generates both an ML-KEM-768 key pair (ek_i, dk_i) and an ML-DSA-65 signing key pair (sk^s_i, pk^s_i) on behalf of the AAV, computes a registration token $\tau_i = H(UID_i \| HUID_i \| msk \| ZID_k)$ binding the identity to the master secret and the zone, assigns an expiry field exp_i, and issues a certificate $Cert_i = \mathsf{DSA.Sign}(sk^s_{\mathcal{TA}}, UID_i \| pk^s_i \| ek_i \| ZID_k \| exp_i \| \tau_i)$. The same expiry field is included in GCS certificates. To avoid mandatory key escrow, an AAV-side generation variant is also supported: the AAV may generate (ek_i, dk_i) and (sk^s_i, pk^s_i) locally and send only $(UID_i, HUID_i, ek_i, pk^s_i)$ to the $\mathcal{TA}$ for certification.

Rather than writing dk_i and sk^s_i directly to non volatile memory which would leave them exposed if an adversary physically captures the drone, the AAV XOR masks them with hash values derived from η_i.

$$\alpha_i = dk_i \oplus H(\eta_i \| UID_i \| \texttt{0x01}) \tag{6}$$

$$\beta_i = sk^s_i \oplus H(\eta_i \| UID_i \| \texttt{0x02}) \tag{7}$$

The plaintext private keys are then erased from RAM. An adversary who dumps the drone's flash memory obtains α_i and β_i, but recovering the actual keys requires knowledge of η_i, which is bound to the hardware and not stored in software accessible memory. The domain separation constants (`0x01` and `0x02`) prevent a related key attack where one masked value could be used to recover the other. The full step by step specification appears in Fig. 2.

4.4. Phase 3: GCS Registration

GCS registration follows the same structure as Phase 2 (Fig. 2), with two differences worth noting. First, the $\mathcal{GCS}_k$ generates its own local master secret $MS_k \xleftarrow{\$} \{0,1\}^{256}$

rather than relying on a hardware embedded fuse value, since ground stations are assumed to operate in physically secure facilities. The masked identity token becomes $HGID_k = H(GID_k \| MS_k)$. Second, after generating the KEM and DSA key pairs for $\mathcal{GCS}_k$, the $\mathcal{TA}$ provides the set of public keys $\mathcal{P}_k = \{(UID_i, ek_i, pk_i^s, exp_i)\}_{i \in \text{Sec}_k}$ for all AAVs assigned to sector k, enabling the GCS to verify incoming authentication requests without contacting the $\mathcal{TA}$ in real time. The GCS stores its private keys using AES-256-GCM encryption under MS_k rather than XOR masking reflecting the stronger storage guarantees available at a ground station. At the end of registration, the GCS also initialises its local autoencoder $\mathcal{M}_k$ with random weights $\mathcal{W}_k^{(0)}$ for federated training.

Figure 2: Phase 2: AAV Registration ($\mathcal{AAV}_i$ registers with $\mathcal{TA}$)

$\mathcal{AAV}_i$		$\mathcal{TA}$
S1: Select unique identity UID_i and retrieve hardware secret η_i (device fuse value).		
S2: Compute masked token: $HUID_i = H(UID_i \| \eta_i)$		
S3: Transmit to $\mathcal{TA}$.	$\xrightarrow{UID_i,\ HUID_i}$	
		S4: Verify $UID_i \notin$ registry; else abort.
		S5: $(ek_i, dk_i) \leftarrow \mathsf{KEM.KG}(1^\lambda)$
		S6: $(sk_i^s, pk_i^s) \leftarrow \mathsf{DSA.KG}(1^\lambda)$
		S7: Assign $\mathcal{AAV}_i$ to sector k; retrieve ZID_k.
		S8: $\tau_i = H(UID_i \| HUID_i \| msk \| ZID_k)$
		S9: Compute certificate: $Cert_i = \mathsf{DSA.Sign}(sk_{\mathcal{TA}}^s, UID_i \| pk_i^s \| ek_i \| ZID_k \| exp_i \| \tau_i)$
		S10: Store in registry: $(UID_i, pk_i^s, ek_i, ZID_k, exp_i, \tau_i, HUID_i)$
	$\xleftarrow{\text{creds}}$	**S11:** Send back: $\{dk_i, sk_i^s, Cert_i, ek_i, pk_i^s, exp_i, \tau_i, ZID_k, pk_{\mathcal{TA}}^s\}$
S12: Protect private keys: $\alpha_i = dk_i \oplus H(\eta_i \| UID_i \| 01)$ $\beta_i = sk_i^s \oplus H(\eta_i \| UID_i \| 02)$		
S13: Store $\{\alpha_i, \beta_i, Cert_i, ek_i, pk_i^s, exp_i, \tau_i, ZID_k, pk_{\mathcal{TA}}^s\}$ in NVM. Erase dk_i, sk_i^s from RAM.		

4.5. Phase 4: Mutual Authentication and Key Agreement

This is the core of the protocol and runs every time an AAV needs to establish a fresh session with its assigned GCS. It consists of three messages exchanged over the public wireless channel. We split the description into two parts for clarity. Phase 4a covers the authentication request (M_1) and its verification, while Phase 4b covers the key agreement response (M_2), the AAV's verification of M_2, and the explicit key confirmation (M_3). The complete step by step specifications appear in Figs. 3 and 4.

Figure 3: Phase 4a: Authentication Request and Verification ($\mathcal{AAV}_i \rightarrow \mathcal{GCS}_k$)

$\mathcal{AAV}_i$		$\mathcal{GCS}_k$
S1: Recover private keys via η_i: $dk_i = \alpha_i \oplus H(\eta_i \| UID_i \| 01)$ $sk_i^s = \beta_i \oplus H(\eta_i \| UID_i \| 02)$		
S2: Fresh nonce: $N_i \xleftarrow{\$} \{0,1\}^{256}$		
S3: Retrieve ek_k of $\mathcal{GCS}_k$; encapsulate: $(ct_1, ss_1) \leftarrow \mathsf{KEM.Enc}(ek_k)$		
S4: Record ts_1; compute zone-bound authenticator: $A_1 = H(UID_i \| N_i \| ss_1 \| ts_1 \| ZID_k \| \tau_i)$		
S5: Sign via ML-DSA: $\sigma_1 = \mathsf{DSA.Sign}(sk_i^s, ct_1 \| A_1 \| ts_1 \| ZID_k)$		
S6: Send M_1.	$\xrightarrow{M_1}$	
		S7: Check freshness: $\lvert ts_{\text{now}} - ts_1 \rvert \le \Delta T$. Abort if stale.
		S8: Verify $\mathcal{TA}$ certificate: $\mathsf{DSA.Vfy}(pk_{\mathcal{TA}}^s, UID_i \| pk_i^s \| ek_i \| ZID_k \| exp_i \| \tau_i, Cert_i) \stackrel{?}{=} 1$ Reject on failure.
		S9: Verify request signature: $\mathsf{DSA.Vfy}(pk_i^s, ct_1 \| A_1 \| ts_1 \| ZID_k, \sigma_1) \stackrel{?}{=} 1$ Terminate on failure.
		S10: Decapsulate: $ss_1 = \mathsf{KEM.Dec}(dk_k, ct_1)$ Abort if $\perp$.
		S11: Recompute and verify: $A_1 \stackrel{?}{=} H(UID_i \| N_i \| ss_1 \| ts_1 \| ZID_k \| \tau_i)$ Failure indicates tampered ciphertext or forged zone binding. If valid, $\mathcal{AAV}_i$ is authenticated.
$\mathcal{AAV}_i$ authenticated. $\mathcal{GCS}_k$ proceeds to Phase 4b.		

$M_1 = \{UID_i, ct_1, N_i, A_1, \sigma_1, Cert_i, ts_1\}$, with ZID_k recovered from $Cert_i$ or the registry

4.5.1. Phase 4a: Authentication Request (M_1)

The AAV initiates the handshake by recovering its private keys from XOR masked storage using η_i, generating a fresh 256 bit nonce N_i, and performing an ML-KEM-768 encapsulation toward the GCS.

$$(ct_1, ss_1) \leftarrow \mathsf{KEM.Enc}(ek_k) \tag{8}$$

This yields a ciphertext ct_1 (1,088 bytes) and a shared secret ss_1 (32 bytes) that only the holder of dk_k can recover. The AAV then computes a zone bound authenticator $A_1 = H(UID_i \| N_i \| ss_1 \| ts_1 \| ZID_k \| \tau_i)$ that ties the shared secret to the AAV's identity, the nonce, the timestamp, and the zone. It signs the payload and transmits

$$M_1 = \{UID_i, ct_1, N_i, A_1, \sigma_1, Cert_i, ts_1\} \tag{9}$$

Here ZID_k is not sent as a separate plaintext field; the GCS recovers it from $Cert_i$ or its local registry before verifying A_1 and σ_1. Upon receiving M_1, the GCS performs four checks in sequence. It verifies timestamp freshness (S7), the $\mathcal{TA}$'s certificate binding (S8), the AAV's signature under pk_i^s (S9), and finally recomputes the authenticator after

decapsulating ss_1 (S10–S11). All four must pass before the AAV is considered authenticated.

Figure 4: Phase 4b: Session Key Agreement and Explicit Confirmation

$\mathcal{AAV}_i$		$\mathcal{GCS}_k$
Message M_2: Key Agreement Response		
		S12: Fresh nonce: $N_k \xleftarrow{\$} \{0,1\}^{256}$ **S13:** Reciprocal encapsulation: $(ct_2, ss_2) \leftarrow \mathsf{KEM.Enc}(ek_i)$ **S14:** Derive session key: $SK = \mathsf{HKDF}(\ ss_1 \| ss_2,\ N_i \| N_k \| \ UID_i \| GID_k \ \| ZID_k)$ **S15:** Authenticator and signature: $A_2 = H(GID_k \ \| N_k \| ss_2 \ \| SK \| ts_2 \ \| ZID_k)$ $\sigma_2 = \mathsf{DSA.Sign}(\ sk^s_k,\ ct_2 \| A_2 \ \| ts_2 \| ZID_k)$
	$\xleftarrow{M_2}$	**S16:** Send M_2.
$\mathcal{AAV}_i$ **Verifies** M_2		
S17: Check freshness: $\lvert ts_{\text{now}} - ts_2 \rvert \le \Delta T$ **S18:** Verify $Cert_k$: $\mathsf{DSA.Vfy}(\ pk^s_{\mathcal{TA}},\ GID_k \| pk^s_k \ \| ek_k \| ZID_k \ \| exp_k \| \tau_k,\ Cert_k) \stackrel{?}{=} 1$ **S19:** Verify signature: $\mathsf{DSA.Vfy}(\ pk^s_k,\ ct_2 \| A_2 \ \| ts_2 \| ZID_k, \sigma_2) \stackrel{?}{=} 1$ **S20:** Decapsulate: $ss_2 = \mathsf{KEM.Dec}(\ dk_i, ct_2)$ **S21:** Derive independently: $SK = \mathsf{HKDF}(\ ss_1 \| ss_2,\ N_i \| N_k \| \ UID_i \| GID_k \ \| ZID_k)$ **S22:** Verify A_2: $A_2 \stackrel{?}{=} H(GID_k \ \| N_k \| ss_2 \ \| SK \| ts_2 \ \| ZID_k)$ Abort if any check S17–S22 fails.		
Message M_3: Explicit Key Confirmation		
S23: Compute confirmation: $A_3 = H(SK \| N_i \ \| N_k \| UID_i \ \| GID_k \| \mathsf{KC})$ $C_3 = \mathsf{AEnc}_{SK}(\ A_3 \| ts_3)$		
S24: Send M_3.	$\xrightarrow{M_3}$	**S25:** Decrypt: $(A_3 \| ts_3) = \mathsf{ADec}_{SK}(C_3)$ **S26:** Verify: $A_3 \stackrel{?}{=} H(SK \ \| N_i \| N_k \ \| UID_i \| GID_k \ \| \mathsf{KC})$ If valid, erase ss_1, ss_2 from RAM. Retain only SK.
Session key SK confirmed. Telemetry encrypted as $\mathsf{AEnc}_{SK}(data \| seq)$.		

$M_2 = \{GID_k, ct_2, N_k, A_2, \sigma_2, Cert_k, ts_2\}$, with ZID_k recovered from $Cert_k$ or the registry, $M_3 = \{C_3\}$

4.5.2. Phase 4b: Key Agreement and Key Confirmation (M_2, M_3)

With $\mathcal{AAV}_i$ authenticated, the GCS performs a reciprocal KEM encapsulation toward the AAV.

$$(ct_2, ss_2) \leftarrow \mathsf{KEM.Enc}(ek_i) \tag{10}$$

This dual KEM structure is not merely redundant. It ensures that even if an adversary manages to break one encapsulation (say, by obtaining dk_k through a side channel attack on the GCS) the session key still depends on ss_2, which requires dk_i to recover. The session key derivation follows

$$SK = \mathsf{HKDF}\big(ss_1 \| ss_2,\ \ N_i \| N_k \| UID_i \| GID_k \| ZID_k\big) \tag{11}$$

The HKDF context string includes both nonces, both identities and the zone identifier providing domain separation across sessions, entities, and airspace sectors. The GCS signs its response and transmits

$$M_2 = \{GID_k, ct_2, N_k, A_2, \sigma_2, Cert_k, ts_2\} \tag{12}$$

Similarly, the AAV recovers ZID_k from $Cert_k$ or the cached registry material before checking the signed M_2 fields.

The AAV mirrors the GCS's verification sequence (certificate, signature, decapsulation, authenticator), independently derives SK using Eq. (11), and verifies A_2. If everything checks out, it sends a key confirmation:

$$M_3 = \big\{C_3 = \mathsf{AEnc}_{SK}(A_3 \| ts_3)\big\} \tag{13}$$

where $A_3 = H(SK \| N_i \| N_k \| UID_i \| GID_k \| \mathsf{KC})$. The literal string KC acts as a domain separator, preventing the confirmation hash from colliding with any authenticator hash. The GCS decrypts C_3, recomputes A_3, and verifies the match. Both sides then erase ss_1 and ss_2 from volatile memory, retaining only SK.

Why three messages instead of two? A two message protocol without explicit key confirmation leaves the GCS uncertain whether the AAV actually derived the same SK. In a LITS deployment where telemetry starts flowing immediately after authentication, a mismatched key would corrupt the first several packets before the error surfaces. The 68 byte M_3 eliminates this race condition at negligible bandwidth cost.

4.6. Phase 5: Secure Federated Model Update Exchange

Once the authenticated channel is established, the GCS begins collecting encrypted telemetry from its assigned AAVs. Over time, this data trains a local anomaly detection autoencoder. Phase 5 specifies how these local models are aggregated across GCS clients without exposing raw telemetry to any central entity. The full specification is in Fig. 5.

The federation operates in rounds. At each round r, the $\mathcal{FS}$ broadcasts the encrypted global model to each $\mathcal{GCS}_k$ under their respective session keys $SK_{k,\mathcal{FS}}$. Each GCS initialises its local model, trains for E epochs on private normal flight data, and applies the Gaussian mechanism for differential privacy (Eq. 5). The noisy weights are encrypted and uploaded, and the $\mathcal{FS}$ performs weighted FedAvg aggregation (Eq. 3). The exchanged object is therefore a DP-noised model update, not a raw gradient vector.

Figure 5: Phase 5: Secure Federated Model Update Exchange ($\mathcal{GCS}_k \longleftrightarrow \mathcal{FS}$)

$\mathcal{GCS}_k$ **(Client)**		$\mathcal{FS}$ **(Server)**
Prerequisite: $\mathcal{GCS}_k$ and $\mathcal{FS}$ share session key $SK_{k,\mathcal{FS}}$ via Phase 4.		
Round r **of Federated Averaging**		
		S1: Broadcast encrypted global model to each client k: $C_G^{(r)} = \mathsf{AEnc}_{SK_{k,\mathcal{FS}}}(\mathcal{W}_G^{(r-1)} \| r \| ts)$
	$\xleftarrow{C_G^{(r)}}$	
S2: Decrypt: $(\mathcal{W}_G^{(r-1)} \| r \| ts) = \mathsf{ADec}_{SK_{k,\mathcal{FS}}}(C_G^{(r)})$. Verify round number r and timestamp freshness. **S3:** Initialise local autoencoder with global weights: $\mathcal{M}_k \leftarrow \mathcal{W}_G^{(r-1)}$ **S4:** Train $\mathcal{M}_k$ for E local epochs on private telemetry $\mathcal{D}_k$ (normal flight records from Phase 4). Local loss: $\mathcal{L}_k = \frac{1}{\lvert\mathcal{D}_k\rvert} \sum_{x \in \mathcal{D}_k} \lVert x - \mathcal{M}_k(x) \rVert_2^2$ Updated weights $\mathcal{W}_k^{(r)}$. **S5:** Apply (ε, δ)-DP via calibrated Gaussian noise: $\tilde{\mathcal{W}}_k^{(r)} = \mathcal{W}_k^{(r)} + \mathcal{N}(0, \sigma_{DP}^2 \cdot \mathbf{I})$ σ_{DP} satisfies privacy budget ε under the Gaussian mechanism.		
S6: Encrypt noised update: $C_k^{(r)} = \mathsf{AEnc}_{SK_{k,\mathcal{FS}}}(\tilde{\mathcal{W}}_k^{(r)} \| r \| GID_k \| ts')$	$\xrightarrow{C_k^{(r)}}$	
		S7: Decrypt each $C_k^{(r)}$, verify round number and freshness, extract $\tilde{\mathcal{W}}_k^{(r)}$. **S8:** Federated Averaging: $\mathcal{W}_G^{(r)} = \sum_{k=1}^{K} \frac{\lvert\mathcal{D}_k\rvert}{\sum_j \lvert\mathcal{D}_j\rvert} \cdot \tilde{\mathcal{W}}_k^{(r)}$ $\mathcal{FS}$ never observes raw $\mathcal{D}_k$; only encrypted, DP noised parameters. **S9:** Proceed to round r+1 (return to S1). After R rounds, broadcast final global model.
After R rounds, each $\mathcal{GCS}_k$ deploys the global anomaly detector locally. Raw telemetry never leaves the $\mathcal{GCS}$ boundary. Only 443 DP noised float32 parameters are exchanged per round.		

Three layers of protection operate simultaneously. The transport layer (AES-256-GCM under $SK_{k,\mathcal{FS}}$) prevents eavesdroppers from reading the model weights. The differential privacy layer (Eq. (5)) prevents the honest but curious $\mathcal{FS}$ from reconstructing individual training samples via gradient inversion [14] and the federated structure itself ensures that raw telemetry never leaves the GCS. The autoencoder architecture is deliberately compact ($12 \to 10 \to 6 \to 3 \to 6 \to 10 \to 12$, just 443 parameters or 1.73 KB), which is not a limitation but a design choice. Smaller models are harder to invert [14], transmit faster over constrained radio links, and converge in fewer federation rounds.

4.7. Phase 6: Zone Handover

In a multi sector LITS deployment AAVs frequently transition between GCS coverage areas as they follow delivery routes or surveillance patterns. Phase 6 handles this without requiring a full re registration with the $\mathcal{TA}$.

The departing AAV computes a handover token $HT = H(SK_{old} \| UID_i \| ZID_k \| ZID_{k'} \| N_i')$ that cryptographically chains the old session to the new one. This token is included in M_1' sent to the new sector's $\mathcal{GCS}_{k'}$, which contacts the old $\mathcal{GCS}_k$ to verify it. If the old GCS does not confirm HT, the new GCS aborts before deriving SK_{new}. The rest of the handover follows Steps S7 through S26 of Phase 4, producing a new session key SK_{new} bound to $ZID_{k'}$. The old session key is erased from memory once M_3' is confirmed. Why not simply run Phase 4 from scratch? Two reasons. The handover token proves to $\mathcal{GCS}_{k'}$ that the AAV was legitimately operating in sector k immediately before preventing an adversary from injecting a rogue AAV by replaying credentials from a different context. And the inter GCS verification of HT allows $\mathcal{GCS}_k$ to log the departure maintaining an audit trail across sectors. The handover completes in three messages, the same as a fresh authentication, but the additional HT provides session continuity. The full specification is in Fig. 6.

Figure 6: Phase 6: Lightweight Session Key Update for Zone Handover

$\mathcal{AAV}_i$		$\mathcal{GCS}_{k'}$ **(new)**
Triggered when $\mathcal{AAV}_i$ **crosses from sector** k **to sector** k'		
S1: Detect zone transition (GPS or command from $\mathcal{GCS}_k$). Still holds valid credentials and old session key SK_{old} with $\mathcal{GCS}_k$. **S2:** Fresh nonce $N_i' \xleftarrow{\$} \{0,1\}^{256}$; encapsulate toward $\mathcal{GCS}_{k'}$: $(ct_1', ss_1') \leftarrow \mathsf{KEM.Enc}(ek_{k'})$ **S3:** Compute handover token chaining old and new sessions: $HT = H(SK_{old} \| UID_i \| ZID_k \| ZID_{k'} \| N_i')$		
S4: Authenticator and signature: $A_1' = H(UID_i \| N_i' \| ss_1' \| ts_1' \| ZID_{k'} \| HT)$ $\sigma_1' = \mathsf{DSA.Sign}(sk_i^s, ct_1' \| A_1' \| ts_1' \| ZID_{k'})$ Send M_1'.	$\xrightarrow{M_1'}$	
		S5: Process M_1' per Steps S7–S11 of Phase 4a. Additionally, contact $\mathcal{GCS}_k$ (via mutual session key) to verify HT, confirming $\mathcal{AAV}_i$ was legitimately in sector k. If HT cannot be confirmed, abort the handover. **S6:** If HT valid, proceed with S12–S16 of Phase 4b to establish SK_{new} bound to $ZID_{k'}$.
	$\xleftarrow{M_2'}$	**S7:** Send response M_2' as in Phase 4b.
S8: Complete verification (S17–S22 of Phase 4b) and send key confirmation M_3'. Erase SK_{old} from memory. All subsequent telemetry encrypted under SK_{new}.	$\xrightarrow{M_3'}$	
Zone handover completes in 3 messages (same as fresh authentication). The handover token HT provides cryptographic continuity with the previous sector session.		

$M_1' = \{UID_i, ct_1', N_i', A_1', HT, \sigma_1', Cert_i, ts_1'\}$

4.8. Post Handshake Telemetry Encryption

After Phase 4 (or Phase 6 for handovers), all telemetry between $\mathcal{AAV}_i$ and $\mathcal{GCS}_k$ is encrypted according to

$$C_{\text{telem}} = \text{AEnc}_{SK}(\text{features}\|\text{seq\_num}\|ts) \tag{14}$$

where "features" is the 12 dimensional telemetry vector (latitude, longitude, altitude, ground speed, vertical speed, heading, RSSI, SNR, packet interval, satellite count, battery voltage, command rate), "seq_num" is a monotonically increasing counter that prevents reordering attacks, and ts is a timestamp. The GCS decrypts each packet, feeds the features into the deployed anomaly detector and flags any record whose reconstruction error exceeds the detection threshold.

5. Security Analysis

This section evaluates PQ-LITS security through two complementary lenses. We first provide an informal, property by property argument that the protocol resists specific attacks then present a formal proof in the Quantum Random Oracle Model (QROM) [43] reducing security to the IND CCA property of ML-KEM-768, the EUF CMA property of ML-DSA-65, and the hardness of Module LWE.

5.1. Informal Security Analysis

We argue each claimed security property with respect to a quantum capable adversary $\mathcal{A}$ operating under the extended Dolev-Yao model.

Mutual Authentication. PQ-LITS achieves mutual authentication through a chain of four verification steps. The GCS verifies the TA issued certificate Cert_i under pk_{TA} (S8), the AAV's request signature σ_1 under pk_i^s (S9), and the zone bound authenticator A_1 binding ss_1, N_i, UID_i, and ZID_k (S11). In the reverse direction, the AAV verifies Cert_k (S18), σ_2 (S19), and A_2 that incorporates the session key SK (S22). An adversary impersonating either entity must forge an ML-DSA-65 signature without the signing key contradicting the EUF CMA security of FIPS 204 under the MSIS assumption.

Session Key Secrecy. The session key $SK = \text{HKDF}(ss_1\|ss_2, N_i\|N_k\|UID_i\|GID_k\|ZID_k)$ depends on two independently generated shared secrets, each from a separate ML-KEM-768 encapsulation. An adversary intercepting ct_1 and ct_2 must solve two independent IND CCA instances. Even if one encapsulation were somehow compromised, the second shared secret remains protected. The HKDF context string ensures domain separation across sessions.

Perfect Forward Secrecy. Each session uses fresh KEM encapsulations under the registered KEM keys on both sides. If $\mathcal{A}$ later compromises a long term signing key sk_i^s, this enables forging future signatures but does not help recover past shared secrets ss_1 and ss_2, which were derived from fresh encapsulation randomness and erased from RAM after key derivation (S24). Long term signing keys participate only in authentication, never in key transport.

Replay Attack Resistance. Each message M_1 and M_2 contains a fresh timestamp and nonce. The GCS rejects any M_1 where $|t_{\text{now}} - ts_1| > \Delta t$ (S7). Even if $\mathcal{A}$ replays M_1 within the freshness window, the authenticator $A_1 = H(UID_i\|N_i\|ss_1\|ts_1\|ZID_k\|\tau_i)$ includes the original timestamp already logged by the GCS. A replayed message fails the freshness or nonce-uniqueness check.

Zone Binding. The zone identifier ZID_k is embedded in three places. The TA issued certificate, the HKDF context string, and the authenticator hash. Redirecting an AAV to zone $ZID_{k'}$ by relaying M_2 from zone k causes the AAV to derive $SK' = \text{HKDF}(ss_1\|ss_2, \ldots\|ZID_{k'}) \neq SK$. The key confirmation step (M_3) will then fail, aborting the session.

Device Capture Resistance. The AAV stores $\alpha_i = dk_i \oplus H(\eta_i\|UID_i\|\texttt{0x01})$ and $\beta_i = sk_i^s \oplus H(\eta_i\|UID_i\|\texttt{0x02})$. An adversary who captures the drone obtains α_i and β_i but not the hardware bound secret η_i. Recovering dk_i amounts to a preimage search on SHA3-256, requiring $\mathcal{O}(2^{128})$ quantum queries under Grover's algorithm.

Telemetry and Gradient Privacy. Raw telemetry never leaves the GCS boundary. Phase 5 transmits only model weight updates, perturbed with Gaussian noise $\mathcal{N}(0, \sigma_{\text{DP}}^2 \cdot \mathbf{I})$ before encryption. The ($\varepsilon \approx 484.5, \delta = 10^{-5}$) accounting point with clipping bound $C = 1$ bounds the information any observer, including the honest but curious federation server can extract about individual telemetry records under the stated Gaussian mechanism [15]. The encrypted GCS to FS channel uses a session key established through the same Phase 4 protocol.

5.2. Formal Security Model

We adopt the Quantum Random Oracle Model (QROM) introduced by Boneh et al. [43] in which a quantum adversary $\mathcal{A}$ may query hash functions on arbitrary quantum superpositions of inputs. This is strictly stronger than the classical Random Oracle Model (ROM) used in prior IoD security proofs [16] because classical ROM proofs do not account for the adversary's ability to evaluate offline primitives in superposition [43]. Our formal treatment follows the framework of Hövelmanns et al. [44] for KEM based authenticated key exchange in the QROM, adapted to accommodate our dual-KEM structure with ML-DSA certificate verification.

5.2.1. Execution Environment

We model the protocol execution with n_U AAVs and n_G GCS nodes, each capable of running up to n_S sessions. Each AAV instance $\Pi_{U_i}^s$ and GCS instance $\Pi_{G_k}^t$ maintains a session state (ss_1, ss_2, SK, sid) and terminates in one of three states: ACCEPT, REJECT, or $\perp$ (incomplete).

5.2.2. Adversary Oracles

The adversary $\mathcal{A}$ may interact with the protocol through the following oracles, where all hash queries ($\mathcal{O}_H$) may be made in quantum superposition.

- $\mathcal{O}_H(|\psi\rangle)$: **Quantum Hash Oracle.** $\mathcal{A}$ submits a quantum state $|\psi\rangle = \sum_x \alpha_x |x, y\rangle$ and receives $\sum_x \alpha_x |x, y \oplus H(x)\rangle$. This models the adversary's ability to evaluate SHA3 256 and HKDF on superpositions. Up to q_H quantum queries are permitted.
- Send(Π_X^s, m): **Active Attack Oracle.** Delivers message m to instance Π_X^s. The instance processes m according to the protocol and returns the outgoing message (or $\perp$). Up to q_S classical queries are permitted.
- Corrupt(X): **Long Term Key Corruption.** Returns the long term signing key sk_X^s and the XOR masked decapsulation key (α_X or β_X). This models physical capture of the device. The hardware secret η_X is *not* returned.
- RevealSK(Π_X^s): **Session Key Reveal.** Returns the session key SK for a completed session. This models known key attacks.
- Test(Π_X^s): **Challenge Oracle.** Flips a coin $b \leftarrow \{0, 1\}$. If $b = 1$, returns the real session key SK. If $b = 0$, returns a uniformly random key $SK^* \leftarrow \{0, 1\}^{256}$. $\mathcal{A}$ outputs a guess b' and wins if $b' = b$. The tested session must be *fresh*: neither the test instance nor its partner has been the target of a RevealSK or Corrupt query (following the eCK freshness definition [45]).

Definition 1 (QROM-AKE Advantage)**.** *The advantage of $\mathcal{A}$ against the PQ-LITS protocol Π in the QROM is:*

$$\mathsf{Adv}_{\Pi}^{AKE\text{-}QROM}(\mathcal{A}) = \left| \Pr[b' = b] - \frac{1}{2} \right|. \tag{15}$$

The protocol is secure if $\mathsf{Adv}_{\Pi}^{AKE\text{-}QROM}(\mathcal{A})$ is negligible for all quantum polynomial time $\mathcal{A}$.

5.3. Main Security Theorem

Theorem 1. *Let $\mathcal{A}$ be a quantum polynomial time adversary against the PQ-LITS protocol Π in the QROM, making at most q_H quantum hash queries, q_S send queries, and q_C corrupt queries. Then there exist quantum polynomial time adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ such that:*

$$\begin{aligned} \mathsf{Adv}_{\Pi}^{\text{AKE-QROM}}(\mathcal{A}) \le\ & n_U \cdot \mathsf{Adv}_{\text{ML-KEM}}^{\text{IND-CCA}}(\mathcal{B}_1) \\ & + n_G \cdot \mathsf{Adv}_{\text{ML-KEM}}^{\text{IND-CCA}}(\mathcal{B}_2) \\ & + (n_U + n_G) \cdot \mathsf{Adv}_{\text{ML-DSA}}^{\text{EUF-CMA}}(\mathcal{B}_3) \\ & + \frac{(q_H + q_S)^2}{2^{256}} + \frac{q_S^2}{2^{256}} \\ & + n_U \cdot n_G \cdot \mathsf{Adv}_{\text{MLWE}}^{n,k,q}(\mathcal{B}_4) \end{aligned} \tag{16}$$

where n_U, n_G denote the number of AAV and GCS identities respectively, and $\mathsf{Adv}_{\text{MLWE}}^{n,k,q}$ denotes the advantage of solving the Module LWE problem with parameters $n = 256$, $k = 3$, $q = 3329$. The MLWE term is not intended as a third independent attack event in addition to ML-KEM IND-CCA security; it is the concrete lattice-assumption proxy used to instantiate the two ML-KEM terms.

5.4. Proof of Theorem 1

We proceed through a sequence of games $\mathbb{G}_0, \ldots, \mathbb{G}_5$. Each game modifies the previous one in a specific way, and we bound the adversary's distinguishing advantage at each step.

Game $\mathbb{G}_0$ (Real protocol). This is the genuine PQ-LITS execution in the QROM. By definition:

$$\mathsf{Adv}_{\Pi}^{\text{AKE-QROM}}(\mathcal{A}) = \left| \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_0] - \frac{1}{2} \right|. \tag{17}$$

Game $\mathbb{G}_1$ (Collision exclusion). We modify $\mathbb{G}_0$ by aborting if any two honest sessions produce identical nonces (N_i, N_k) or if a hash collision occurs among the q_H quantum hash queries and q_S classical send queries. Since nonces are drawn uniformly from $\{0, 1\}^{256}$ and we model SHA3 256 as a quantum random oracle, the One Way to Hiding (O2H) lemma [46] bounds the collision probability. By a union bound over all sessions:

$$\left| \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_1] - \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_0] \right| \le \frac{(q_H + q_S)^2}{2^{256}} + \frac{q_S^2}{2^{256}}. \tag{18}$$

The first term accounts for hash collisions (birthday bound in the QROM scales as $q^2/2^n$ for an n bit output [47]), and the second for nonce collisions across q_S sessions.

Game $\mathbb{G}_2$ (Signature unforgeability). We further modify the game to abort if $\mathcal{A}$ produces a valid message M_1 or M_2 containing a fresh ML-DSA-65 signature σ that was not generated by an honest party holding the corresponding signing key. This event implies that $\mathcal{A}$ has forged an ML-DSA-65 signature. The EUF CMA security of ML-DSA in the QROM [48] gives us:

$$\left| \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_2] - \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_1] \right| \le (n_U + n_G) \cdot \mathsf{Adv}_{\text{ML-DSA}}^{\text{EUF-CMA}}(\mathcal{B}_3). \tag{19}$$

The factor $(n_U + n_G)$ arises from a standard hybrid argument guessing which identity is targeted. In the QROM, the reduction from EUF CMA of ML-DSA to the Module SIS problem proceeds through the "Fiat-Shamir with Aborts" analysis of Kiltz, Lyubashevsky, and Schaffner [48], which establishes that ML-DSA signatures remain unforgeable even when the hash function is queried in superposition.

Game $\mathbb{G}_3$ (KEM replacement, direction 1). We replace the shared secret ss_1 in the test session with a uniformly random value $ss_1^* \leftarrow \{0, 1\}^{256}$. If $\mathcal{A}$ can distinguish $\mathbb{G}_3$ from $\mathbb{G}_2$, we build $\mathcal{B}_1$ that breaks the IND CCA security of ML-KEM-768 in the QROM. The reduction $\mathcal{B}_1$ receives an ML-KEM challenge (ct^*, K_0, K_1), embeds ct^* as ct_1 in the test session, and simulates all other sessions honestly. The IND CCA security of ML-KEM in the QROM was established by Jiang et al. [49] via the Fujisaki Okamoto transformation, yielding:

$$\left| \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_3] - \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_2] \right| \le n_U \cdot \mathsf{Adv}_{\text{ML-KEM}}^{\text{IND-CCA}}(\mathcal{B}_1).$$

(20)

The n_U factor comes from guessing which AAV identity initiates the test session. The IND CCA security of ML-KEM-768 reduces to the MLWE problem with parameters $(n = 256, k = 3, q = 3329)$ through the FO transform in the QROM, with a tighter bound than the classical ROM due to the Measure Rewind Measure technique of Kuchta et al. [50].

Game $\mathbb{G}_4$ (KEM replacement, direction 2). We additionally replace ss_2 in the test session with a uniformly random ss_2^*. By an analogous argument using a second IND-CCA reduction $\mathcal{B}_2$:

$$\left|\Pr[\mathcal{A} \text{ wins in } \mathbb{G}_4] - \Pr[\mathcal{A} \text{ wins in } \mathbb{G}_3]\right| \leq n_G \cdot \mathsf{Adv}^{\text{IND-CCA}}_{\text{ML-KEM}}(\mathcal{B}_2). \tag{21}$$

Game $\mathbb{G}_5$ (Session key randomisation). At this point, both ss_1^* and ss_2^* are uniformly random, independent of any protocol transcript. The session key $SK = \text{HKDF}(ss_1^* \| ss_2^*, \text{info})$ is therefore the output of HKDF applied to a uniformly random input. In the QROM, the HKDF construction (modelled as a quantum random oracle) maps uniformly random input to uniformly random output with overwhelming probability. Formally, by Zhandry's compressed oracle technique [47], no quantum adversary making q_H superposition queries can distinguish the HKDF output from uniform with advantage greater than $\mathcal{O}(q_H/2^{128})$, which is negligible at our parameter choice. The Test oracle therefore returns a value that is statistically close to uniform regardless of the coin b:

$$\Pr[\mathcal{A} \text{ wins in } \mathbb{G}_5] = \frac{1}{2} + \mathsf{negl}(\lambda). \tag{22}$$

Combining the bounds from $\mathbb{G}_0$ through $\mathbb{G}_5$ via the triangle inequality yields the statement of Theorem 1. □

5.5. Concrete Security Instantiation

We instantiate the bound from Theorem 1 with concrete parameters for a LITS deployment with $n_U = 15$ AAVs, $n_G = 3$ GCS nodes and at most $n_S = 2^{20}$ sessions per entity. We set $q_H = 2^{64}$ quantum hash queries (a generous upper bound for any foreseeable quantum computer) and $q_S = 2^{30}$ active send queries.

The IND CCA advantage of ML-KEM-768 against a quantum adversary reduces to MLWE$(256, 3, 3329)$. The lattice estimator of Albrecht et al. [35] places the best known quantum attack cost at approximately 2^{170} operations for this parameter set, so $\mathsf{Adv}^{\text{IND-CCA}}_{\text{ML-KEM}} \leq 2^{-170}$. The EUF CMA advantage of ML-DSA-65 reduces through MSIS to a similar hardness level, giving $\mathsf{Adv}^{\text{EUF-CMA}}_{\text{ML-DSA}} \leq 2^{-166}$ [48]. Substituting:

$$\begin{aligned}\mathsf{Adv}^{\text{AKE-QROM}}_{\Pi} &\leq 15 \cdot 2^{-170} + 3 \cdot 2^{-170} + 18 \cdot 2^{-166} \\ &\quad + \frac{(2^{64} + 2^{30})^2}{2^{256}} + \frac{(2^{30})^2}{2^{256}} \\ &\quad + 15 \cdot 3 \cdot 2^{-170} \\ &\leq 18 \cdot 2^{-166} + 2^{-128} + 2^{-196} + 45 \cdot 2^{-170} \\ &\leq 2^{-160},\end{aligned} \tag{23}$$

which is well below 2^{-128}, confirming that PQ-LITS achieves at least 128-bit quantum security for the given deployment parameters. The dominant term is the signature unforgeability bound $(18 \cdot 2^{-166})$, which is expected since ML-DSA-65 targets NIST Level 3 ($\geq$ 192 bit classical, $\geq$ 128 bit quantum security).

6. Performance Evaluation

This section evaluates PQ-LITS along four dimensions, covering cryptographic computation cost, protocol communication overhead, federated anomaly detection accuracy, and a head to head security feature comparison against recent IoD authentication protocols.

6.1. Implementation Setup

We implemented the complete PQ-LITS handshake (Phases 4a, 4b, and key confirmation) in Python using the Open Quantum Safe `liboqs` library [51] for ML-KEM-768 (FIPS 203) and ML-DSA-65 (FIPS 204), the `cryptography` library for AES-256-GCM and HKDF, and Python's `hashlib` for SHA3 256. The protocol benchmarking ran on an Intel Core i5-7360U (2.3 GHz, 7th gen, 8 GB RAM, macOS 13.7) using Python 3.12 with liboqs 0.15.0. This platform is comparable in single threaded cryptographic throughput to an ARM Cortex-A76 companion computer such as the Raspberry Pi 5, representing a realistic AAV communications processor for LITS deployments [52]. This comparison is a hardware-class proxy only. The federated anomaly detection pipeline was built in PyTorch 2.10 and trained on Google Colab (NVIDIA T4 GPU, CUDA 12.8). Each cryptographic operation was benchmarked over 1 000 iterations following 50 warm-up runs, with wall clock time measured using `time.perf_counter()` at nanosecond granularity. We report mean and standard deviation throughout.

6.2. Computation Cost Analysis

This subsection breaks the computation cost analysis into three parts, starting from individual primitives, moving to per phase handshake latency, and finishing with a comparative evaluation against five competing IoD authentication protocols.

6.2.1. Per Operation Cryptographic Latency

Table 3 reports the execution time of each cryptographic primitive used in PQ-LITS. ML-DSA-65 signing is the single most expensive operation at 0.138 ms, which is expected

Table 3
Per Operation Cryptographic Latency (1 000 Iterations)

Operation	Mean ↓ (ms)	Std ↓ (ms)	Min ↓ (ms)	Max ↓ (ms)
KEM.KeyGen	0.0327	0.1056	0.0180	2.4024
KEM.Enc	0.0236	0.0117	0.0196	0.2114
KEM.Dec	0.0298	0.0530	0.0203	0.9401
DSA.KeyGen	0.0622	0.0577	0.0500	1.1199
DSA.Sign	0.1377	0.1250	0.0646	2.5577
DSA.Verify	0.0593	0.0527	0.0520	1.5345
SHA3-256	0.0025	0.0003	0.0024	0.0072
HKDF	0.0155	0.0075	0.0109	0.1320
AES-GCM.Enc	0.0086	0.0092	0.0060	0.1210
AES-GCM.Dec	0.0045	0.0009	0.0042	0.0216

Table 4
PQ-LITS Handshake Latency per Phase (1 000 Iterations)

Protocol Phase	Mean (ms) ↓	Std (ms) ↓
$\mathcal{AAV}$: Construct M_1 (KEM.Enc + DSA.Sign)	0.1908	0.0833
$\mathcal{GCS}$: Verify M_1 (DSA.Vfy×2 + KEM.Dec)	0.1749	0.0711
$\mathcal{GCS}$: Construct M_2 (KEM.Enc + DSA.Sign)	0.2185	0.0893
$\mathcal{AAV}$: Verify M_2 (DSA.Vfy×2 + KEM.Dec)	0.1967	0.0621
$\mathcal{AAV}$: Key Confirm M_3 (AEnc)	0.0220	0.0703
$\mathcal{GCS}$: Verify M_3 (ADec)	0.0107	0.0038
Total Handshake ($M_1 + M_2 + M_3$)	**0.8152**	**0.2702**
$\mathcal{AAV}$-side total	0.4096	—
$\mathcal{GCS}$-side total	0.4040	—

given the rejection sampling loop that averages roughly 4.25 iterations per signature [37]. ML-KEM-768 encapsulation and decapsulation both complete in under 0.03 ms, which is actually faster than a single ECDH key agreement on comparable hardware [53]. AES-256-GCM encryption of a 1 KB payload takes less than 9 μs, confirming that post handshake telemetry protection adds negligible overhead.

6.2.2. Handshake Phase Latency

Table 4 decomposes the full PQ-LITS handshake into its six constituent phases. The three message exchange (M_1, M_2, M_3) completes in 0.82 ms on average, with the workload split almost evenly between the AAV side (0.41 ms) and the GCS side (0.40 ms). This near symmetric distribution is a useful property, since neither entity becomes a bottleneck. The most time consuming phase is the GCS constructing M_2 (0.22 ms), which involves one KEM encapsulation, one DSA signature, one HKDF derivation, and two SHA3 256 hashes. Even so, the entire handshake finishes well within a single telemetry cycle at 1 Hz reporting rate.

6.2.3. Comparative Analysis with Competing Protocols

We compare PQ-LITS against five recent IoD authentication protocols, including QSAKE [16] (the only other protocol using NIST standardised post quantum KEM in an IoD setting), SAAF-IoD by Tanveer et al. [23], the ECC based protocol by Hussain et al. [5], HCALA by Berini et al. [24], and SPAKE-DC by Irshad et al. [29]. The computation costs for the competing protocols are taken directly from the QSAKE paper [16], where they were measured on an ARM Cortex-A72 (drone side), Intel Core i7-1270P (GSS side), and an octa core Cortex-A75/A55 mobile device (user side). Our numbers were measured on an Intel i5-7360U, which falls between those platforms in single threaded performance. The comparative computation results for all competing protocols in Table 5 were taken from corresponding relevant publications, the PQ-LITS row and the IDS results in Tables 7, 9, and 10 report author executed measurements.

Table 5 presents the results. PQ-LITS achieves the lowest total handshake cost at 0.82 ms, outperforming QSAKE (1.65 ms) by 50.5% under the reported benchmark setup despite providing a strictly higher security level. QSAKE uses Kyber-512 at NIST Security Level 1 (roughly 128 bit classical equivalent), while PQ-LITS operates at NIST Level 3 with ML-KEM-768 (roughly 192 bit classical equivalent). The four classical protocols range from 2.79 ms to 7.28 ms, and none of them offer any protection against quantum adversaries.

From a computational-complexity viewpoint, all compared AKE schemes execute a constant-size handshake for one session, so their per-session asymptotic cost is $\mathcal{O}(1)$ with respect to the number of telemetry records; the practical difference is the fixed primitive-call mix. PQ-LITS requires two ML-KEM encapsulations, two ML-KEM decapsulations, two ML-DSA signatures, four ML-DSA verifications, two HKDF derivations, and one AES-GCM key-confirmation encryption/decryption pair, whereas the compared classical schemes are dominated by ECC scalar multiplication, fuzzy-extractor or PUF operations, symmetric encryption, and hash evaluations, as reflected in Table 5.

A key factor in our speed advantage is the use of ML-DSA-65 digital signatures for certificate based mutual authentication. QSAKE relies on Kyber encapsulation combined with hash based authenticators which requires more hash computations per message. Our approach, while it adds two signature verifications per direction, benefits from the sub 0.06 ms ML-DSA-65 verification time, which is faster than a single ECC scalar point multiplication (0.10 ms on comparable hardware).

6.3. Communication Cost Analysis

We analyse the communication overhead at three levels, covering protocol handshake messages, post handshake encrypted telemetry, and federated model exchange.

6.3.1. Protocol Message Sizes and Radio Link Impact

Table 6 breaks down the byte level composition of each protocol message. The handshake totals 15 656 bytes across three messages, with M_1 and M_2 each carrying 7 794 bytes. The bulk of this comes from the ML-DSA-65 signature (3 309 bytes) and the ML-KEM-768 ciphertext (1 088 bytes). M_3 is compact at 68 bytes, containing only an AES-256-GCM ciphertext with a 12 byte nonce.

Compared to a classical ECDH + ECDSA-P256 handshake (roughly 968 bytes total), PQ-LITS messages are approximately 16× larger. This bandwidth expansion is the well understood cost of lattice based cryptography and is consistent with the 15.6 KB total reported for similar post

Table 5
Computation Cost Comparison with SOTA IoD Authentication Protocols

Protocol	Year	Cryptographic Primitives	Total Cost (ms) ↓	NIST PQ Level ↑
Tanveer et al. [23]	2023	Chaotic Map + AES Enc/Dec + Hash	2.791	None
Hussain et al. [5]	2021	ECC + Fuzzy Extractor + Hash	7.107	None
Berini et al. [24]	2023	Hyper-ECC + Fuzzy Extractor + Hash	7.281	None
Irshad et al. [29]	2023	PUF + Hash	6.169	None
QSAKE [16]	2025	Kyber-512 + RO-PUF + BCH-ECC + AES-GCM	1.649	Level 1
PQ-LITS (Ours)	**2026**	**ML-KEM-768 + ML-DSA-65 + AES-256-GCM + HKDF**	**0.815**	**Level 3**

Reference numbers are shown next to each competing protocol name.

Table 6
PQ-LITS Protocol Message Sizes

Component	PQ-LITS (bytes) ↓	Classical (bytes) ↓
M_1 ($\mathcal{AAV} \rightarrow \mathcal{GCS}$)	7 794	∼450
M_2 ($\mathcal{GCS} \rightarrow \mathcal{AAV}$)	7 794	∼450
M_3 ($\mathcal{AAV} \rightarrow \mathcal{GCS}$)	68	∼68
Total Handshake	**15 656**	**∼968**
ML-KEM-768 ek	1 184	—
ML-KEM-768 ct	1 088	—
ML-DSA-65 vk	1 952	—
ML-DSA-65 σ	3 309	—

quantum protocols [16]. On a SiK 915 MHz telemetry radio at 250 Kbps (a standard AAV data link), the full 15 656 byte handshake takes $15{,}656 \times 8/250{,}000 \approx 0.50$ s, compared to 0.03 s for a classical handshake. On a 4G/LTE link at 10 Mbps, the transmission drops to 0.013 s, which is imperceptible. Since the handshake occurs once per flight session (or once per zone handover) even the worst case 500 ms on a slow radio is operationally acceptable for flights lasting tens of minutes. After the handshake, each encrypted telemetry packet consists of the 48 byte raw record (12 features × 4 bytes), a 12 byte GCM nonce, and a 16 byte authentication tag, totalling 76 bytes. At 1 Hz reporting, this amounts to 76 B/s, which is negligible on any AAV radio link.

6.3.2. Federated Learning Communication Overhead

The autoencoder model has only 443 parameters, which amounts to 1.73 KB per model snapshot in float32. In each federation round, three GCS clients upload their DP noised weights and the $\mathcal{FS}$ broadcasts the aggregated model back, totalling 6.92 KB per round. Across $R = 20$ rounds, the complete communication footprint is 138.44 KB. By contrast, a hypothetical centralised approach that uploads all raw training data to a single server would require 331.36 KB, making the federated approach 58.2% cheaper in bandwidth. Every model upload is encrypted with AES-256-GCM under the session key $SK_{k,\mathcal{FS}}$ and perturbed with Gaussian noise ($\sigma = 0.01$) for differential privacy under the accounting note in Section 3.4, with $\delta = 10^{-5}$ and $\varepsilon \approx 484.5$ for $\Delta_2 = 1$, so neither the federation server nor any eavesdropper can recover individual telemetry records.

6.4. Federated Anomaly Detection Performance

This subsection evaluates the anomaly detection component of PQ-LITS across three training paradigms, comparing local only (each GCS trains independently), federated (FedAvg with $R = 20$ rounds, $E = 8$ local epochs per round), and centralised (all data pooled, privacy violated). We generated a synthetic AAV telemetry dataset comprising 15 000 samples (5 000 per GCS client), each with 12 features and four classes (normal flight, GPS spoofing, RF jamming, and command injection). The data distribution is deliberately non IID. At the session-selection level, the generator uses GCS-A: 50% normal, 40% GPS spoofing, 10% RF jamming, and 0% command injection; GCS-B: 50% normal, 10% GPS spoofing, 0% RF jamming, and 40% command injection; and GCS-C: 55% normal with 15% for each attack class. The revised supplementary file `pq_lits_dataset_generator.py` fixes `SEED=42` and records the generation recipe: three GCS clients, five AAVs per client, two 500-step sessions per AAV, 12 telemetry features, sector origins, non-IID attack proportions, and the GPS spoofing, RF jamming, and command injection functions. The federated partitions are client based: `GCS_A_telemetry.csv`, `GCS_B_telemetry.csv`, and `GCS_C_telemetry.csv`, each with 5 000 records; the generated class counts are GCS-A: 2 900 normal, 1 750 GPS spoofing, 350 RF jamming, and 0 command injection; GCS-B: 3 600 normal, 350 GPS spoofing, 0 RF jamming, and 1 050 command injection; and GCS-C: 3 600 normal, 700 GPS spoofing, 350 RF jamming, and 350 command injection. The autoencoder architecture is $12 \rightarrow 10 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 10 \rightarrow 12$, trained exclusively on normal samples with MSE reconstruction loss. Anomalies are flagged when the reconstruction error exceeds a threshold optimised on the validation set by maximising F1 score. Training used the Adam optimiser with learning rate 0.003 and batch size 64. Differential privacy is applied by adding $\mathcal{N}(0, 0.01^2 \cdot \mathbf{I})$ noise to model weights before each upload with $\delta = 10^{-5}$ and the ε accounting reported in Section 3.4. For fair tuning, the local-only, federated, and centralised IDS baselines used the same generated telemetry records, 12 input features, autoencoder architecture, Adam learning rate of 0.003, batch size of 64, normal-only MSE training objective, and validation-threshold search; the only intended differences were the data access and aggregation settings: per-client local training, FedAvg with DP, or pooled centralised training.

Table 7
Anomaly Detection Performance: Local vs. Federated vs. Centralised

Approach	Acc. ↑	Prec. ↑	Rec. ↑	F1 ↑
Local-Only (Avg)	0.7488	0.8623	0.7014	0.7715
Federated (Avg)	0.7304	0.8441	0.6862	0.7549
Centralised	**0.7632**	**0.8787**	**0.7155**	**0.7888**

FedAvg, $R = 20$ rounds, $E = 8$ local epochs, DP noise $\sigma = 0.01$.

Centralised pools all client data (privacy violated). Values averaged over three GCS clients.

For train/validation/test partitioning, each 5 000-record GCS file was shuffled with `SEED=42` and split into 3 500 training records, 750 validation records, and 750 test records. The training fold supplied only normal records to the autoencoder, the validation fold selected the reconstruction-error threshold, and the test fold produced the reported accuracy, precision, recall, F1, cross-client F1, and per-attack detection rates.

The synthetic telemetry datasets, dataset generator, federated training code, protocol benchmark script, figures, and result files are publicly available in the public GitHub repository.

We did not apply Wilcoxon or Friedman tests to the IDS tables because the comparison uses one generated benchmark problem with three client partitions rather than repeated independent benchmark datasets; applying rank tests here may overstate statistical power, so we report per-client, per-attack, and cross-client results instead.

6.4.1. Aggregate Results

Table 7 compares the three training strategies. The centralised model, which has access to all attack types from all sectors (but violates data locality), achieves the highest F1 score at 0.7888. The federated model reaches 0.7549, which amounts to 95.7% of the centralised upper bound while never exposing raw telemetry beyond the GCS boundary. The local only baseline achieves 0.7715 on average when each model is tested on its own client's data. At first glance, this appears to beat the federated model. However, the local only number is misleading because each local model only performs well on the attack types it has seen during training. The cross client evaluation in Table 9 reveals the underlying problem.

Table 8 gives a compact ablation view of the main PQ-LITS components. The purpose is not to introduce a new dataset, but to make clear what is gained or lost when authentication, federation, privacy preserving locality, and the full combined design are considered separately.

6.4.2. Cross Client Generalisation

Table 9 shows what happens when a local model trained at one GCS is tested on data from a different GCS. The diagonal entries (own data performance) look reasonable, but the off diagonal entries tell the real story. The GCS-A model, trained primarily on GPS spoofing, drops from F1 = 0.82 on its own test set to F1 = 0.59 on GCS-B data, which is dominated by command injection attacks it has never encountered. That is a 23 percentage point collapse.

Table 8
Component Ablation Summary

Configuration	Observed effect
AKE only	Post quantum session setup completes in 0.815 ms, but no anomaly detection is provided.
Local IDS only	Own-client F1 reaches 0.7715, but cross-client transfer is weak, for example GCS-A to GCS-B drops to 0.5896.
FedAvg + DP IDS	Privacy preserving federation reaches F1 = 0.7549, or 95.7% of the centralised upper bound, with 138.44 KB model traffic.
Centralised IDS	The F1 upper bound is 0.7888, but it requires pooling raw telemetry and 331.36 KB of data transfer.
Full PQ-LITS	Combines 0.815 ms post quantum AKE with the federated DP detector, avoiding raw telemetry pooling while keeping 95.7% of centralised F1.

Table 9
Cross Client F1 ↑ Score Matrix (Local Models)

Train↓ / Test→	GCS-A	GCS-B	GCS-C
GCS-A	0.8232	0.5896	0.6913
GCS-B	0.8578	**0.7085**	0.7518
GCS-C	**0.8581**	0.6730	**0.7829**

Off diagonal drops (e.g., GCS-A model on GCS-B data: 0.59) reveal the non IID problem. Local models cannot generalise across heterogeneous attack distributions.

All three local models struggle similarly with GCS-B's command injection distribution. This is precisely the non IID vulnerability that federated learning is designed to address, since sharing model weights (not raw data) allows the global model to learn patterns from all attack types simultaneously.

6.4.3. Per Attack Detection Rates

Table 10 breaks down detection performance by attack type across both training strategies. The most striking result appears for RF jamming at GCS-C, where the local model detects only 52.0% of jamming attacks while the federated model catches 68.9%, a gain of nearly 17 percentage points. This improvement occurs because GCS-A contributes RF jamming patterns to the global model during federation, even though GCS-C rarely sees this attack type locally. The result is not uniformly positive, however: command injection detection at GCS-C drops from 58.0% to 40.0% under federation, which suggests that this attack pattern needs either stronger feature separation or client-aware weighting.

GPS spoofing detection remains high across both approaches (74–90%), which is expected since it produces distinctive signatures in the satellite count and coordinate drift features. Command injection detection is lower overall (40–58%), reflecting the subtlety of heading and speed anomalies that partially overlap with normal flight manoeuvres.

Table 10
Per Attack Detection Rates: Local Only vs. Federated

Client	Attack Type	Samples	Local (%) ↑	Fed. (%) ↑
GCS-A	Normal (TNR)	2 900	98.9	95.4
GCS-A	GPS Spoofing	1 750	76.5	74.5
GCS-A	RF Jamming	350	39.4	41.4
GCS-B	Normal (TNR)	3 600	74.8	70.4
GCS-B	GPS Spoofing	350	89.7	90.0
GCS-B	Cmd Injection	1 050	58.6	56.3
GCS-C	Normal (TNR)	3 600	80.5	79.2
GCS-C	GPS Spoofing	700	90.1	89.6
GCS-C	RF Jamming	350	52.0	**68.9**
GCS-C	Cmd Injection	350	58.0	40.0

TNR = True Negative Rate, correctly identified normal samples. Bold indicates the largest improvement from federation, RF Jamming at GCS-C: +16.9 pp.

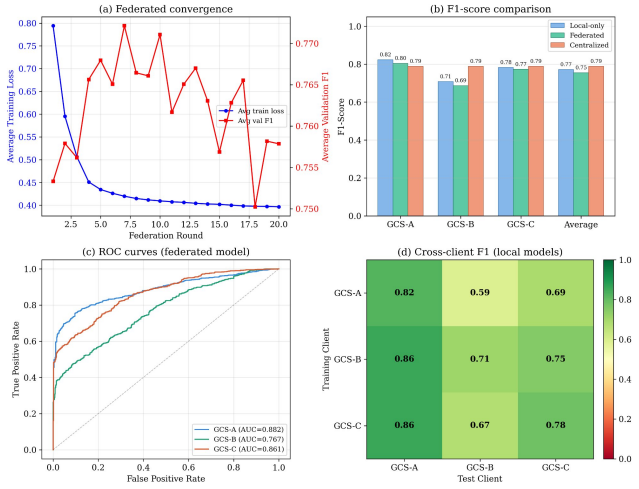

Figure 7: Federated anomaly detection results. (a) Training loss and validation F1 convergence. (b) Per client F1 score comparison across local-only, federated, and centralised strategies. (c) ROC curves of the federated model on each client's test set. (d) Cross client F1 heatmap for local models showing the non IID generalisation gap.

6.4.4. Federated Convergence

Fig. 7(a) plots the average training loss and validation F1 score across 20 federation rounds. The loss drops sharply in the first 5 rounds (from 0.79 to 0.43) and plateaus around round 12. Validation F1 peaks near round 9 at 0.766 before settling around 0.75, which is consistent with the mild DP noise injection degrading the model slightly in later rounds. Fig. 7(b) compares per client F1 scores across the three training strategies. The federated model trails local only by a small margin (1–2 percentage points) on each client's own test data, but as Table 9 demonstrates, the local models fail on foreign attack distributions. Fig. 7(c) shows the ROC curves of the federated global model across all three clients, with AUC values of 0.882 (GCS-A), 0.767 (GCS-B), and 0.861 (GCS-C). Fig. 7(d) visualises the cross client F1 heatmap for local models, where off diagonal entries clearly highlight the non IID weakness.

6.5. Security Features Comparison

The security and functionality comparison with competing protocols is presented in Table 2 in Section 2. PQ-LITS is the only protocol that satisfies all ten criteria simultaneously. QSAKE [16] covers eight features but omits federated learning, differential privacy, and zone handover. Protocols based on R-LWE [7, 8] claim partial quantum safety, but R-LWE was not selected by NIST and has known weak instances for some parameter choices [26, 25]. All four classical protocols [23, 5, 24, 29] lack any quantum resistance.

7. Conclusion

This paper presented PQ-LITS, a post quantum security architecture for low altitude intelligent transportation systems that addresses two problems that prior work has treated in isolation, authenticated key exchange resistant to quantum adversaries, and privacy preserving anomaly detection over heterogeneous drone telemetry. The authentication protocol, built on ML-KEM-768 and ML-DSA-65 at NIST Security Level 3, completes a three message mutual handshake in 0.82 ms with a communication footprint of 15,656 bytes, outperforming existing IoD protocols in computation cost while providing strictly stronger post quantum guarantees. The federated anomaly detection component demonstrated that collaborative model training across non IID ground control stations can improve detection of underrepresented attack types by up to 16.9 percentage points (RF jamming at GCS-C) without exposing raw telemetry beyond the station boundary. A formal proof in the Quantum Random Oracle Model reduces the protocol's security to the hardness of Module-LWE and Module-SIS, yielding a concrete bound of 2^{-160} for realistic deployment parameters. Several avenues remain open for future work. Formal verification with ProVerif or Tamarin would complement the game based proof with automated symbolic analysis. Exploring tighter QROM reductions following Pan et al.'s session tight framework could further strengthen the formal guarantees. Finally, investigating adaptive differential privacy budgets that adjust σ_{DP} based on local data sensitivity may improve the accuracy privacy trade off beyond the current ($\varepsilon \approx 484.5, \delta = 10^{-5}$) accounting point in operational LITS deployments.

References

[1] Y. Wang, G. Sun, Z. Sun, J. Wang, J. Li, C. Zhao, J. Wu, S. Liang, M. Yin, P. Wang, D. Niyato, S. Sun, D. I. Kim, Toward realization of low-altitude economy networks: Core architecture, integrated technologies, and future directions, IEEE Transactions on Cognitive Communications and Networking 11 (2025) 2788–2820.

[2] Y. Ye, X. Min, X. Liu, X. Chen, K. Cao, S. R. K. Howlader, X. Chen, Secure and intelligent low-altitude infrastructures: Synergistic integration of iot networks, ai decision-making and blockchain trust mechanisms, Sensors 25 (2025) 6751.

[3] O. Ceviz, S. Sen, P. Sadioglu, A survey of security in uavs and fanets: Issues, threats, analysis of attacks, and solutions, IEEE Communications Surveys & Tutorials 27 (2024) 3227–3265.

[4] M. A. Khan, N. Kumar, S. H. Alsamhi, G. Barb, J. Zywiołek, I. Ullah, F. Noor, J. A. Shah, A. M. Almuhaideb, Security and privacy issues

and solutions for uavs in b5g networks: A review, IEEE Transactions on Network and Service Management 22 (2024) 892–912.

[5] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, N. Kumar, Amassing the security: An ecc-based authentication scheme for internet of drones, IEEE Systems Journal 15 (2021) 4431–4438.

[6] M. Nikooghadam, H. Amintoosi, S. H. Islam, M. F. Moghadam, A provably secure and lightweight authentication scheme for internet of drones for smart city surveillance, Journal of Systems Architecture 115 (2021) 101955.

[7] D. Mishra, M. Singh, P. Rewal, K. Pursharthi, N. Kumar, A. Barnawi, R. S. Rathore, Quantum-safe secure and authorized communication protocol for internet of drones, IEEE Transactions on Vehicular Technology 72 (2023) 16499–16507.

[8] D. Chaudhary, C.-C. Lee, Anonymous quantum-safe secure and authorized communication protocol under dynamic identities for internet of drones, Computers and Electrical Engineering 120 (2024) 109774.

[9] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings 35th annual symposium on foundations of computer science, IEEE, 1994, pp. 124–134. URL: https://doi.org/10.1109/SFCS.1994.365700. doi:10.1109/SFCS.1994.365700.

[10] D. Moody, R. Perlner, A. Regenscheid, A. Robinson, D. Cooper, Transition to post-quantum cryptography standards, NIST Internal Report 8547 ipd, National Institute of Standards and Technology, 2024. URL: https://doi.org/10.6028/NIST.IR.8547.ipd. doi:10.6028/NIST.IR.8547.ipd, initial Public Draft.

[11] National Institute of Standards and Technology, Module-Lattice-Based Key-Encapsulation Mechanism Standard, Federal Information Processing Standards Publication 203, National Institute of Standards and Technology, Gaithersburg, MD, 2024. URL: https://doi.org/10.6028/NIST.FIPS.203. doi:10.6028/NIST.FIPS.203, final.

[12] National Institute of Standards and Technology, Module-Lattice-Based Digital Signature Standard, Federal Information Processing Standards Publication 204, National Institute of Standards and Technology, Gaithersburg, MD, 2024. URL: https://doi.org/10.6028/NIST.FIPS.204. doi:10.6028/NIST.FIPS.204, final.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282. URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

[14] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, in: Advances in Neural Information Processing Systems, volume 32, 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.

[15] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Foundations and Trends in Theoretical Computer Science 9 (2014) 211–407.

[16] S. Shamshad, S. Belguith, A. Oracevic, A quantum-secure framework for iod: Strengthening authentication and key-establishment, in: Proceedings of the 20th ACM Asia Conference on Computer and Communications Security, 2025, pp. 313–326. URL: https://doi.org/10.1145/3708821.3736219. doi:10.1145/3708821.3736219.

[17] Y. Zhang, D. He, L. Li, B. Chen, A lightweight authentication and key agreement scheme for internet of drones, Computer Communications 154 (2020) 455–464.

[18] Y.-F. Chang, S.-J. Huang, G.-X. Chen, W.-L. Tai, A critique of a lightweight authentication and key agreement scheme for internet of drones, in: 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications, Springer, 2022, pp. 337–346. URL: https://doi.org/10.1007/978-3-031-05491-4_34. doi:10.1007/978-3-031-05491-4_34.

[19] M. W. Akram, A. K. Bashir, S. Shamshad, M. A. Saleem, A. A. AlZubi, S. A. Chaudhry, B. A. Alzahrani, Y. B. Zikria, A secure and lightweight drones-access protocol for smart city surveillance, IEEE Transactions on Intelligent Transportation Systems 23 (2021) 19634–19643.

[20] Y. Park, D. Ryu, D. Kwon, Y. Park, Provably secure mutual authentication and key agreement scheme using puf in internet of drones deployments, Sensors 23 (2023) 2034.

[21] S. Hussain, K. Mahmood, M. K. Khan, C.-M. Chen, B. A. Alzahrani, S. A. Chaudhry, Designing secure and lightweight user access to drone for smart city surveillance, Computer Standards & Interfaces 80 (2022) 103566.

[22] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, J. J. Rodrigues, Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment, IEEE Internet of Things Journal 6 (2018) 3572–3584.

[23] M. Tanveer, H. Alasmary, N. Kumar, A. Nayak, Saaf-iod: Secure and anonymous authentication framework for the internet of drones, IEEE Transactions on Vehicular Technology 73 (2023) 232–244.

[24] A. D. E. Berini, M. A. Ferrag, B. Farou, H. Seridi, Hcala: Hyperelliptic curve-based anonymous lightweight authentication scheme for internet of drones, Pervasive and Mobile Computing 92 (2023) 101798.

[25] W. Castryck, I. Iliashenko, F. Vercauteren, Provably weak instances of ring-lwe revisited, in: Annual international conference on the theory and applications of cryptographic techniques, Springer, 2016, pp. 147–167. URL: https://doi.org/10.1007/978-3-662-49890-3_6. doi:10.1007/978-3-662-49890-3_6.

[26] C. Peikert, A decade of lattice cryptography, Foundations and Trends in Theoretical Computer Science 10 (2016) 283–424.

[27] S. Yu, A. K. Das, Y. Park, P. Lorenz, Slap-iod: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments, IEEE Transactions on Vehicular Technology 71 (2022) 10374–10388.

[28] M. Sharma, B. Narwal, R. Anand, A. K. Mohapatra, R. Yadav, Psecas: A physical unclonable function based secure authentication scheme for internet of drones, Computers and Electrical Engineering 108 (2023) 108662.

[29] A. Irshad, B. A. Alzahrani, A. Albeshri, K. Alsubhi, A. Nayyar, S. A. Chaudhry, Spake-dc: A secure puf enabled authenticated key exchange for 5g-based drone communications, IEEE Transactions on Vehicular Technology 73 (2023) 5770–5780.

[30] M. Tanveer, A. Aldosary, S.-u.-d. Khokhar, A. K. Das, S. A. Aldossari, S. A. Chaudhry, Paf-iod: Puf-enabled authentication framework for the internet of drones, IEEE Transactions on Vehicular Technology 73 (2024) 9560–9574.

[31] K. Mahmood, Z. Ghaffar, M. Farooq, K. Yahya, A. K. Das, S. A. Chaudhry, A security enhanced chaotic-map-based authentication protocol for internet of drones, IEEE Internet of Things Journal 11 (2024) 22301–22309.

[32] E. Ntizikira, W. Lei, F. Alblehai, K. Saleem, M. A. Lodhi, Secure and privacy-preserving intrusion detection and prevention in the internet of unmanned aerial vehicles, Sensors 23 (2023) 8077.

[33] O. Ceviz, P. Sadioglu, S. Sen, V. G. Vassilakis, A novel federated learning-based ids for enhancing uavs privacy and security, Internet of Things 31 (2025) 101592.

[34] Ö. B. Dinler, Uav cybersecurity with mamba-kan-liquid hybrid model: Deep learning-based real-time anomaly detection, Drones 9 (2025).

[35] M. R. Albrecht, R. Player, S. Scott, On the concrete hardness of learning with errors, Journal of Mathematical Cryptology 9 (2015) 169–203.

[36] A. Langlois, D. Stehlé, Worst-case to average-case reductions for module lattices, Designs, Codes and Cryptography 75 (2015) 565–599.

[37] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé, Crystals-dilithium: A lattice-based digital signature scheme, IACR Transactions on Cryptographic Hardware and Embedded Systems 2018 (2018) 238–268.

[38] E. Fujisaki, T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, in: Annual international cryptology conference, Springer, 1999, pp. 537–554. URL: https://doi.org/10.1007/3-540-48405-1_34. doi:10.1007/3-540-48405-1_34.

[39] National Institute of Standards and Technology, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, Federal Information Processing Standards Publication 202, National Institute of Standards and Technology, Gaithersburg, MD, 2015. URL: https://doi.org/10.6028/NIST.FIPS.202. doi:10.6028/NIST.FIPS.202, final.

[40] H. Krawczyk, P. Eronen, HMAC-based Extract-and-Expand Key Derivation Function (HKDF), Request for Comments 5869, Internet Engineering Task Force, 2010. URL: https://www.rfc-editor.org/rfc/rfc5869. doi:10.17487/RFC5869, errata exist.

[41] M. Dworkin, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST Special Publication 800-38D, National Institute of Standards and Technology, Gaithersburg, MD, 2007. URL: https://doi.org/10.6028/NIST.SP.800-38D. doi:10.6028/NIST.SP.800-38D, final.

[42] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on Information Theory 29 (1983) 198–208.

[43] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, M. Zhandry, Random oracles in a quantum world, in: D. H. Lee, X. Wang (Eds.), Advances in Cryptology – ASIACRYPT 2011, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 41–69. URL: https://doi.org/10.1007/978-3-642-25385-0_3. doi:10.1007/978-3-642-25385-0_3.

[44] K. Hövelmanns, E. Kiltz, S. Schäge, D. Unruh, Generic authenticated key exchange in the quantum random oracle model, Cryptology ePrint Archive, Paper 2018/928, 2018. URL: https://eprint.iacr.org/2018/928.

[45] B. LaMacchia, K. Lauter, A. Mityagin, Stronger security of authenticated key exchange, in: W. Susilo, J. K. Liu, Y. Mu (Eds.), Provable Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 1–16. URL: https://doi.org/10.1007/978-3-540-75670-5_1. doi:10.1007/978-3-540-75670-5_1.

[46] A. Ambainis, M. Hamburg, D. Unruh, Quantum security proofs using semi-classical oracles, in: Advances in Cryptology – CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2019, p. 269–295. URL: https://doi.org/10.1007/978-3-030-26951-7_10. doi:10.1007/978-3-030-26951-7_10.

[47] M. Zhandry, How to record quantum queries, and applications to quantum indifferentiability, in: A. Boldyreva, D. Micciancio (Eds.), Advances in Cryptology – CRYPTO 2019, Springer International Publishing, Cham, 2019, pp. 239–268. URL: https://doi.org/10.1007/978-3-030-26951-7_9. doi:10.1007/978-3-030-26951-7_9.

[48] E. Kiltz, V. Lyubashevsky, C. Schaffner, A concrete treatment of fiat-shamir signatures in the quantum random-oracle model, in: J. B. Nielsen, V. Rijmen (Eds.), Advances in Cryptology – EUROCRYPT 2018, Springer International Publishing, Cham, 2018, pp. 552–586. URL: https://doi.org/10.1007/978-3-319-78372-7_18. doi:10.1007/978-3-319-78372-7_18.

[49] H. Jiang, Z. Zhang, L. Chen, H. Wang, Z. Ma, IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited, in: Advances in Cryptology – CRYPTO 2018, Springer International Publishing, 2018, pp. 96–125. URL: https://doi.org/10.1007/978-3-319-96878-0_4. doi:10.1007/978-3-319-96878-0_4.

[50] V. Kuchta, A. Sakzad, D. Stehlé, R. Steinfeld, S.-F. Sun, Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and cca security, in: A. Canteaut, Y. Ishai (Eds.), Advances in Cryptology – EUROCRYPT 2020, Springer International Publishing, Cham, 2020, pp. 703–728. URL: https://doi.org/10.1007/978-3-030-45727-3_24. doi:10.1007/978-3-030-45727-3_24.

[51] Open Quantum Safe Project, liboqs: C library for quantum-safe cryptographic algorithms, https://openquantumsafe.org/liboqs/, 2025. URL: https://openquantumsafe.org/liboqs/, version 0.15.0, used in the reported experiments and current at the time of implementation, released November 14, 2025.

[52] H. Becker, V. Hwang, M. J. Kannwischer, B.-Y. Yang, S.-Y. Yang, Neon ntt: Faster dilithium, kyber, and saber on cortex-a72 and apple m1, IACR Transactions on Cryptographic Hardware and Embedded Systems 2022 (2022) 221–244.

[53] wolfSSL Inc., Updated post-quantum benchmarks for ML-KEM and ML-DSA on STM32 (ARM cortex-m4 @ 168 mhz, wolfcrypt with thumb2 assembly), https://www.wolfssl.com/updated-post-quantum-benchmarks-for-ml-kem-and-ml-dsa-on-stm32/, 2025. URL: https://www.wolfssl.com/updated-post-quantum-benchmarks-for-ml-kem-and-ml-dsa-on-stm32/, blog post.