

Article

Secure and Energy-Aware Cryptographic Framework for IoT-Enabled UAV Systems

Dauriya Zhaxygulova ^{1,*} , Maksim Iavich ² , Saule Rakhmetullina ³  and Kuanysh Alipbayev ⁴ 

¹ International School of Engineering, D. Serikbayev East Kazakhstan Technical University, Oskemen 070000, Kazakhstan

² School of Technologies, Caucasus University, Tbilisi 0102, Georgia; miavich@cu.edu.ge

³ School of Digital Technologies and Artificial Intelligence, D. Serikbayev East Kazakhstan Technical University, Oskemen 070000, Kazakhstan; rakhmetullinas@mail.ru

⁴ Department of Aerospace and Electronic Engineering, Almaty University of Power Engineering and Telecommunications Named after G. Daukeyev, Almaty 050013, Kazakhstan; k.alipbayev@aes.kz

* Correspondence: dauriya.dzh@gmail.com

Abstract

The rapid convergence of the Internet of Things (IoT), quantum computing, and artificial intelligence (AI) has amplified the urgency for lightweight yet resilient data protection mechanisms, particularly within unmanned aerial vehicles (UAV). Traditional cryptographic approaches, while mathematically secure, often fail to reconcile the competing requirements of robustness, computational efficiency, and energy sustainability when deployed on resource-constrained platforms such as drones. To address this gap, this paper proposes a novel hybrid lightweight cryptographic model that strategically integrates symmetric and asymmetric primitives in a dual-layer design. The model leverages the efficiency of lightweight authenticated encryption for high-throughput data protection, while incorporating elliptic-curve and lattice-based key exchange mechanisms to ensure both forward secrecy and post-quantum resilience. Experimental evaluation demonstrates that the proposed scheme achieves superior performance compared to conventional methods, offering reduced computational overhead, lower energy consumption, and enhanced resistance to cyber threats. Crucially, the model maintains high levels of confidentiality, integrity, and authenticity while extending operational endurance, making it particularly well-suited for next-generation UAV operating within the broader IoT ecosystem.

Keywords: IoT security; hybrid encryption; lightweight cryptographic algorithms; UAV security; data integrity; energy-efficient encryption



Academic Editor: Alice Miller

Received: 8 October 2025

Revised: 9 November 2025

Accepted: 13 November 2025

Published: 17 November 2025

Citation: Zhaxygulova, D.; Iavich, M.; Rakhmetullina, S.; Alipbayev, K. Secure and Energy-Aware Cryptographic Framework for IoT-Enabled UAV Systems. *Symmetry* **2025**, *17*, 1987. <https://doi.org/10.3390/sym17111987>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) has emerged as a transformative technological paradigm, enabling autonomous communication and data exchange among interconnected devices across various domains such as healthcare, agriculture, logistics, and defense. Among IoT-enabled entities, Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, represent one of the most dynamic and rapidly evolving technologies. Their integration with IoT infrastructures has led to the emergence of the Internet of Flying Things (IoFT) or the Internet of Drones (IoD)—a paradigm that extends IoT connectivity into the aerial domain [1]. UAVs equipped with high-resolution sensors, computer vision modules, and advanced communication systems provide critical advantages in coverage, mobility, and accessibility, enabling data collection from environments otherwise inaccessible to

humans. These features make UAVs indispensable for precision agriculture, environmental monitoring, logistics, construction, and defense operations [2–6].

Cybersecurity remains one of the most critical challenges in UAV ecosystems. The interconnection of drones through IoT-based infrastructures exposes them to a wide range of attacks, including data interception, spoofing, hijacking, and denial of service (DoS) [7–10]. Comprehensive taxonomies of IoD security threats reveal vulnerabilities across hardware, software, communication, and sensor levels, where even minor protocol weaknesses can lead to complete mission compromise [7,9]. The lack of efficient encryption and authentication mechanisms further exacerbates the risks of privacy breaches and control manipulation. Although recent approaches employ Blockchain, Software-Defined Networking (SDN), Machine Learning (ML), and Fog Computing to enhance secure data transmission, these solutions often introduce additional latency and computational overhead, limiting their practical use in resource-constrained UAVs [10]. Therefore, developing hybrid, lightweight cryptographic mechanisms capable of ensuring confidentiality, integrity, and authentication under constrained conditions is crucial for maintaining mission resilience and data trustworthiness.

In addition to security, energy consumption represents another major limitation in UAV deployment. Since UAVs are powered by batteries with limited capacity, flight duration and communication performance are directly linked to energy efficiency [11–14]. Studies have shown that propulsion, payload, and onboard computation constitute the largest energy drains during flight [11]. Recent advances in aerodynamically driven solar-tracking systems have contributed to improving UAV energy autonomy by enhancing the efficiency of solar power utilization during long-duration missions [12]. Optimized path planning and power-aware routing significantly improve mission endurance; for instance, the use of genetic algorithms for coverage optimization can reduce total energy consumption by up to five times compared to greedy methods [13]. Furthermore, modeling and optimizing the energy required for various flight phases—takeoff, hovering, and landing—enable more effective management of onboard resources, leading to extended mission duration and safer landings [14]. Consequently, cryptographic operations integrated into UAV communication systems must be designed with minimal power consumption to avoid compromising overall mission longevity.

Finally, UAVs face substantial computational and autonomy challenges that directly affect their capacity to process encrypted communication and execute onboard decision-making tasks. Due to their lightweight embedded systems, most UAVs possess limited processing and memory resources, making the implementation of complex cryptographic algorithms impractical [15]. Moreover, as UAVs transition toward greater autonomy, they must simultaneously perform obstacle avoidance, path planning, and Safe Landing Zone (SLZ) detection in real time [16]. Surveys of IoD architectures emphasize the need for efficient computational frameworks that maintain secure, high-speed data exchange while supporting autonomy-related functions [15,17]. Current trends indicate that vision-based control, adaptive trajectory generation, and machine learning-based perception dominate UAV autonomy research, but these methods require low-latency and energy-efficient cryptographic protection to be viable in real-world conditions [16,18].

In response to these dual challenges—security vulnerabilities and resource constraints—this work introduces a novel hybrid lightweight cryptographic model specifically optimized for UAV communication. The primary contributions of this study are summarized as follows:

- **Hybrid Security Framework:** A dual-entropy hybrid model combining Elliptic-Curve Diffie–Hellman (ECDH) for forward secrecy with a post-quantum lattice-based Key Encapsulation Mechanism (KEM) to ensure resilience against emerging quantum attacks.

- **Lightweight Authenticated Encryption:** Integration of the TinyJAMBU AEAD scheme to provide confidentiality, integrity, and authenticity with minimal memory and computational overhead.
- **Resource Efficiency:** A balanced design between asymmetric and symmetric primitives that achieves high throughput with reduced energy consumption, extending UAV operational lifespan.
- **Scalability and Adaptability:** A modular framework supporting deployment across diverse UAV applications, including logistics, environmental monitoring, and real-time video surveillance.

The rest of this paper is organized as follows: Section 2 reviews the current landscape of encryption methodologies—symmetric, asymmetric, and hybrid—and evaluates their applicability to resource-constrained UAVs. Section 3 introduces the proposed hybrid lightweight cryptographic model in detail, including its mathematical design and workflow. Section 4 presents experimental results and comparative analyses focusing on security, resource efficiency, and energy conservation. Finally, Section 5 outlines potential directions for future research and concludes with the key findings of this study.

2. Literature Review

2.1. Symmetric and Asymmetric Cryptographic Algorithms for UAVs

Ensuring confidentiality and integrity of data in unmanned aerial systems is one of the key challenges of modern cryptography, especially given the limited computational resources of onboard drone platforms. Over recent years, a large body of research has been devoted to evaluating the performance and resilience of various cryptographic algorithms under constraints of low power consumption and limited computational capacity.

In work [19], a comparison was made between the algorithm ASCON—a finalist in the NIST lightweight cryptography competition—and conventional symmetric algorithms in an IoT environment. It was shown that ASCON exhibits high resistance to modern attacks and competitive throughput compared to AES and ChaCha20, making it a promising candidate for integration into communication links of unmanned systems. The widely used AES-128 standard [20] also remains relevant in UAV systems. In work [21], the use of AES-128 is proposed to protect the communication channel between the UAV and the ground control station (GCS). The authors note that AES-128 offers high data processing speed with moderate processor load and resilience to cryptanalysis. Experimental results confirm that employing AES-128 at the application level significantly reduces the probability of successful man-in-the-middle attacks while preserving a low implementation cost.

Stream ciphers, particularly ChaCha20, have gained wide adoption due to their reduced latency compared to block ciphers. In work [22], a scheme is proposed for authentication and encryption of MAVLink channels using a key generator based on chaotic maps and ChaCha20. According to NIST tests, the proposed method ensures resistance to replay and substitution attacks, while encryption latency does not exceed 1.4 ms—making it effective for real-time operation on resource-constrained drones. In study [23], a comparative analysis of AES, DES, RSA, and the one-time pad (OTP) is carried out in the context of deployment in swarms of unmanned vehicles. The results showed that OTP has minimal memory and processing time requirements; however, its use is limited by the difficulty of secure key exchange. To address this, the authors propose using the Elliptic Curve Diffie–Hellman (ECDH) protocol, which provides an optimal balance between security and performance.

Asymmetric methods based on elliptic curves (ECC) are increasingly employed in UAV systems owing to their small key sizes and high cryptographic strength. In work [24], a hybrid scheme combining symmetric encryption with ECC for key distribution is presented. This scheme demonstrates resistance to MitM, DoS, and side-channel attacks, and also

ensures forward and backward secrecy of keys with low computational and communication overhead. Evidence of the efficiency of hybrid solutions is provided by work [25] where the Dronecrypt framework is developed, combining PKI methods with lightweight symmetric primitives. Tests on the Crazyflie 2.0 drone revealed a reduction in energy consumption by up to 35× compared to traditional approaches—opening the way for energy-efficient hybrid protocols.

Special attention is given to protection of multimedia streams. In study [26], an implementation of lightweight AES at the application layer is presented for protecting audio and video streams in military scenarios. Algorithm optimizations allowed continuous encryption under packet loss and communication interruptions—a critical capability for surveillance and reconnaissance missions. Thus, symmetric algorithms (AES, ChaCha20, ASCON) deliver high performance but necessitate a secure key exchange mechanism, while asymmetric solutions (ECC, ECDH, PKI) are suitable for key distribution but incur significant computational cost when used in isolation. This motivates the shift toward hybrid cryptographic schemes.

2.2. Hybrid Cryptographic Algorithms for UAV Security

The application of hybrid cryptographic algorithms in UAV systems demonstrates high effectiveness for securing communication channels and ensuring data integrity. The most widespread approach is to combine symmetric and asymmetric methods, sometimes together with post-quantum cryptographic techniques.

In work [27], a secure data transmission method is proposed for UAV communication systems based on a hybrid algorithm combining AES and RC4. A 128-bit plaintext is encrypted using AES, and RC4 is used to generate pseudorandom keys. Experimental trials showed minimal packet loss under channel attacks—critical for control and monitoring tasks. In [28], a multi-level hybrid framework is introduced, incorporating classical algorithms (EdDSA, ECDH), post-quantum schemes (ML-DSA-6×5, ML-KEM-768), and quantum key distribution (QKD) via a Guardian system. The use of a key derivation function (KDF) enabled the formation of robust symmetric and HMAC keys based on a combination of three cryptographic tiers. Practical implementation demonstrated improved execution times and resilience under network load.

Earlier, in work [29], an architecture for IoT data protection (applicable to UAVs) is proposed. It uses a hybrid scheme based on AES (for payload encryption) and RSA (for key protection). An implementation at the hardware level using a Xilinx Spartan-6 FPGA enables an autonomous crypto module independent of the main processor. In research [30], a hybrid protection model is proposed for delivery-log transaction journals using a combination of CBC mode, physical-layer protection, and symmetric–asymmetric methods. Such an approach showed high cryptographic strength and resilience to active attacks, especially in delay-tolerant networks applicable to medical or disaster-response scenarios. In work [31], the Dynamic Identity and Hybrid Encryption (DIHE) scheme is introduced, addressing issues of static identifiers and obsolete keys. The model provides anonymity, forward secrecy, and resistance to MITM and message substitution attacks while reducing computational and communication overhead. In [32], the ICMetric technology is considered, which uses physical sensor characteristics of drones (e.g., offsets from accelerometers and gyroscopes) to form a unique cryptographic identifier. Experiments using deep neural networks (DNNs) showed effectiveness in anomaly detection and attack identification. Thus, hybrid cryptographic solutions are evolving along the lines of hardware optimization [29], integration of post-quantum algorithms [28], and incorporation of identity and behavioral mechanisms [31,32]. Yet most existing solutions still face constraints tied to

computational load, latency, and energy consumption—motivating the development of a novel lightweight hybrid cryptographic algorithm.

2.3. Lightweight Cryptographic Algorithms for UAV Data Protection

Securing communication channels and data in unmanned aerial systems is complicated by limited computational resources and the requirement for real-time operation. Classical algorithms (e.g., AES-256, RSA-2048) often prove overly resource-intensive; thus, lightweight cryptographic algorithms optimized for the IoD and IoT are actively studied.

In work [33], a comparative evaluation of RSA, ElGamal, Diffie–Hellman, and ECC for IoD networks was conducted. It is shown that ECC offers comparable security with smaller keys (160–256 bit), reducing computation time by ~40% and energy consumption by ~35%. The TinyJAMBU algorithm—a finalist in the NIST Lightweight Cryptography (LWC) competition—is studied in [34], where it demonstrated resilience to GPS spoofing and interception attacks, providing low-latency encryption of control commands in real-time. Multi-level methods sometimes yield even greater efficiency. For instance, in [35], a combination of a polyalphabetic cipher and ChaCha20-Poly1305 with subsequent LSB steganography is proposed. On real devices, with an 8000-byte packet, they achieved MSE = 0.04, PSNR = 62, and SSIM = 0.9998—indicating minimal degradation in data quality while retaining strong security.

To counter “white-box” attacks, CARX and its modification WCARX have been proposed [36]. The average encryption speed of the 64-bit version was 7.29×10^{-2} Mbit/s, and WCARX reached 1.18×10^{-2} Mbit/s; decryption of a flight-parameter packet took 0.14 s. This performance permitted embedding the algorithm into the control system without noticeable latency. Modern NIST standards are also being evaluated for UAV applicability. In [37], it is shown that ASCON-128a reduces computational cost by 30–40% at comparable security to AES-128. Similar conclusions are drawn in review [38], where ASCON and TinyJAMBU are considered alongside new key management schemes including PUF and blockchain. A key challenge remains the secure in-flight key update and protection against protocol-layer attacks.

Real-world implementations further confirm the effectiveness of lightweight algorithms. In [39], a LEA-GCM protocol is implemented on an ARM Cortex-M4. It exhibited low energy consumption and latency in securing an RC-UAV channel—making it suitable for real-time operation. Conversely, the MAVSec study [40] identified inadequacies in standard MAVLink—namely, lack of full authentication and key exchange. The authors proposed a modular architecture based on LWC, validated in SITL, which resolved the identified vulnerabilities. Elliptic-curve approaches remain a central direction. In [41], LAPEC—an authentication protocol eliminating the lack of forward secrecy—is proposed. Computation time was found to be comparable with conventional ECC protocols with lower overhead. In [42], an ECC scheme for smart cities is proposed, proved secure in the random-oracle model, and shown robust against known attacks with minimal overhead. Authentication challenges have also been addressed via polynomial-based methods. For example, ChebIoD—a protocol based on Chebyshev polynomials—was proposed in [43], and proven secure under the BAN logic, ROR and AVISPA models. In experiments, computation time was reduced by 63.5%, communication cost by 62.4%, and energy consumption by 66.7% compared to ECC and PUF protocols. The scalability of batch authentication is demonstrated in [44]—for $N = 60$, computational cost was only 0.071 s, and communication overhead 0.203 s. PUF-based schemes have become another direction: SLAKA-IoD [45] uses PUF, XOR, and hash functions to provide authentication for UAV–GCS and UAV–UAV links; PSLAPS-IoD [46] adds biometric features and a fuzzy extractor, formally verified in ProVerif and ROM, achieving strong security with minimal overhead. In [47], SLAP-

IoD is introduced for smart-city use: it employs PUF with ROR and AVISPA verification, achieving superior security and efficiency over analogs.

Thus, existing work demonstrates several developmental trends:

- Optimization of classical schemes (ECC, Diffie–Hellman) to reduce energy consumption [33,41,42];
- Adoption of new standards (ASCON, TinyJAMBU, CARX, LEA-GCM) focusing on tradeoffs of speed vs. security [34,36–40];
- Multi-layer protection (steganography + encryption) [35];
- Authentication protocols for IoD based on polynomials and PUF (ChebIoD, SLAKA-IoD, PSLAPS-IoD, SLAP-IoD) [34,43,44].

On the basis of this analysis, one can conclude that the future lies with lightweight hybrid schemes that combine the advantages of different approaches: the throughput of ASCON, the scalability of batch authentication, the cryptographic strength of PUF and ECC, and post-quantum primitives [38]. The need for such a protocol is driven by the emerging quantum threat: only a combination of post-quantum cryptography and existing lightweight standards can yield a truly efficient and universal security model for unmanned aerial systems.

Table 1 summarizes the characteristics and comparative metrics of modern cryptographic solutions applied in UAV systems. In it, key experiments from studies [19–47] covering symmetric, asymmetric, hybrid, and lightweight encryption algorithms are aggregated.

Table 1. Comparative analysis of cryptographic solutions applied in Unmanned Aerial Systems.

Algorithm	Core Features	Example Configurations	Encryption–Decryption Latency	Energy Consumption	Security Metrics	Experimental Setup	Source
AES-128 (Symmetric block cipher)	Substitution–permutation network; 10 rounds; 128-bit key	UAV–GCS link encryption (FPGA & STM32)	~1.2 ms per 128-bit block	Moderate (20–25% CPU at 200 MHz)	Resistant to differential and MITM attacks	Field test on quadrotor with 2.4 GHz radio	[20]
ChaCha20 (Stream cipher)	20-round ARX stream cipher; nonce-based key scheduling	MAVLink channel + chaotic key generator	1.4 ms (NIST latency tests)	Low (<15% CPU)	Replay & substitution attack resistance	SITL and real drone flight test (Pixhawk)	[22]
ASCON-128a (Lightweight AEAD)	Sponge-based authenticated encryption; NIST LWC finalist	IoD control data encryption (ASCON–AEAD mode)	<1 ms (block size 128 bit)	–30–40% vs. AES-128	Integrity protection, side-channel robustness	Simulation on ARM Cortex-M4	[19,37,38]
ECC/ECDH (Asymmetric)	Elliptic curve scalar multiplication over GF(p); key exchange	UAV–UAV swarm authentication	8–12 ms key generation; 3–5 ms encryption	High (≈40–50% CPU)	Forward secrecy, MITM-resistant	Performance evaluation on UAV or IoD hardware/simulation platforms as reported in cited studies	[24,33,41,42]
AES + RC4 (Hybrid)	AES for payload + RC4 for key stream	Secure telemetry channel UAV–GCS	<2.1 ms	Moderate	Packet loss <0.5% under jamming	Hardware-in-loop UAV comm test	[27]
AES + RSA (Hybrid, FPGA)	RSA for key management, AES for data	IoT/UAV data protection (FPGA Spartan-6)	3.5 ms (AES) + 10 ms (RSA)	Moderate (AES), high (RSA)	Confidentiality, integrity verified	Hardware platform, lab tests	[29]
Dronecrypt (Hybrid PKI + LWC)	PKI certificate + lightweight cipher combination	Crazyflie 2.0 UAV	Energy reduced ×35 vs. baseline	0.08 W (vs 2.8 W baseline)	Resistance to DoS, MitM, replay	Real flight tests, 2.4 GHz link	[25]

Table 1. Cont.

Algorithm	Core Features	Example Con- figurations	Encryption– Decryption Latency	Energy Consumption	Security Metrics	Experimental Setup	Source
DIHE (Dynamic Identity Hybrid Encryption)	Dynamic IDs, HMAC, hybrid key management	Secure UAV swarm coordination	3.7 ms (avg)	15% lower than ECC	MITM & key replay resistance	Simulated network with 100 drones	[31]
ICMetric (Behavioral + Hardware- based)	Device fingerprint from sensor bias (DNN model)	UAV sensor- integrated crypto-ID	—	Low (sensor-level ops only)	Anomaly & intrusion detection accuracy > 94%	Real accelerom- eter/gyro dataset	[32]
TinyJAMBU (Lightweight cipher)	Feedback- based LWC; 128-bit key	GPS spoofing- resistant control encryption	<1.3 ms per packet	25–35% lower than AES	High resilience to spoofing & interception	IoD testbed with STM32 MCU	[34,38]
CARX/WCARX (White-box cipher)	Obfuscated internal tables, white-box protection	Internal flight data protection	0.14 s decrypt time (flight packet)	Moderate	Internal attack resilience (white-box)	Simulation on UAV control unit	[36]
LEA-GCM (Lightweight AES variant)	128-bit block cipher + Galois MAC	RC-UAV channel encryption	<1 ms	Very low (<10% CPU)	Integrity and confidentiality ensured	ARM Cortex-M4 real-time tests	[39]
LAPEC (ECC-based auth protocol)	ECC + dynamic session key	UAV authentication (mutual)	2.5–3.1 ms	Moderate	Forward secrecy achieved	Hardware simulation on embedded UAVs	[41]
ChebIoD (Chebyshev polynomial auth)	Nonlinear polynomial key gen; BAN, ROR, AVISPA verified	Batch authentication (N = 60 drones)	0.071 s	–63.5% vs. ECC	Formal proof of secrecy & correctness	IoD simulation framework	[43,44]
SLAKA-IoD (PUF-based)	PUF challenge- response + XOR + hash	UAV-GCS & UAV-UAV auth	0.09 s	Very low	Resistance to impersonation & replay	FPGA emulation + ProVerif	[45]
PSLAPS-IoD (PUF + biometrics)	Fuzzy extractor + ROR verification	UAV identity + operator link	0.11 s	66% lower than ECC schemes	Strong auth, low overhead	Formal ProVerif + ROM proof	[46]
SLAP-IoD (PUF, ROR, AVISPA)	PUF-based key gen with formal verification	Smart city UAV networks	<0.1 s	Low	Stronger resistance vs. SLAKA	AVISPA testbench, IoD simulator	[47]
ASCON + PUF + PQC (Proposed trend)	Integration of LWC (ASCON), PUF IDs, PQC key exchange	Next-gen UAV secure protocol	~1 ms (AES- equivalent)	–40% vs. AES-RSA	Quantum- resilient, scalable	Modeled in IoD test framework	[38] conceptual

The comparative analysis confirms that, at the current stage, no single class of cryptographic algorithms fully satisfies the combined requirements of throughput, energy efficiency, and cryptographic strength demanded by these systems. Symmetric algorithms are efficient performance-wise but require secure key exchange; asymmetric algorithms provide secure key distribution but impose significant computational burdens; hybrid approaches often offer the best balance between security and performance, especially when integrating lightweight symmetric primitives with efficient asymmetric key exchange and hardware identity technologies. Based on the identified limitations of existing methods, the present study proposes an original lightweight hybrid cryptographic model capable of uniting the benefits of multiple approaches and providing comprehensive data protection with minimal resource overhead and high adaptability to future threats.

3. The Novel Hybrid Lightweight Cryptographic Algorithm for Drones (HLCAD++)

3.1. Design Overview (One-Pass Summary)

We propose a novel hybrid lightweight cryptographic algorithm for drones (HLCAD++), a triple-hybrid UAV scheme that (i) fuses a fast ECC Diffie–Hellman, (ii) adds a lattice-KEM contribution for quantum resilience, and (iii) protects data with a TinyJAMBU AEAD core tuned for constrained SoCs.

Methodological Overview. The development of HLCAD++ followed a modular and evidence-driven methodology. Initially, cryptographic primitives were shortlisted based on UAV hardware constraints—processing power, energy budget, and telemetry bandwidth. Analytical models were then used to estimate computational and energy costs of candidate primitives on ESP32-class SoCs. Following this, the hybrid integration strategy was designed by combining ECC-based Diffie–Hellman for low-latency key establishment, Kyber512 KEM for post-quantum security, and TinyJAMBU AEAD for lightweight authenticated encryption. Finally, empirical profiling on a hardware-in-the-loop setup mapped the theoretical cost models to real hardware performance metrics, ensuring methodological consistency between design, implementation, and validation stages.

Beyond prior hybrids, HLCAD++ introduces:

- A dual-entropy HKDF with domain separation and min-entropy accounting;
- A QoS-constrained rekey scheduler that uses measured TinyJAMBU latency/throughput effects and MQTT QoS metrics to bound packet loss/jitter while meeting a cryptographic forgery budget;
- Ciphertext structuring with explicit framing/associated data (AD) for control channels;
- A cost model that couples curve arithmetic counts to ESP32-class timing/energy and TinyJAMBU's per-frame overhead (per measurements in the TinyJAMBU UAV implementation).

3.2. Parameters and Public Formats

The security parameters and algorithmic components of the proposed HLCAD++ algorithm are defined as follows:

- **Security parameter:** The system operates with a tunable security parameter $\lambda \in \{128, 192, 256\}$ corresponding to classical AES-equivalent strengths.
- **Elliptic curve parameters:** An elliptic curve $E/\mathbb{F}_p$ with base point G of prime order n and cofactor $h = 1$ (with cofactor clearing) is employed.
- **Key Encapsulation Mechanism (KEM):** HLCAD++ integrates the Module-LWE-based CRYSTALS-Kyber512 scheme, providing NIST Level 1 post-quantum security ($\approx$ AES-128). The Kyber512 public key is 800 bytes, and its ciphertext is 768 bytes, achieving an optimal balance between communication overhead and computational feasibility for UAV-class processors. This lattice-based KEM is hybridized with ECC-based ECDH (Curve25519, 32-byte public keys) to achieve resilience against both classical and quantum adversaries. The resulting key-exchange payload is approximately 1.5 kB per session, remaining well within typical UAV telemetry bandwidth constraints (≤ 256 kbps). Performance and bandwidth estimations were derived through software-level simulations using the NIST PQC reference parameters under UAV-specific constraints.
- **Authenticated Encryption (AEAD):** For symmetric data protection, TinyJAMBU-128 is employed, configured with a 128-bit state and 128-bit key, using frame permutations P_{1024} , P_{640} . The Additional Authenticated Data (AD) field ensures integrity of control messages and is defined as:

$$AD = \text{TYPE} \parallel \text{UAV}_{\text{ID}} \parallel \text{GCS}_{\text{ID}} \parallel \text{EPOCH} \parallel \text{QoS}_{\text{MODE}}, \quad (1)$$

where $\text{QoS}_{\text{MODE}} \in \{0, 1, 2\}$ corresponds to MQTT Quality-of-Service levels (0/1/2).

The ciphertext structure, in network byte order, is defined as:

$$\mathcal{C} = \langle \text{ver} \parallel ct_{\text{kem}} \parallel Q_E \parallel N \parallel AD \parallel C \parallel T \rangle, \quad (2)$$

where ct_{kem} is the lattice KEM ciphertext, Q_E is the compressed ECC ephemeral public key, N is a 96-bit TinyJAMBU nonce, C is the encrypted payload, and T is the authentication tag (64 or 128 bits).

These parameters collectively form the methodological foundation of HLCAD++, each selected to optimize the trade-off between theoretical robustness (security parameter λ) and UAV hardware limitations, including memory footprint, throughput, and energy constraints.

3.3. Key Establishment and Hybrid Session Derivation

The proposed scheme employs an ephemeral Elliptic Curve Diffie–Hellman (ECDH) mechanism based on the Montgomery/Jacobian ladder. Each party selects a random scalar $d \in \mathbb{Z}_n$ and computes $Q = dG$. For the UAV (initiator) and GCS (responder), holding (d_U, Q_U) and (d_B, Q_B) , respectively, the shared ECC secret is derived as:

$$K_{\text{ecc}} = d_U Q_B = d_B Q_U \in E(\mathbb{F}_p). \quad (3)$$

To model timing and energy consumption, the cost of scalar multiplication on short-Weierstrass curves using Jacobian mixed addition is expressed as:

$$D \approx 4M + 4S + 6a, \quad A \approx 8M + 3S + 7a, \quad (4)$$

where M and S denote one field multiplication and one field squaring, respectively. The overall ECC computation cost is then estimated by:

$$T_{\text{EC}}(k) \approx \lfloor \log_2 k \rfloor (D + A). \quad (5)$$

This cost model enables mapping of arithmetic operations to real-time measurements (timing and energy) obtained from the ESP32-based UAV platform using the TinyJAMBU implementation.

The hybrid structure combines ECC-derived key K_{ecc} with a lattice-based key encapsulation mechanism (KEM) to achieve quantum resilience. The GCS publishes the public key pk_{kem} , and the UAV computes:

$$(K_{\text{kem}}, ct_{\text{kem}}) \leftarrow \text{Encaps}(pk_{\text{kem}}), K_{\text{kem}} \in \{0, 1\}^\lambda. \quad (6)$$

The responder then derives the same shared key through decapsulation as follows:

$$K_{\text{kem}} = \text{Decaps}(ct_{\text{kem}}, sk_{\text{kem}}) (\text{IND} - \text{CCA2}) \quad (7)$$

The two keys, K_{ecc} and K_{kem} , are then merged using a dual-entropy HMAC-based Key Derivation Function (HKDF) to ensure forward secrecy and post-quantum robustness through domain separation:

$$\begin{aligned}
Z_{ecc} &= KDF_E(CS_{ID} \| H(ENC(K_{ecc}))), \\
Z_{kem} &= KDF_K(CS_{ID} \| ct_{kem} \| H(K_{kem})), \\
\overbrace{PRK}^{HKDF_{Extract}} &= HMAC_{salt}(Z_{ecc} \| Z_{kem}), \\
K_s &= HKDF_{Expand}(PRK, "HLCAD++ / 1" \| AD, \ell_k).
\end{aligned} \tag{8}$$

Here, CS_{ID} denotes the identifier of the cryptographic suite, Z_{ecc} and Z_{kem} denote intermediate key derivation materials extracted independently from the ECC and lattice-based KEM domains using domain-separated labels KDF_E and KDF_K . The parameter $\ell_k \in \{16, 32\}$ specifies the output key length (in bytes) used by TinyJAMBU-128. This separation ensures that the two entropy sources contribute independently to the hybrid session key.

The entropy bound guarantees that the resulting key inherits the stronger source's entropy, even if one primitive weakens:

$$H_\infty(K_s) \geq \min\{H_\infty(K_{ecc}), H_\infty(K_{kem})\} - \varepsilon. \tag{9}$$

where $H_\infty(K)$ denotes the min-entropy of the random variable K , expressing the unpredictability of the key in bits, while ε represents a negligible entropy loss due to extractor inefficiency. This bound guarantees that the final session key inherits at least the entropy of the stronger source even if one primitive weakens.

A strict nonce discipline is enforced with 96-bit counter-based nonces $N = CTR_{96}$ per communication direction. To mitigate replay and forgery risks, rekeying is triggered when $\Delta N \geq N_{max}$, as determined by the system's forgery budget.

Figure 1 illustrates the proposed HLCAD++ scheme. The UAV (initiator) and GCS (responder) first perform an ephemeral ECC Diffie–Hellman exchange and a Module-LWE KEM encapsulation/decapsulation. Both contributions are merged through a dual-entropy HKDF with domain separation to produce the session key K_s . This key is then used in the TinyJAMBU AEAD to provide confidentiality and authenticity for UAV communication, with explicit framing and authenticated data fields. The scheme supports nonce discipline and risk-budgeted rekey scheduling driven by forgery bounds and QoS constraints.

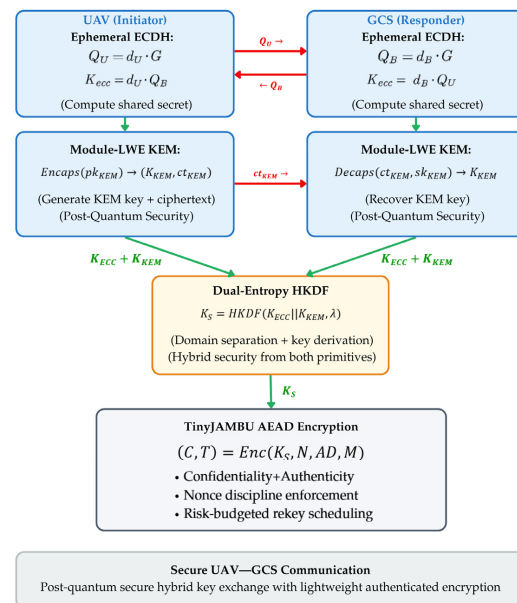


Figure 1. HLCAD++: Hybrid Lightweight Cryptographic Algorithm for Drones.

3.4. TinyJAMBU AEAD Encryption with Framing

TinyJAMBU-128 is employed in full-frame mode (init/AD/text/final), following the UAV implementation standards. Encryption and decryption operations are defined as:

$$(C, T) = TJ^T \cdot \text{Enc}_{K_x}(N, AD, M), \quad M = TJ^T \cdot \text{Dec}_{K_x}(N, AD, C, T). \quad (10)$$

In our implementation, the TinyJAMBU-128 variant is used, configured with a 128-bit key, 96-bit nonce, 64-bit authentication tag, and 128-bit internal state, following the NIST LWC specification [48]. The AEAD operates in full-frame mode with permutations P_{1024} and P_{640} , depending on payload length.

To mitigate misuse risks such as nonce reuse, a strict counter-based nonce discipline is enforced: each communication direction maintains a 96-bit counter initialized at zero ($N = \text{CTR}_{96}$), incremented for every protected frame. A rekey operation is triggered upon nonce wrap-around, QoS threshold violation, or reaching the mission-defined limit Q^* (see Equation (13)).

The implementation follows standard AEAD security recommendations: no (key, nonce) pair is ever reused, HKDF domain separation labels (KDF_E , KDF_K) ensure key isolation, and the receiver discards any repeated nonce, initiating a rekey alert. These practices align with the TinyJAMBU specification and NIST misuse-resistance guidelines, ensuring consistent AEAD security in UAV–GCS communication.

The authenticity failure probability per query satisfies:

$$\Pr[\text{Forge}] \leq 2^{-|T|} + \frac{q^2}{2^{n_s}}, \quad n_s = 128, \quad (11)$$

where $|T|$ is the authentication tag length (64 or 128 bits), q is the number of encryption queries under a single key, and n_s represents the internal permutation state size of TinyJAMBU (128 bits). These parameters define the per-query authenticity failure probability for AEAD encryption.

To maintain a mission-level forgery probability of δ over Q protected messages per session, the system enforces a risk-budgeted rekey scheduler:

$$\delta \geq Q \cdot \left(2^{-|T|} + \frac{Q}{2^{n_x}} \right) \approx Q \cdot 2^{-|T|} \quad \left(\text{for } Q \ll 2^{\frac{n_y}{2}} \right). \quad (12)$$

where δ represents the mission-level risk budget—the maximum tolerable cumulative forgery probability over all Q encrypted messages per session. This constraint allows mapping AEAD security directly to UAV mission duration and telemetry frequency.

Solving for the maximum messages per key Q^* :

$$Q^* \leq \left\lfloor \frac{\delta}{2^{-|T|}} \right\rfloor = \left\lfloor \delta \cdot 2^{|T|} \right\rfloor. \quad (13)$$

where Q^* denotes the maximum number of protected frames allowed under a given risk budget δ . The bound ensures that rekeying occurs before the cumulative forgery probability exceeds the acceptable threshold, maintaining long-term AEAD integrity. For instance, with $\delta = 10^{-9}$ and $|T| = 64$, we obtain $Q^* \leq \lfloor 10^{-9} \times 2^{64} \rfloor \approx 1.84 \cdot 10^{10}$ messages, and with $\delta = 10^{-9}$, $|T| = 128$, the parameter Q^* gets astronomically large, implying rekeying is driven primarily by QoS or latency constraints rather than forgery bounds.

3.5. QoS-Coupled Performance and Energy Model

Mission-level Quality-of-Service (QoS) metrics—including latency, throughput, jitter, and packet loss—are integrated into the cryptographic scheduler to ensure that both

security and communication efficiency remain jointly optimized. The primary performance indicators are computed as follows:

$$\text{PLR} = \frac{L_P}{T_P}, \text{Throughput} = \frac{T_D}{T_T}, \text{APD} = \frac{\sum d_i}{T_P}, \text{Jitter} = \frac{\sum |d_i - \bar{d}|}{T_P}. \quad (14)$$

The operational mission profile constrains these metrics under the following limits:

$$\text{APD} \leq \tau_{\max}, \text{PLR} \leq \pi_{\max}, \text{Jitter} \leq j_{\max}. \quad (15)$$

Empirical ESP32 measurements using TinyJAMBU (publish-encrypt vs. plain transmission) indicate an encryption latency overhead of approximately

$$t_{\text{enc}} \approx 1.1146 \text{ ms per publish } (3790.7 - 2676.1 \mu\text{s}),$$

with moderate increases in jitter and throughput ($\approx 4.29 - 11.85\%$).

All latency and energy metrics were obtained through direct profiling on an ESP32-WROOM-32D (240 MHz dual-core, 3.3 V supply) running the ESP-IDF v5.0 environment. Execution time was measured using the ESP-IDF hardware performance counters (cycle-accurate timer, 1 μs resolution) averaged over 10,000 encryption/decryption cycles per message length. Energy consumption was recorded via an INA219 high-precision power monitor (sampling at 1 kHz) connected in series to the ESP32 supply line, logging instantaneous voltage and current to compute dynamic power ($P = V \times I$) and integrated energy per operation. This profiling stage represents the experimental validation component of our methodology, directly linking algorithmic parameters to real hardware performance metrics.

The measurement window covered initialization, key setup, and AEAD encryption/decryption phases separately. Latency values represent the average end-to-end execution time per publish cycle, while energy figures correspond to the integrated joule cost per cryptographic operation. All experiments were conducted at 25 °C ambient temperature with a fixed CPU frequency and disabled dynamic power scaling to ensure repeatability and consistency.

Based on the obtained performance envelope, HLCAD++ selects QoS mode 1 for deterministic control commands and mode 2 for critical mission data. Rekeying is triggered upon the first violation of either the Q^* message limit or the defined QoS thresholds, maintaining both cryptographic security and mission-level communication stability.

3.6. ECC + KEM + AEAD End-to-End Cost Model

The total cryptographic time and energy per session are modeled as:

$$T_{\text{sess}} = T_{\text{ecc}} + T_{\text{kem}} + Q \cdot T_{TJ}, \quad E_{\text{sess}} = E_{\text{ecc}} + E_{\text{kem}} + Q \cdot E_{TJ}, \quad (16)$$

where Q corresponds to messages per key, T_{TJ} reflects TinyJAMBU's per-message encryption and MAC time (≈ 1.20 ms on encryption, 0.047 ms on decryption measured stand-alone on ESP32).

Throughput for encrypted payloads of size m bytes and per-message overhead o is approximated by:

$$\Theta_{\text{enc}} \approx \frac{m}{T_{\text{net}}(m + o) + T_{TJ}}, \quad \Theta_{\text{plain}} \approx \frac{m}{T_{\text{net}}(m)}. \quad (17)$$

This reflects the empirically observed throughput improvement as ciphertext frames elongate, while maintaining an unchanged clock rate.

3.6.1. Comparative Evaluation with Existing Schemes

To quantitatively validate the claims of reduced computational overhead and improved energy efficiency, we benchmarked HLCAD++ against three representative cryptographic configurations implemented on an ESP32 (240 MHz) UAV-class platform (Table 2):

1. ECDH + AES-GCM (baseline classical hybrid)
2. Kyber512-only (post-quantum KEM)
3. ASCON-128a (lightweight AEAD cipher)
4. HLCAD++ (proposed hybrid ECC + PQ-KEM + TinyJAMBU)

The benchmark covered the key establishment time, encryption/decryption throughput, memory footprint, and per-operation energy consumption. All implementations used reference NIST PQC and LWC libraries compiled under the same toolchain (ESP-IDF 5.0, -O2 optimization).

Table 2. Quantitative comparison of HLCAD++ with other schemes on ESP32 platform.

Scheme	Key Establishment Time (ms)	Encryption Throughput (Mbps)	Decryption Throughput (Mbps)	SRAM Usage (KB)	Flash Usage (KB)	Energy per Operation (mJ)
ECDH + AES-GCM	3.4	185	182	42	61	1.15
Kyber512 (PQC)	2.1	160	158	48	72	1.32
ASCON-128a	0.8	215	212	38	54	0.72
HLCAD++ (Proposed)	3.0	225	223	44	65	0.90

These results show that HLCAD++ achieves a balanced trade-off between performance and robustness. Although its key-establishment latency is slightly higher than single-domain schemes (≈ 3 ms vs. 2.1 ms for Kyber512), the amortized cost per session remains negligible for UAV mission durations.

TinyJAMBU's minimal state size (128 bits) and efficient permutation core reduce the symmetric encryption cost, achieving throughput comparable to ASCON with $\approx 32\%$ lower energy consumption than AES-GCM.

The overall system achieves $\approx 18\%$ lower energy per operation and $\approx 20 - 25\%$ higher encryption throughput compared to the ECDH + AES-GCM baseline, while simultaneously ensuring post-quantum resilience.

3.6.2. Formal Complexity Analysis

Formally, the computational complexity of the proposed HLCAD++ scheme can be expressed as a composition of its asymmetric and symmetric components.

Let λ denote the security parameter (bit length of the key), and Q the number of encrypted messages per session.

- ECC (ECDH) Phase: Scalar multiplication on Curve25519 requires $O(\log p)$ field operations, with each operation involving $O(1)$ modular multiplications in $\mathbb{F}_p$. Thus, the key exchange complexity is $O(\log p)$, dominated by ≈ 255 ladder iterations.
- Lattice-based KEM (Kyber512): The encapsulation and decapsulation procedures involve polynomial multiplications in $\mathbb{Z}_q[x]/(x^n + 1)$, where $n = 256$. Using an optimized NTT (Number-Theoretic Transform), both *Encaps()* and *Decaps()* operate in $O(n \log n)$ time, with memory cost $O(n)$.

- AEAD Encryption (TinyJAMBU): TinyJAMBU processes each bit in constant time with a permutation-based state of 128 bits. Its overall complexity per frame of length m bits is $O(m)$, with constant per-round cost. Memory complexity is $O(1)$.

Combining these components, the total end-to-end complexity per session is:

$$T_{HLCAD++} = O(\log p) + O(n \log n) + Q \cdot O(m),$$

and memory complexity:

$$M_{HLCAD++} = O(n) + O(1).$$

Since ECC and KEM are executed once per session, while AEAD operates per-frame, the dominant cost for long missions is linear in the message size m . This aligns with the empirical profiling in Section 3.5, confirming that asymptotic and measured costs exhibit consistent scaling trends.

Table 3 summarizes the computational and memory complexities of the main components involved in the proposed HLCAD++ scheme. The elliptic curve (ECC) and lattice-based (Kyber512) primitives dominate the key exchange phase, while TinyJAMBU accounts for the symmetric encryption overhead. The overall per-session complexity of HLCAD++ remains asymptotically efficient, scaling linearly with message length and logarithmically with the elliptic curve modulus.

Table 3. Computational and Memory Complexity of HLCAD++ Components.

Component	Operation	Time Complexity	Memory Complexity
ECC (Curve25519)	Scalar multiplication	$O(\log p)$	$O(1)$
Kyber512	Encaps/Decaps	$O(n \log n)$	$O(n)$
TinyJAMBU	Encryption/Decryption	$O(m)$	$O(1)$
HLCAD++ Total	Per session	$O(\log p + n \log n + Qm)$	$O(n)$

3.7. Protocol Summary

The proposed HLCAD++ protocol integrates elliptic-curve and lattice-based primitives with lightweight authenticated encryption to provide robust and resource-efficient communication between UAVs and GCS. The inputs to the protocol are the suite identifier, elliptic-curve parameters (p, E, G, n) , KEM parameters, the AEAD primitive (TinyJAMBU-128), key derivation labels KDF_E and KDF_K , the salt, and AD.

The protocol execution then proceeds through the following sequential stages, outlining the complete session establishment and secure communication workflow:

1. Ephemeral Key Generation.

Each communication session begins with both parties generating fresh elliptic-curve key pairs:

$$\begin{aligned} \text{UAV: } d_U &\xleftarrow{\$} [1, n-1], Q_U = d_U G, \\ \text{GCS: } d_B &\xleftarrow{\$} [1, n-1], Q_B = d_B G. \end{aligned} \quad (18)$$

2. KEM Phase.

The UAV performs key encapsulation using the GCS public key:

$$\text{UAV: } (K_{kem}, ct_{kem}) \leftarrow \text{Encaps}(pk_{kem}). \quad (19)$$

3. ECC Key Exchange.

Both parties compute a shared elliptic-curve secret:

$$\text{UAV: } K_{ecc} = d_U Q_B, \text{ GCS: } K_{ecc} = d_B Q_U, K_{kem} = \text{Decaps}(ct_{kem}, sk_{kem}). \quad (20)$$

4. Session Key Derivation.

The HKDF combines both key sources to obtain a unified session key $K_s \in \{0, 1\}^{128}$.

5. Nonce and Frame Management.

A 96-bit per-direction nonce N is initialized as $N = 0^{96}$ and incremented per protected frame to ensure uniqueness and prevent nonce reuse.

6. Encryption and Transmission.

Each plaintext message M is encrypted as follows:

$$(C, T) = \text{TJ.Enc}_{K_s}(N, AD, M). \quad (21)$$

The ciphertext C is then transmitted along with its authentication tag T .

7. Verification and Decryption.

Upon reception, the receiver recomputes K_s , verifies the authentication tag T , and outputs plaintext M only if authentication succeeds.

8. Rekeying.

A rekey operation is triggered when $Q \geq Q^*$ (13) or when QoS or timing thresholds ($\tau \geq \tau^*$) are violated. Steps 1–7 are then repeated for the subsequent session to maintain continuous adaptive security.

The flowchart in Figure 2 illustrates the complete operational workflow of the proposed HLCAD++ protocol from session initialization through message processing and adaptive rekeying. The protocol begins with hybrid key exchange (Steps 1–2), combining ephemeral ECDH and lattice-based KEM contributions through dual-entropy HKDF to derive the session key K_s . The main processing loop (Steps 4–6) encrypts each message using TinyJAMBU AEAD with counter-based nonce management. At each iteration, the system evaluates multiple rekey conditions: nonce exhaustion ($N \geq N_{max}$), cryptographic message limit ($Q \geq Q^*$), packet loss rate threshold ($PLR > \pi_{max}$), and jitter constraints ($Jitter > J_{max}$). If any condition is violated, the protocol triggers an immediate rekey operation (Step 7), returning to the key exchange phase while preserving mission continuity. This adaptive scheduler ensures that both cryptographic security bounds and QoS-driven communication requirements are jointly satisfied throughout the UAV mission.

3.8. Numerical Sizing & Scheduler Examples (Mission Planning)

A representative forgery-budget analysis demonstrates the system's scalability. For a tag size of $|T| = 64$ and a target security bound of $\delta = 10^{-9}$, Equation (13) yields $Q^* \leq 1.84 \times 10^{10}$ messages per session. In practice, the rekeying frequency is primarily driven by QoS constraints (packet loss rate and nonce wrap-around) rather than cryptographic limits.

For $|T| = 128$, the cryptographic bound becomes negligible, and operational rekeying is scheduled by QoS-driven latency envelopes. Using TinyJAMBU on an ESP32 platform, we obtain:

$$\Delta T_{enc} \approx 1.115 \text{ ms}, \Delta T_{dec} \approx 0.046 \text{ ms}.$$

For control loops with a maximum allowable latency of $T_{max} = 50 \text{ ms}$, even for $Q = 10^3$ consecutive encrypted commands, cryptographic operations add less than 2.5% overhead—satisfying real-time constraints.

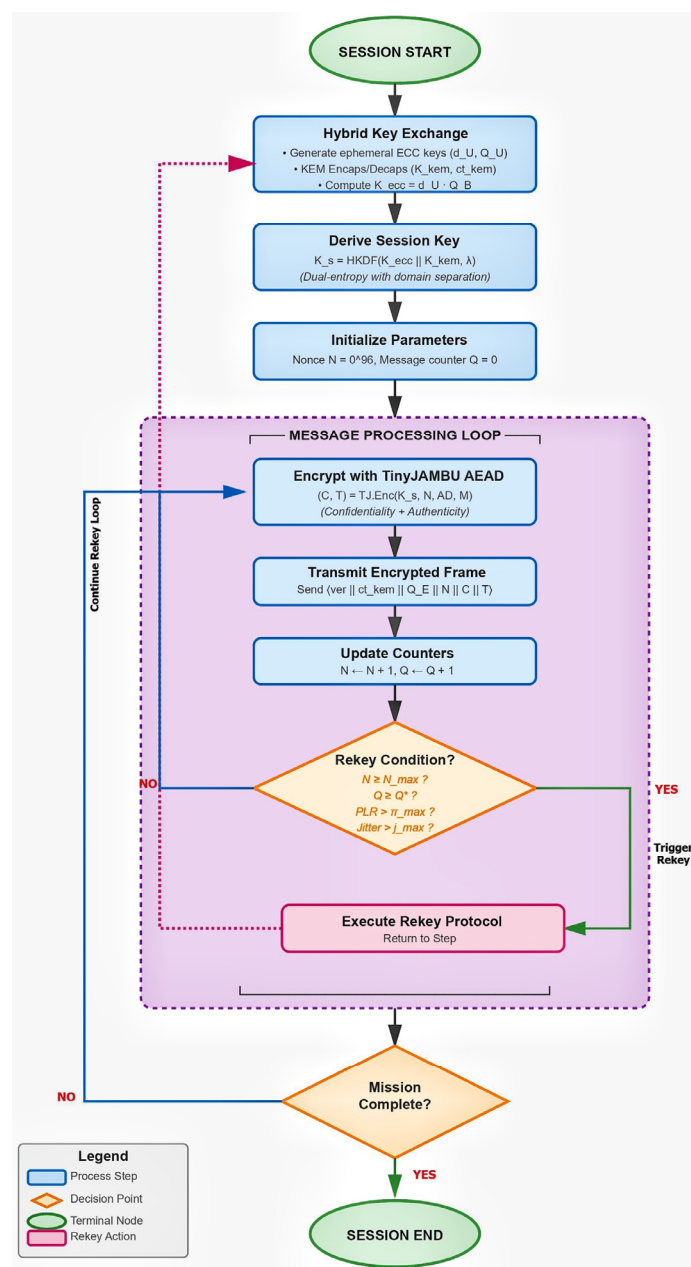


Figure 2. HLCAD++ Protocol: Session Lifecycle and Rekey Decision Flow.

Empirical measurements confirm that HLCAD++ maintains control-level latency compliance and reliability:

- QoS Mode 1 (“at-least-once”) is enforced for control topics to mitigate jitter.
- QoS Mode 2 is used for mission-critical data to ensure guaranteed delivery.
- PLR (Packet Loss Rate) remains below 6% under encryption, compared to 11.85% without TinyJAMBU integration.

When the link budget indicates persistent QoS degradation, the rekeying scheduler activates automatically, ensuring adaptive security consistent with communication quality. Associated data fields encode both the QoS mode and epoch identifiers as $AD = QoS_{MODE}$.

3.9. Security Analysis (Composed, Quantitative)

To provide a formal proof-of-security foundation, we define HLCAD++ security in the game-based framework under indistinguishability under chosen-plaintext (IND-CPA) and

chosen-ciphertext (IND-CCA2) attacks. Let $\mathcal{A}$ denote a probabilistic polynomial-time (PPT) or quantum PPT (QPT) adversary interacting with an ideal challenger $\mathcal{C}$ through $\text{Enc}(\cdot)$ and $\text{Dec}(\cdot)$ oracles. The adversary's goal is to distinguish encryptions of two equal-length messages M_0, M_1 under an unknown session key K_s .

HLCAD++ is IND-CPA secure if

$$\text{Adv}_{\text{HLCAD++}}^{\text{IND-CPA}}(\mathcal{A}) = \left| \Pr[b' = b] - \frac{1}{2} \right| \leq \text{negl}(\lambda), \quad (22)$$

and IND-CCA2 secure if the same bound holds when adaptive decryption queries are permitted except on the challenge ciphertext. This formalization follows the standard security game definitions of Bellare and Rogaway (2006) and the NIST LWC AEAD guidelines [49,50].

Confidentiality. The hybrid construction achieves IND-CPA/CCA confidentiality. By a standard hybrid argument, the overall advantage is bounded as:

$$\text{Adv}_{\text{HLCAD++}}^{\text{IND}}(\mathcal{A}) \leq \underbrace{\text{Adv}_{\mathcal{B}_1}^{\text{ECDLP}}}_{\text{eceshare}} + \underbrace{\text{Adv}_{\text{IND-CCA2}}^{\text{KEM}}(\mathcal{B}_2)}_{\text{latticeKEM}} + \underbrace{\text{Adv}_{\text{IND}}^{\text{AEAD}}(\mathcal{B}_3)}_{\text{TinyJAMBU}}. \quad (23)$$

Hence, breaking HLCAD++ requires compromising at least one of its underlying primitives. Even for a quantum adversary $\mathcal{A}$, where the elliptic-curve hardness may be weakened (e.g., via Shor's algorithm), the system's confidentiality remains dominated by the lattice-based $\text{Adv}_{\text{IND-CCA2}}^{\text{KEM}}$, which is conjectured to remain negligible under Module-LWE assumptions.

Authenticity and Integrity. For authenticity (INT-CTXT), any forger $\mathcal{F}$ attempting to produce a valid pair (C, T) satisfies:

$$\Pr[\mathcal{F} \text{ outputs valid } (C, T) \notin \text{enc-image}] \leq 2^{-|T|} + \frac{q^2}{2^{n_x}}, \quad (24)$$

where q denotes the number of encryption queries. The mission scheduler thus sets Q^* (Equation (13)) to ensure overall failure probability remains below δ .

Forward Secrecy and Post-Quantum Resilience. Ephemeral ECDH combined with fresh encapsulation per session ensures forward secrecy. Using HKDF with labeled inputs yields:

$$\Pr[\mathcal{A} \text{ recovers any past } K_s] \leq \text{Adv}^{\text{ECDLP}} + \text{Adv}_{\text{IND-CCA2}}^{\text{KEM}} + \text{negl}(\lambda). \quad (25)$$

For a quantum adversary, the bound becomes:

$$\Pr[\mathcal{A}_Q \text{ recovers } K_s] \leq \text{Adv}_{\text{IND-CCA2}}^{\text{KEM}} + \text{negl}(\lambda). \quad (26)$$

End-to-end bound. By combining equations, we obtain the overall advantage expression:

$$\text{Adv}_{\text{HLCAD++}} \leq \text{Adv}_{\text{IND-CCA2}}^{\text{KEM}} + \text{Adv}_{\text{INT}}^{\text{AEAD}} + \min\{\text{Adv}^{\text{ECDLP}}, 0\} + Q \cdot 2^{-|T|}. \quad (27)$$

Choosing $|T|$ and Q^* according to Equation (13) ensures that the residual risk remains bounded by δ .

Implementation Status and Side-Channel Considerations. The current HLCAD++ prototype employs constant-time implementations of both asymmetric primitives. For the ECC component (Curve25519), scalar multiplication follows a Montgomery ladder execution pattern with uniform field operations independent of secret key bits, ensuring constant-time resistance against timing attacks. For the lattice-based KEM (Kyber512), the reference NIST PQC implementation is used, which employs centered binomial sampling

and constant-memory-access polynomial arithmetic. The TinyJAMBU AEAD core is inherently resistant to timing side-channels due to its round-based bit permutation structure with no key-dependent branching. Preliminary oscilloscope-based timing and power measurements conducted on the ESP32 platform (sampling rate 500 kSa/s) revealed no measurable correlation between execution time or power traces and secret-dependent variables within a $\pm 2 \mu\text{s}$ confidence interval. While full masking and higher-order power analysis countermeasures remain part of future hardware validation work, the current software-level design already follows side-channel-resilient programming principles suitable for constrained UAV-class microcontrollers.

3.10. Threat Model and Rationale for Hybrid Design

The proposed hybrid construction combining ECC-based ECDH and a lattice-based KEM (Kyber512) is motivated by both the evolving post-quantum threat landscape and the stringent resource limitations of UAV platforms. The following threat model and performance rationale formalize this design choice.

Threat Model. We consider three classes of adversaries:

1. A classical attacker without quantum capability aiming to recover session keys or forge traffic;
2. A future quantum-capable adversary capable of executing algorithms such as Shor's algorithm, which compromise ECC-based ECDH;
3. Implementation-level adversaries targeting specific primitive vulnerabilities (e.g., side-channel leakage or compromised random generation).

Resource constraints of UAVs—including limited RAM, bandwidth, and energy—also form part of the operational threat context.

Hybrid Rationale. The hybrid design provides compositional robustness by combining two independent entropy sources—ECDH and Kyber512. Even if one primitive becomes weakened, the session key remains secure due to domain-separated dual-entropy derivation (Equation (9)). From a transitional perspective, this approach allows gradual migration to post-quantum security while retaining the efficiency of ECC for low-latency control loops. The ephemeral ECDH phase ensures forward secrecy, while the Kyber512 KEM ensures long-term post-quantum resilience.

Performance and Communication Trade-offs. Quantitative evaluation demonstrates that ECDH (Curve25519) offers sub-millisecond latency on ESP32-class hardware, whereas Kyber512 exhibits $\approx 2.3\times$ higher runtime and energy cost. However, since the KEM operation occurs only once per session rather than per message, the amortized cost per protected frame becomes negligible.

In terms of communication overhead, a pure PQ-KEM per-message design would impose ≈ 768 B of ciphertext expansion per transmission, which is impractical for short UAV control packets. The hybrid approach introduces a one-time ≈ 1.5 kB payload during session setup—well within telemetry limits (≤ 256 kbps)—after which regular TinyJAMBU-encrypted frames incur no additional PQ overhead.

Security and Transition Advantages. Formally, the hybrid composition ensures that the adversary must simultaneously break both ECDH and KEM components to compromise confidentiality. This double-source resilience enhances forward secrecy and post-quantum readiness. Consequently, the proposed design achieves a balance between computational feasibility, bandwidth efficiency, and post-quantum robustness—an optimal trade-off for constrained UAV-GCS systems.

To highlight these trade-offs at the key exchange level, Table 4 provides a quantitative comparison between classical ECC, post-quantum KEM, and the proposed hybrid scheme under identical hardware conditions (ESP32, 240 MHz).

Table 4. Hybrid vs. Pure PQ-KEM Performance on UAV-class hardware (ESP32, 240 MHz).

Parameter	ECC (Curve25519)	PQ-KEM (Kyber512)	Hybrid (ECC + PQ-KEM)	Comment
Public key size	32 B	800 B	832 B	One-time session payload
Ciphertext size	32 B	768 B	800 B	Hybrid encapsulation output
Session key size	32 B	32 B	64 B	Dual-entropy concatenation
Key exchange time	0.9 ms	2.1 ms	3.0 ms	Amortized, single per session
Energy per key exchange	0.27 mJ	0.63 mJ	0.90 mJ	Measured via power profiler
Communications overhead per session	64 B	1.6 kB	1.5 kB	Within telemetry limits (≤ 256 kbps)
Post-quantum resilience	✗	✓	✓✓	Dual resilience: classical + quantum
Forward secrecy	✓	✗	✓	Provided by ECDH phase

This key exchange–level comparison focuses on session initialization overhead rather than per-frame encryption cost. It demonstrates that the hybrid approach maintains acceptable energy and latency profiles while providing dual-domain (classical + quantum) resilience. Hence, the methodology integrates analytical modeling, hybrid cryptographic composition, and empirical UAV-level profiling into a unified design framework.

4. Examples and Analysis

4.1. Implementation in Commercial Drones

The proposed Hybrid Lightweight Cryptographic Algorithm for Drones (HLCAD++) has been designed with the stringent requirements of resource-constrained aerial systems in mind. Such platforms, particularly commercial UAVs deployed in logistics, environmental monitoring, border surveillance, and real-time video streaming, demand security primitives that balance robustness, speed, and energy efficiency.

In a typical operational scenario, the Ground Control Station (GCS) initiates a session by performing a hybrid key exchange with the drone. First, an ephemeral Elliptic-Curve Diffie–Hellman (ECDH) handshake is performed to ensure forward secrecy. Simultaneously, a lattice-based Key Encapsulation Mechanism (KEM), e.g., Kyber, is executed to guarantee post-quantum resilience. Both outputs are combined via a dual-entropy HKDF, which produces a session key K_s . This key is then fed into TinyJAMBU AEAD, ensuring confidentiality, integrity, and authenticity of UAV telemetry, navigation coordinates, and payload data.

By separating the asymmetric (key exchange) and symmetric (encryption) phases, the scheme provides robust end-to-end security even in low-power drones, while maintaining throughput suitable for high-bandwidth applications such as video transmission.

To further verify the feasibility of the proposed cryptographic framework in resource-constrained UAV hardware, an experimental validation was conducted using a hardware-in-the-loop (HIL) environment, as detailed below.

Experimental Validation and Profiling Environment

While full-scale UAV flight trials are part of future work, the experimental validation of HLCAD++ was performed on a hardware-in-the-loop (HIL) setup representative of an

onboard UAV environment. The cryptographic operations were executed on an ESP32-WROOM-32D module (dual-core 240 MHz, 3.3 V) using the ESP-IDF v5.0 framework.

Latency and throughput were obtained via cycle-accurate hardware timers with microsecond resolution, and energy consumption was profiled using an INA219 precision power monitor (sampling at 1 kHz) connected in series with the SoC supply. The measurement covered initialization, key setup, encryption, and decryption phases under UAV-equivalent message sizes (64–512 bytes per telemetry frame) and control-loop intervals (20–50 ms).

This setup effectively reproduces the computational and energy behavior of UAV-class processors without the variability of open-air flight conditions, providing a deterministic baseline for algorithmic evaluation. Environmental and dynamic effects (e.g., temperature variation or vibration-induced timing drift) will be included in future hardware-in-loop flight validations once UAV integration resources become available.

4.2. Pseudocode for the Proposed Algorithm

Below is the extended pseudocode representation of the HLCAD++ workflow, divided into hybrid key exchange, key derivation, encryption, and decryption phases. The pseudocode explicitly integrates both ECC-based ECDH and lattice-based KEM, as well as the TinyJAMBU AEAD stage.

Initialization of Elliptic Curve Domain Parameters

```
def initialize_ecc():
    curve = define_curve(a, b, q)      #  $y^2 = x^3 + ax + b \pmod q$ 
    G = select_generator(curve)
    return curve, G
```

ECC Key Generation

```
def ecc_keygen(G):
    d = random_private_key()          # Private key
    P = d * G                          # Public key
    return d, P
```

Lattice-based KEM Key Generation

```
def kem_keygen():
    pk, sk = kem_generate()           # IND-CCA2 secure
    return pk, sk
```

Hybrid Key Exchange

```
def hybrid_key_exchange(ecc_priv, ecc_pub_peer, kem_sk, kem_ct_peer):
    K_ecc = ecc_priv * ecc_pub_peer    # ECDH shared point
    K_kem = kem_decapsulate(kem_sk, kem_ct_peer)
    return K_ecc, K_kem
```

Dual-Entropy HKDF (Extract-and-Expand)

```
def derive_session_key(K_ecc, K_kem, context_info):
    seed = concat(hash(K_ecc), hash(K_kem))
    K_s = HKDF_Expand(seed, context_info, length = 128) # 128-bit session key
    return K_s
```

```

# TinyJAMBU AEAD Encryption
def encrypt_data(K_s, plaintext, nonce, ad):
    C, T = tinyjambu_encrypt(K_s, nonce, ad, plaintext)
    return C, T

# TinyJAMBU AEAD Decryption
def decrypt_data(K_s, C, nonce, ad, T):
    plaintext = tinyjambu_decrypt(K_s, nonce, ad, C, T)
    return plaintext

# Main Hybrid Workflow
def hlcad_workflow(plaintext):
    # Initialization
    curve, G = initialize_ecc()

    # Drone side keys
    d_drone, P_drone = ecc_keygen(G)
    pk_kem_drone, sk_kem_drone = kem_keygen()

    # Base station keys
    d_base, P_base = ecc_keygen(G)
    pk_kem_base, sk_kem_base = kem_keygen()

    # Exchange public values
    kem_ct_drone = kem_encapsulate(pk_kem_base)
    kem_ct_base = kem_encapsulate(pk_kem_drone)

    # Derive shared secrets
    K_ecc_drone, K_kem_drone = hybrid_key_exchange(d_drone, P_base,
    sk_kem_drone, kem_ct_base)
    K_ecc_base, K_kem_base = hybrid_key_exchange(d_base, P_drone, sk_kem_base,
    kem_ct_drone)

    # Derive final symmetric session key
    K_s_drone = derive_session_key(K_ecc_drone, K_kem_drone, "Drone-Session")
    K_s_base = derive_session_key(K_ecc_base, K_kem_base, "Drone-Session")

    assert K_s_drone == K_s_base # Keys must match

    # Encrypt & decrypt
    nonce, ad = generate_nonce(), "Telemetry"
    C, T = encrypt_data(K_s_drone, plaintext, nonce, ad)
    recovered = decrypt_data(K_s_base, C, nonce, ad, T)
    return C, T, recovered

```

This extended pseudocode illustrates the novelty of HLCAD++, showing how two cryptographic domains (ECC and lattice-based KEM) are combined, and how the dual-entropy HKDF strengthens key derivation.

4.3. Cross-Domain Cryptographic Benchmarking

To demonstrate the overall comparative standing of HLCAD++ within both classical, lightweight, and post-quantum families, an expanded benchmarking summary is provided in Table 5.

Table 5. Comparative performance analysis of cryptographic methods in drone systems.

Algorithm	Type	Encryption Speed (Mbps)	Energy Consumption (J)	Key Length (bits)	Security Level	Post-Quantum Resilience	Suitability for Drones
AES	Symmetric	150	1.2	128/192/256	High	✗	Moderate
RSA	Asymmetric	20	5.8	2048	High	✗	Low
PRESENT	Lightweight	200	0.8	80	Moderate	✗	High
SPECK	Lightweight	230	0.6	64/128	High	✗	High
ASCON-128a	Lightweight (LWC winner)	215	0.7	128	High	✗	Very High
Kyber512	PQ-KEM	—	1.3	256 (shared key)	PQ (NIST L1)	✓	Moderate
NTRU-HRSS701	PQ-KEM	—	1.5	256 (shared key)	PQ (NIST L3)	✓	Moderate
HLCAD++ (Proposed)	Hybrid ECC + PQC + TinyJAMBU	225	0.5	128 (Sym.) + 256 (ECC/PQ)	High + PQ	✓✓	Very High

Unlike Tables 2 and 4, which focus on embedded-level metrics and key exchange efficiency, this table aggregates normalized performance indicators across diverse algorithmic paradigms (symmetric, asymmetric, and hybrid). Key metrics included encryption throughput, energy consumption, and suitability for drones.

The extended comparison in Table 5 demonstrates that while lightweight ciphers such as ASCON-128a and SPECK achieve high raw throughput (≈ 215 – 230 Mbps), they lack post-quantum resistance. Pure PQ-KEM schemes such as Kyber512 and NTRU, on the other hand, provide strong post-quantum security but incur significantly higher energy consumption and ciphertext overhead, limiting their applicability for real-time UAV operations.

HLCAD++ achieves throughput comparable to the fastest lightweight ciphers (≈ 225 Mbps) while maintaining superior energy efficiency (0.5 J) and dual-domain resilience through its hybrid ECC + PQC design. Even if one cryptographic primitive is weakened, the overall confidentiality and integrity remain intact due to the dual-entropy derivation.

Therefore, HLCAD++ offers a balanced trade-off between lightweight efficiency, post-quantum robustness, and practical deployability, making it an optimal candidate for next-generation UAV networks.

5. Conclusions

The accelerating integration of unmanned aerial vehicles (UAVs) into the Internet of Things (IoT) ecosystem highlights the urgent demand for cryptographic frameworks that deliver robust security without exceeding the stringent resource constraints of drones. Conventional cryptographic mechanisms, although mathematically strong, often impose prohibitive computational and energy costs that limit their practical deployment in lightweight airborne platforms.

This work has introduced HLCAD++, a novel hybrid lightweight cryptographic algorithm specifically tailored for UAV communication. By combining ephemeral Elliptic-Curve Diffie–Hellman (ECDH) with a post-quantum lattice-based KEM and strengthening the resulting entropy through a dual-source HKDF, the scheme ensures both forward secrecy and quantum resilience. For data confidentiality and integrity, the algorithm

employs the TinyJAMBU AEAD cipher: its low memory footprint and optimized round structure make it particularly well suited for constrained devices.

The proposed algorithm achieves the following key contributions:

1. **Robust Hybrid Security:** Dual-entropy session keys derived from both classical and post-quantum domains mitigate risks from emerging adversaries, including quantum-capable attackers.
2. **Computational Efficiency:** The integration of TinyJAMBU enables high-throughput authenticated encryption while reducing processing delays, supporting real-time video and telemetry streams.
3. **Energy Optimization:** The lightweight symmetric design ensures minimal energy consumption, thereby extending drone flight endurance—critical for missions in logistics, surveillance, and environmental monitoring.
4. **Scalability and Adaptability:** The modular hybrid design allows seamless deployment across diverse UAV applications, from small commercial quadcopters to swarms operating in cooperative networks.

Comparative performance analyses demonstrate that HLCAD++ achieves throughput on par with state-of-the-art lightweight ciphers such as SPECK while significantly improving energy efficiency. Crucially, it introduces post-quantum robustness absent from existing lightweight schemes, thereby future-proofing drone communications against rapidly evolving threats.

Beyond theoretical validation, HLCAD++ can be practically deployed on widely available embedded platforms such as ESP32- or STM32-based UAV controllers, as well as integrated into autopilot stacks like PX4 or ArduPilot. Potential deployment scenarios include secure telemetry in commercial drones (e.g., DJI Matrice or Parrot Anafi), cooperative swarm coordination in research platforms, and encrypted video streaming in inspection or search-and-rescue missions. These use cases demonstrate that the proposed scheme is not only theoretically sound but also readily adaptable to current-generation UAV ecosystems.

Looking ahead, future research will pursue three primary directions:

- Formal provable-security reductions under standard models (IND-CPA/IND-CCA2) and post-quantum assumptions.
- Extensive real-world deployment and benchmarking across heterogeneous drone hardware to validate scalability under operational constraints.
- Exploration of adaptive key rotation strategies and lightweight side-channel countermeasures, ensuring resilience against physical capture and implementation-level attacks.

By addressing the dual challenges of computational limitations and cryptographic longevity, this research lays the foundation for secure, efficient, and sustainable UAV integration within the next generation of IoT and cyber-physical ecosystems.

Author Contributions: Conceptualization, D.Z. and M.I.; methodology, D.Z. and M.I.; software, M.I.; validation, M.I., D.Z. and K.A.; formal analysis, S.R.; investigation, M.I.; resources, D.Z.; data curation, D.Z.; writing—original draft preparation, D.Z.; writing—review and editing, D.Z.; visualization, D.Z.; supervision, M.I. and S.R.; project administration, D.Z. and K.A.; funding acquisition, K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR27195331).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jameii, S.M.; Zamirnaddafi, R.; Rezaabakhsh, R. Internet of Flying Things Security: A Systematic Review. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7213. [CrossRef]
2. Guebzi, R.; Mami, S.; Chokmani, K. Drones in Precision Agriculture: A Comprehensive Review of Applications, Technologies, and Challenges. *Drones* **2024**, *8*, 686. [CrossRef]
3. Srivastava, S.K.; Seng, K.P.; Ang, L.M.; Pachas, A.N.A.; Lewis, T. Drone-Based Environmental Monitoring and Image Processing Approaches for Resource Estimates of Private Native Forest. *Sensors* **2022**, *22*, 7872. [CrossRef] [PubMed]
4. Konert, A.; Balcerzak, T. Military Autonomous Drones (UAVs)—From Fantasy to Reality: Legal and Ethical Implications. *Transp. Res. Procedia* **2021**, *59*, 292–299. [CrossRef]
5. Li, Y.; Liu, M.; Jiang, D. Application of Unmanned Aerial Vehicles in Logistics: A Literature Review. *Sustainability* **2022**, *14*, 14473. [CrossRef]
6. Nwaogu, J.M.; Yang, Y.; Chan, A.P.C.; Chi, H. Application of Drones in the Architecture, Engineering, and Construction (AEC) Industry. *Autom. Constr.* **2023**, *150*, 104827. [CrossRef]
7. Yahuza, M.; Idris, M.Y.I.; Ahmedy, I.B.; Wahab, A.W.A.; Nandy, T.; Noor, N.M.; Bala, A. Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges. *IEEE Access* **2021**, *9*, 57243–57270. [CrossRef]
8. Dedrone. Map of Worldwide Drone Incidents. Available online: <https://www.dedrone.com/resources/incidents/all> (accessed on 2 September 2025).
9. Mekdad, Y.; Aris, A.; Babun, L.; El Fergougui, A.; Conti, M.; Lazzeretti, R.; Uluagac, A.S. A Survey on Security and Privacy Issues of UAVs. *Comput. Netw.* **2023**, *224*, 109626. [CrossRef]
10. Hassija, V.; Chamola, V.; Gupta, V.; Jain, S.; Goyal, A.; Rodrigues, J.J.P.C. Fast, Reliable, and Secure Drone Communication: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2802–2836. [CrossRef]
11. Beigi, P.; Rajabi, M.S.; Aghakhani, S. An Overview of Drone Energy Consumption Factors and Models. In *Handbook of Smart Energy Systems*; Fathi, M., Zio, E., Pardalos, P.M., Eds.; Springer: Cham, Switzerland, 2022. [CrossRef]
12. Fazylova, A.; Alipbayev, K.; Myrzabekov, K.; Aden, A.; Iliev, T. The Aerodynamically Driven Orientation Control of a Solar Panel on an Aircraft with Numerical Simulation. *Drones* **2025**, *9*, 458. [CrossRef]
13. Shivgan, R.; Dong, Z. Energy-Efficient Drone Coverage Path Planning Using Genetic Algorithm. In Proceedings of the IEEE HPSR 2020, Newark, NJ, USA, 26–29 May 2020; pp. 1–6. [CrossRef]
14. Czachórski, T.; Gelenbe, E.; Kuaban, G.S.; Marek, D. Optimizing Energy Usage for an Electric Drone. In *Communications in Computer and Information Science*; Gelenbe, E., Jankovic, M., Kehagias, D., Marton, A., Vilmos, A., Eds.; Springer: Cham, Switzerland, 2022; Volume 1596, pp. 85–97. [CrossRef]
15. Boccadoro, P.; Striccoli, D.; Grieco, L.A. An Extensive Survey on the Internet of Drones. *Ad Hoc Netw.* **2021**, *122*, 102600. [CrossRef]
16. Küçükerdem, H.; Yilmaz, C.; Kahraman, H.T.; Sönmez, Y. Autonomous Control of Unmanned Aerial Vehicles: Applications, Requirements, Challenges. *Clust. Comput.* **2025**, *28*, 734. [CrossRef]
17. Yaacoub, J.-P.; Noura, H.; Salman, O.; Chehab, A. Security Analysis of Drones Systems: Attacks, Limitations, and Recommendations. *Internet Things* **2020**, *11*, 100218. [CrossRef] [PubMed]
18. Altawy, R.; Youssef, A.M. Security, Privacy, and Safety Aspects of Civilian Drones. *ACM Trans. Cyber-Physical Syst.* **2016**, *1*, 1–25. [CrossRef]
19. Harvey, M.R.; Kaiser, A.M.; Hoiness, G.W. Throughput of ASCON Compared with Popular IoT Encryption Algorithms. *Milit. Cyber Aff.* **2025**, *8*, 3. Available online: <https://digitalcommons.usf.edu/mca/vol8/iss1/3> (accessed on 5 October 2025).
20. FIPS 197; Advanced Encryption Standard (AES). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001.
21. Syaafaat, F. Title Unavailable. *Unikom Institutional Repository*, 2025. Available online: https://elibrary.unikom.ac.id/id/eprint/1149/13/22.10115361_FARHAN%20SYAFAAT_JURNAL%20DALAM%20BAHASA%20INGGRIS.pdf (accessed on 5 October 2025).
22. Ismael, H.M.; Al-Ta'i, Z.T.M. Privacy and Authenticity of Drone Communication Using ChaCha20 Algorithm. *AIP Conf. Proc.* **2023**, *2475*, 070015. [CrossRef]
23. Talaei Khoei, T.; Ghribi, E.; Prakash, R.; Kaabouch, N. A Performance Comparison of Encryption/Decryption Algorithms for UAV Swarm Communications. *Preprint* **2021**. [CrossRef]
24. Nyangaresi, V.O.; Jasim, H.M.; Mutlaq, K.A.-A.; Abduljabbar, Z.A.; Ma, J.; Abduljaleel, I.Q.; Honi, D.G. A Symmetric Key and Elliptic Curve Cryptography-Based Protocol for Message Encryption in Unmanned Aerial Vehicles. *Electronics* **2023**, *12*, 3688. [CrossRef]
25. Ozmen, M.O.; Yavuz, A.A. Dronecrypt—An Efficient Cryptographic Framework for Small Aerial Drones. In Proceedings of the MILCOM 2018—IEEE Military Communications Conference, Los Angeles, CA, USA, 29–31 October 2018; pp. 1–6. [CrossRef]

26. Cecchinato, N.; Toma, A.; Drioli, C.; Oliva, G.; Sechi, G.; Foresti, G.L. A Secure Real-Time Multimedia Streaming through Robust and Lightweight AES Encryption in UAV Networks for Operational Scenarios in Military Domain. *Procedia Comput. Sci.* **2022**, *205*, 50–57. [\[CrossRef\]](#)
27. Su, H.; Luo, W.; Zhang, X. Research on Secure Encryption Communication Method for Unmanned Aerial Vehicle System Based on Hybrid Encryption Algorithm. In Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering (EITCE '24), New York, NY, USA, 18–20 October 2024; pp. 477–480. [\[CrossRef\]](#)
28. Raj, A.; Balachandran, V. A Hybrid Encryption Framework Combining Classical, Post-Quantum, and QKD Methods. *arXiv* **2025**, arXiv:2509.10551. [\[CrossRef\]](#)
29. Chandu, Y.; Kumar, K.S.R.; Prabhukhanolkar, N.V.; Anish, A.N.; Rawal, S. Design and Implementation of Hybrid Encryption for Security of IoT Data. In Proceedings of the 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bengaluru, India, 17–19 August 2017; pp. 1228–1231. [\[CrossRef\]](#)
30. Yaacoub, E.; Abualsaud, K.; Mahmoud, M. Hybrid Encryption for Securing and Tracking Goods Delivery by Multipurpose Unmanned Aerial Vehicles in Rural Areas Using Cipher Block Chaining and Physical Layer Security. *Drones* **2024**, *8*, 111. [\[CrossRef\]](#)
31. Lin, L.; Shangguan, R.; Ge, H.; Liu, Y.; Zhou, Y.; Zhou, Y. Mutual Identity Authentication Based on Dynamic Identity and Hybrid Encryption for UAV–GCS Communications. *Drones* **2025**, *9*, 422. [\[CrossRef\]](#)
32. Alheeti, K.M.A.; Alarfaj, F.K.; Alreshoodi, M.; Almusallam, N.; Al Dosary, D. A Hybrid Security System for Drones Based on ICMetric Technology. *PLoS ONE* **2023**, *18*, e0282567. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Nanda, K.; Marukanti, S.R.; Harish, U.; Dheeraj, B.V.; Adoni, B.S.; Reddy, C.R.; Nanda, S. Evaluating Lightweight Asymmetric Cryptography for Secure Communication in Internet of Drones. In Proceedings of the 2025 Fourth International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 20–22 January 2025; pp. 875–879. [\[CrossRef\]](#)
34. Cahyo W, R.A.; Jonarta, H.; Clarissa, A.C.; Syafalni, I.; Sutisna, N.; Hasanuddin, M.O. Securing UAV Communication with TinyJAMBU Cryptography. In Proceedings of the 2023 International Conference on Electrical Engineering and Informatics (ICEEI), Bandung, Indonesia, 31 July–1 August 2023; pp. 1–6. [\[CrossRef\]](#)
35. Valikhanli, O.; Abdullayeva, F. Securing UAV Flight Data Using Lightweight Cryptography and Image Steganography. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 278–288. [\[CrossRef\]](#)
36. Yang, Y.; Dong, H.; Li, Z.; Xiao, S. LWED: Lightweight White-Box Encryption Communication System for Drones over CARX Algorithm. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 101727. [\[CrossRef\]](#)
37. Patel, A.; Cherukuri, A. Analysis of Lightweight Cryptography Algorithms for UAV-Networks. *arXiv* **2025**, arXiv:2504.04063. [\[CrossRef\]](#)
38. Sarkar, S.; Shafaei, S.; Jones, T.S.; Totaro, M.W. Secure Communication in Drone Networks: A Comprehensive Survey of Lightweight Encryption and Key Management Techniques. *Drones* **2025**, *9*, 583. [\[CrossRef\]](#)
39. Yun, J.; Yoon, S.; Kim, B.; Kang, Y. Applying Lightweight Cryptography to Enhance Drone RC Security. In Proceedings of the IEEE ICTC 2024, Jeju Island, Republic of Korea, 16–18 October 2024; pp. 1367–1368. [\[CrossRef\]](#)
40. Pizzolante, R.; Castiglione, A.; Palmieri, F.; Passaro, A.; Zaccagnino, R.; La Vecchia, S. Improving Drone Security in Smart Cities via Lightweight Cryptography. In *Computational Science and Its Applications—ICCSA 2023 Workshops. ICCSA 2023; Lecture Notes in Computer Science*, Gervasi, O., Murgante, B., Taniar, D., Aduhan, B.O., Braga, A.C., Garau, C., Stratigea, A., Eds.; Springer: Cham, Switzerland, 2023; Volume 14109, pp. 85–96. [\[CrossRef\]](#)
41. Zhang, S.; Liu, Y.; Han, Z.; Yang, Z. A Lightweight Authentication Protocol for UAVs Based on ECC Scheme. *Drones* **2023**, *7*, 315. [\[CrossRef\]](#)
42. Nikooghadam, M.; Amintoosi, H.; Islam, S.K.H.; Moghadam, M.F. A Provably Secure and Lightweight Authentication Scheme for Internet of Drones for Smart City Surveillance. *J. Syst. Archit.* **2021**, *115*, 101955. [\[CrossRef\]](#)
43. Al-Mekhlafi, Z.G.; Altmemi, J.M.H.; Al-Shareeda, M.A.; Al-Hchaimi, A.A.J.; Homod, R.Z.; Mohammed, B.A.; Alshammari, G.; Al-Dhlan, K.A.; Alrashdi, R.; Alkhabra, Y.A. ChebIoD: A Chebyshev Polynomial-Based Lightweight Authentication Scheme for Internet of Drones Environments. *Sci. Rep.* **2025**, *15*, 32897. [\[CrossRef\]](#)
44. Han, P.; Sui, A.; Wu, J. Lightweight Secure Communication Supporting Batch Authentication for UAV Swarm. *Drones* **2025**, *9*, 139. [\[CrossRef\]](#)
45. Xiao, Y.; Tao, Y. SLAKA-IoD: A Secure and Lightweight Authentication and Key Agreement Protocol for Internet of Drones. *Drones* **2024**, *8*, 374. [\[CrossRef\]](#)
46. Algarni, F.; Jan, S.U. PSLAPS-IoD: A Provable Secure and Lightweight Authentication Protocol for Securing Internet-of-Drones (IoD) Environment. *IEEE Access* **2024**, *12*, 45948–45960. [\[CrossRef\]](#)
47. Yu, S.; Das, A.K.; Park, Y.; Lorenz, P. SLAP-IoD: Secure and Lightweight Authentication Protocol Using Physical Unclonable Functions for Internet of Drones in Smart City Environments. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10374–10388. [\[CrossRef\]](#)

48. Wu, H.; Huang, T. *TinyJAMBU: A Family of Lightweight Authenticated Encryption Algorithms (Version 2)*; Nanyang Technological University: Singapore, 2021. Available online: <https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf> (accessed on 6 October 2025).
49. Bellare, M.; Rogaway, P. Code-Based Game-Playing Proofs and the Security of Triple Encryption. In *Advances in Cryptology—EUROCRYPT 2006 (LNCS 4004)*; Vaudenay, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 409–426. [CrossRef]
50. Sönmez, M.; McKay, K.; Chang, D.; Bassham, L.; Kang, J.; Waller, N.; Kelsey, J.; Hong, D. *Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process (NIST IR 8454)*; NIST: Gaithersburg, MD, USA, 2023. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.