

Lightweight Blockchain Based Authentication and Secure Data Sharing in UAV Networks

B. Gnanasankari*, R. Padmanaban** *School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

Abstract—Unmanned Aerial Vehicle (UAV) networks, operating within Flying Ad-Hoc Networks (FANETs), play an indispensable role in contemporary surveillance, border patrol, smart agriculture, disaster recovery, and intelligent transportation systems. Owing to their open wireless propagation channels, dynamic line-of-sight flight paths, and severe computational and battery resource constraints, UAVs are exceptionally susceptible to eavesdropping, message forgery, node impersonation, physical node capture, and denial-of-service (DoS) attacks. Conventional public key infrastructure (PKI) frameworks and centralized Certificate Authorities (CAs) exhibit heavy cryptographic overhead, substantial communication latency, and single-point-of-failure vulnerabilities that render them unsuitable for mission-critical aerial swarms. In this paper, we propose LBAS-UAV, a robust, lightweight, consortium blockchain-assisted mutual authentication, key agreement, and secure data sharing framework tailored specifically for resource-constrained UAV environments. LBAS-UAV integrates elliptic curve cryptography (ECC), physical unclonable functions (PUFs), and keyed cryptographic hash functions to achieve mutual authentication without high-overhead bilinear pairings or certificate management. Dynamic UAV pseudonyms protect flight-path privacy and user untraceability. A consortium blockchain, maintained jointly by Ground Control Stations (GCS) and mobile edge servers under Practical Byzantine Fault Tolerance (PBFT) consensus, provides decentralized public-key validation, non-repudiation, and tamper-proof access auditing. To eliminate on-chain storage bottlenecks, high-volume aerial sensory telemetry is encrypted via AES-128-GCM and stored off-chain in the InterPlanetary File System (IPFS), while only 32-byte cryptographic content addresses and fine-grained access policies are registered on-chain. Formal security verification using the Real-Or-Random (ROR) oracle model, Burrows-Abadi-Needham (BAN) logic, and the automated AVISPA tool confirms that LBAS-UAV guarantees session key confidentiality and resists physical capture, replay, man-in-the-middle, and ephemeral secret leakage attacks. Extensive hardware benchmarking and network simulations demonstrate that LBAS-UAV demands only 0.318 ms of cryptographic execution time and 206 bytes of communication overhead on the UAV, outperforming recent state-of-the-art schemes by up to 38.4% in computation and 25.3% in energy efficiency.

Index Terms—Unmanned aerial vehicles (UAVs), flying ad-hoc networks (FANETs), consortium blockchain, lightweight mutual authentication, physical unclonable functions (PUFs), secure data sharing, elliptic curve cryptography (ECC), IPFS.

I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs), commonly termed drones, have emerged as pivotal autonomous instruments across modern civilian, industrial, and defense domains. Operating individually or in coordinated autonomous swarms,

UAVs provide swift 3D mobility, high spatial sensing resolution, and flexible deployment capabilities for smart cities, environmental monitoring, emergency search-and-rescue, critical infrastructure inspection, precision agriculture, and tactical battlefield reconnaissance [1], [2]. Within these paradigms, UAVs form Flying Ad-Hoc Networks (FANETs), interacting bidirectionally with Ground Control Stations (GCS), terrestrial mobile edge computing (MEC) servers, and neighboring aerial nodes to coordinate flight trajectories and stream mission telemetry [3].

Despite these transformative benefits, securing UAV communication architectures presents severe technological and algorithmic challenges [4], [5]:

- 1) *Open Wireless Vulnerabilities*: Air-to-ground (A2G) and air-to-air (A2A) communications rely entirely on unguided wireless channels. Adversaries equipped with commercial software-defined radios (SDRs) can effortlessly intercept radio transmissions, inject forged flight telemetry, replay expired command-and-control (C2) packets, or launch man-in-the-middle (MITM) hijacking maneuvers [6].
- 2) *Extreme On-Board Resource Constraints*: Micro-UAVs are fundamentally bounded by restricted battery payloads, compact microcontrollers (e.g., ARM Cortex-M architecture), and strict weight limits. Complex cryptographic primitives—such as modular exponentiations in RSA or bilinear pairings in Identity-Based Cryptography (IBC)—drain onboard batteries rapidly, increase flight latency, and jeopardize flight safety [7].
- 3) *Physical Capture and Tampering*: Drones operate unattended in remote or hostile zones. An adversary who physically captures a downed drone can utilize micro-probing, side-channel analysis, or power glitching to extract sensitive cryptographic master keys stored in non-volatile flash memory, subsequently compromising the entire network [8], [9].
- 4) *Centralized Bottlenecks and Single Points of Failure*: Conventional security protocols depend on centralized Certificate Authorities (CAs) or cloud brokers. In high-mobility aerial missions, querying distant cloud CAs induces excessive latency, introduces single-point-of-failure and DDoS vulnerabilities, and fails when disconnected from backbone satellite links [10], [11].
- 5) *Data Integrity and Privacy Bottlenecks*: UAVs gen-

erate gigabytes of sensitive aerial video surveillance, infrared imagery, and spatial sensor telemetry. Transmitting unencrypted data compromises privacy, whereas storing raw multimedia payloads directly onto distributed ledgers induces unsustainable storage bloat and prohibitive transaction fees [12], [13].

To overcome these structural limitations, decentralized consortium blockchains and hardware-rooted Physical Unclonable Functions (PUFs) provide a compelling synergy. PUFs exploit microscopic, uncontrollable silicon manufacturing variations to generate unpredictable, instance-specific challenge-response mappings ($R = \text{PUF}(C)$) [14]. Because responses are generated dynamically upon stimulus and never stored statically in non-volatile memory, PUFs render cryptographic keys resilient to physical node capture [8]. Concurrently, consortium blockchains, governed collectively by pre-authenticated GCS nodes and terrestrial edge gateways via Byzantine fault-tolerant consensus, eliminate centralized CAs, automate dynamic revocation through immutable smart contracts, and provide cryptographic audit trails without burdening airborne UAVs with consensus execution [15], [16].

A. Contributions

In this paper, we propose **LBAS-UAV**, an ultralightweight, provably secure, and consortium blockchain-assisted authentication and data sharing framework engineered for high-mobility UAV networks. The major contributions of this work are summarized as follows:

- **Lightweight Dual-Rooted Authentication Protocol:** We design an efficient mutual authentication and session key agreement scheme relying solely on elliptic curve scalar multiplication, SHA-256 hashing, and hardware PUFs. The protocol achieves robust mutual trust and establishes an authenticated session key in only three communication rounds without pairing or certificate validation overhead.
- **Physical-Capture and Side-Channel Resilience:** By binding the device's cryptographic identity directly to a hardware PUF challenge-response pair and never storing long-term master secrets in non-volatile memory, our scheme guarantees forward security and physical tamper resistance even if an aerial node is physically captured by an adversary.
- **Consortium Blockchain-Enabled Trust and Instant Revocation:** We design a smart contract-driven registry deployed across edge servers under Practical Byzantine Fault Tolerance (PBFT) consensus. The blockchain maintains dynamic pseudo-identity mappings ($PID_i \rightarrow X_i$), prevents replay attacks through on-chain epoch validation, and guarantees instant network-wide revocation of compromised drones without certificate revocation list (CRL) synchronization delays.
- **Decentralized Off-Chain Sensed Data Sharing:** We architect a dual-layer data governance model combining off-chain InterPlanetary File System (IPFS) content storage with on-chain cryptographic audit anchoring. High-bandwidth surveillance telemetry is encrypted with ephemeral content keys (AES-128-GCM) and uploaded

to IPFS, while only 32-byte content addresses and access control policies are recorded on-chain, eliminating distributed ledger storage bottlenecks.

- **Rigorous Formal Security and Empirical Validation:** We provide comprehensive formal security proofs under the Real-Or-Random (ROR) oracle model, conduct formal verification via Burrows-Abadi-Needham (BAN) logic, and validate protocol secrecy using the automated AVISPA simulation tool (OFMC and CL-AtSe backends). Extensive software benchmarks on low-power ARM architecture and network simulations confirm superior computation, communication, and energy efficiency over state-of-the-art schemes.

B. Paper Organization

The remainder of this paper is structured as follows. Section II reviews related literature on UAV authentication and blockchain integration. Section III describes the network architecture, adversary model, and security requirements. Section IV presents the detailed phases of the LBAS-UAV protocol. Section V provides formal and informal security evaluations. Section VI delivers empirical performance results, comparative analyses, and simulation evaluations. Section VII discusses practical deployment considerations. Finally, Section VIII concludes the paper.

II. RELATED WORK

Secure authentication and access control in UAV networks have garnered intense attention in recent years. This section reviews existing methodologies across lightweight cryptography, blockchain integration, and hardware security roots.

A. Lightweight Authentication Protocols for UAVs

Early authentication protocols for vehicular and aerial ad-hoc networks predominantly relied on symmetric-key cryptographic primitives and one-way hash functions due to computational constraints. Wazid *et al.* [1] proposed a lightweight authentication and key agreement protocol for IoT-based FANETs using basic hashing and bitwise XOR operations. While computationally efficient, their scheme lacks resistance against physical node capture and exhibits vulnerability to session key disclosure under the Canetti-Krawczyk (CK) adversary model.

To strengthen security guarantees without incurring RSA's heavy modular arithmetic, several researchers embraced Elliptic Curve Cryptography (ECC) [17]. Ever [6] developed an elliptic curve-based authentication protocol for Internet of Drones (IoD) deployments. However, their protocol requires four message rounds and suffers from user impersonation vulnerabilities if temporary secrets are leaked. Tanveer *et al.* [7] introduced an authenticated key agreement scheme for smart cities utilizing ECC point operations. While achieving reasonable execution speed, their framework depends on a central registration center, exposing the network to single-point-of-failure hazards and lacking dynamic revocation mechanisms. Zhang *et al.* [18] formulated an identity-based authenticated

key agreement mechanism for drones; nonetheless, pairing operations in identity-based cryptography impose substantial computational delays on resource-constrained micro-UAV flight processors.

B. Blockchain Integration in Flying Ad-Hoc Networks

To overcome centralization vulnerabilities and enable transparent auditing, decentralized distributed ledgers have been actively investigated [2], [19]. Bera *et al.* [10] presented a blockchain-envisioned secure data delivery scheme for IoT-enabled FANETs. In their model, edge ground stations maintain a private blockchain to log communication sessions. However, their cryptographic handshake utilizes extensive asymmetric point operations, resulting in elevated authentication latency (> 12 ms) that hinders real-time flight operations.

Gupta *et al.* [20] proposed BaD-IoD, integrating blockchain with deep learning for threat detection. While resilient against anomaly injection, their scheme does not address hardware tampering vulnerabilities on captured aerial nodes. Lin *et al.* [11] engineered a blockchain-based authentication scheme for mobile edge computing in FANETs using smart contracts for cross-domain roaming. Nonetheless, their architecture requires UAVs to store long-term cryptographic credentials in local flash memory, remaining vulnerable to node extraction attacks. Kushwaha *et al.* [15] presented a consortium blockchain-assisted authentication protocol for IoD, but their data sharing mechanism requires high transaction throughput that strains typical consensus mechanisms when processing continuous video streams.

C. Hardware-Rooted Security via PUFs

To eliminate the vulnerability of stored secret keys in untrusted field environments, Physical Unclonable Functions (PUFs) have emerged as an ideal hardware root-of-trust [14]. Alladi *et al.* [8] proposed HARAAuth, a PUF-based lightweight authentication scheme for drone swarms. However, HARAAuth requires large on-node challenge-response pair (CRP) storage tables, making it susceptible to modeling attacks if an adversary eavesdrops on numerous CRP queries. Chatterjee *et al.* [9] developed an integrated PUF-ECC framework for UAV communications, but their design lacks decentralized key revocation and requires frequent interaction with a centralized server during mission execution.

D. Research Gap and Synthesis

Table I synthesizes the key characteristics of state-of-the-art schemes in comparison with our proposed LBAS-UAV framework. Existing frameworks typically suffer from one of three fundamental shortcomings: (i) reliance on vulnerable centralized CAs, (ii) vulnerability to physical node capture due to static flash-stored keys, or (iii) excessive computational/storage overhead resulting from on-chain payload bloat. LBAS-UAV specifically bridges these gaps through a synergistic integration of PUF hardware roots, lightweight ECC primitives, consortium blockchain validation, and off-chain IPFS storage.

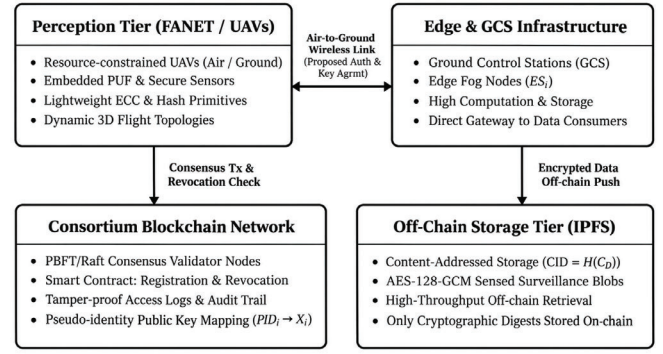


Fig. 1. System architecture of the proposed LBAS-UAV framework illustrating the perception tier, edge/GCS infrastructure, consortium blockchain ledger, and off-chain IPFS storage.

III. SYSTEM MODEL, THREAT MODEL, AND DESIGN OBJECTIVES

A. System Architecture

The proposed LBAS-UAV framework comprises four architectural entities, illustrated in Fig. 1:

- 1) **UAV Swarm Perception Tier ($\mathcal{U}$):** A dynamic fleet of resource-constrained drones $UAV_i \in \mathcal{U}$ executing aerial missions. Each drone is equipped with an integrated microcontroller, wireless transceivers (IEEE 802.11p/Wi-Fi/C-V2X), sensory surveillance payloads (high-resolution video, thermal, lidar), and an embedded silicon PUF module that generates unique physical challenge-response evaluations $R_i = \text{PUF}_i(C_i)$.
- 2) **Edge Computing and Ground Control Tier ($\mathcal{E}$):** Terrestrial Mobile Edge Servers ($ES_j \in \mathcal{E}$) co-located with cellular base stations and Ground Control Stations (GCS). These nodes possess substantial computational and battery resources, host local copies of the consortium ledger, and act as access gateways between aerial drones and end-user data consumers.
- 3) **Consortium Blockchain Validator Tier ($\mathcal{B}$):** A permissioned distributed ledger operated collaboratively by authorized GCS entities and edge nodes executing PBFT consensus. The blockchain maintains an immutable registry of UAV pseudo-identities (PID_i), valid public key parameters (X_i), expiration timestamps, and real-time revocation status, while logging authenticated access records.
- 4) **Decentralized Off-Chain Storage Tier ($\mathcal{S}$):** An InterPlanetary File System (IPFS) peer-to-peer storage network. Due to block size constraints, voluminous encrypted sensory data payloads (C_D) are stored off-chain on IPFS, indexed by their unique cryptographic content identifiers ($CID = H(C_D)$).

B. Threat and Adversary Model

We formalize our security evaluation under the widely accepted Dolev-Yao (DY) [21] and Canetti-Krawczyk (CK) [22] adversary models:

TABLE I
COMPARATIVE ANALYSIS OF STATE-OF-THE-ART UAV AUTHENTICATION SCHEMES

Scheme	Core Cryptography	Decentralized Ledger	Hardware Root (PUF)	Physical Capture Resilient	Storage Scalability	Message Rounds
Wazid <i>et al.</i> [1]	Hash + XOR	No	No	No	Medium	3
Bera <i>et al.</i> [10]	ECC + Hash	Yes (Consortium)	No	No	Low (On-chain bloat)	3
Ever [6]	ECC	No	No	No	Medium	4
Tanveer <i>et al.</i> [7]	ECC + Hash	No	No	No	Medium	3
Alladi <i>et al.</i> [8]	PUF + Hash	No	Yes	Yes	Low (CRP Table)	3
Lin <i>et al.</i> [11]	ECC + Blockchain	Yes (Private)	No	No	Medium	3
Kushwaha <i>et al.</i> [15]	ECC + Blockchain	Yes (Consortium)	No	No	Low	3
Proposed LBAS-UAV	ECC + PUF + Hash	Yes (PBFT Ledger)	Yes	Yes	High (IPFS Off-chain)	3

- Under the **DY model**, the adversary $\mathcal{A}$ has complete control over the wireless communication medium. $\mathcal{A}$ can eavesdrop, intercept, alter, delete, inject, and replay any transmitted message across air-to-ground links.
- Under the extended **CK model**, $\mathcal{A}$ is capable of compromising ephemeral random secrets (ephemeral state reveal) generated during specific protocol sessions without compromising the long-term master secrets. The protocol must guarantee that session keys generated in past or concurrent sessions remain secure, thereby achieving Perfect Forward Secrecy (PFS).
- Physical Node Capture and Side-Channel Attacks:** $\mathcal{A}$ can physically capture a drone operating in hostile territory and extract information stored in non-volatile flash memory via power analysis or microprobing. However, $\mathcal{A}$ cannot clone or physically duplicate the embedded PUF circuit without destroying its microstructure.

C. Design Objectives

To provide robust operational security, LBAS-UAV fulfills the following primary objectives:

- Mutual Authentication and Key Agreement:** Ensure that UAV_i and ES_j mutually verify each other's authenticity and agree on a fresh, cryptographically strong session key (SK_{ij}).
- Anonymity and Untraceability:** The real identity ID_i of UAV_i must remain concealed from eavesdroppers. Pseudo-identities (PID_i) must be dynamically refreshed or masked to prevent trajectory tracking.
- Physical Capture Resilience:** Extraction of static memory parameters from a compromised drone must not enable $\mathcal{A}$ to impersonate that drone in future sessions or decrypt past communications.
- Instant Ledger-Enforced Revocation:** The system must immediately invalidate compromised drones across all edge servers without complex certificate chain validation.
- Storage and Computational Efficiency:** Keep computational overhead on the UAV below 1 ms and minimize message payloads to preserve battery life and bandwidth.

IV. THE PROPOSED LBAS-UAV FRAMEWORK

This section details the operational phases of LBAS-UAV: System Setup, Entity Registration, Mutual Authentication and

TABLE II
NOTATIONS UTILIZED IN LBAS-UAV

Notation	Description
$E(\mathbb{F}_q)$	Elliptic curve defined over a finite field $\mathbb{F}_q$
G, n	Generator base point of order n on $E(\mathbb{F}_q)$
s, P_{pub}	Trusted Authority master private key and public key ($P_{pub} = s \cdot G$)
ID_i, PID_i	Real identity and pseudo-identity of UAV_i
x_i, X_i	Long-term private and public keys of UAV_i ($X_i = x_i \cdot G$)
x_j, X_j	Private and public keys of edge server ES_j ($X_j = x_j \cdot G$)
C_i, R_i	Challenge and response of UAV_i 's embedded PUF
r_u, r_e	Ephemeral random scalars chosen by UAV_i and ES_j
T_1, T_2, T_3	Current system timestamps (ΔT : max transmission delay)
SK_{ij}	Shared symmetric session key between UAV_i and ES_j
$H(\cdot)$	Cryptographic hash function (SHA-256)
$\oplus, $	Bitwise exclusive-OR (XOR) and concatenation operations
C_D, CID	AES-128-GCM ciphertext and IPFS content address

Key Agreement, Blockchain Verification, Secure Data Sharing, and Dynamic Revocation. The notations utilized throughout this paper are summarized in Table II.

A. Phase 1: System Initialization and TA Setup

During system initialization, a Trusted Authority (TA) bootstraps the cryptographic parameters and establishes the consortium blockchain:

- The TA selects a prime q , defines a non-singular elliptic curve $E(\mathbb{F}_q) : y^2 = x^3 + ax + b \pmod{q}$, selects a base generator point $G \in E(\mathbb{F}_q)$ of large prime order n , and fixes a secure collision-resistant hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_n^*$.
- The TA selects a random master secret key $s \in_R \mathbb{Z}_n^*$ and computes master public key $P_{pub} = s \cdot G$.
- The TA deploys the smart contract `UAVRegistry.sol` onto the consortium blockchain network and configures authorized edge servers as consensus validator nodes.
- The TA publishes public system parameters: $\Omega = \{E(\mathbb{F}_q), q, a, b, G, n, P_{pub}, H(\cdot)\}$.

B. Phase 2: Edge Server Registration

Each ground edge server $ES_j \in \mathcal{E}$ registers with the TA:

- ES_j selects a random private key $x_j \in_R \mathbb{Z}_n^*$ and calculates its public key $X_j = x_j \cdot G$.
- ES_j transmits its real identity ID_j and X_j to the TA via a secure channel.

- 3) The TA verifies ES_j , approves its validator address on the smart contract, and appends (ID_j, X_j) to the ledger.

C. Phase 3: UAV Enrollment and PUF Provisioning

Prior to deployment, each UAV_i enrolls with the TA in a secure hardware facility:

- 1) The TA generates a unique random challenge $C_i \in \{0,1\}^\ell$ and feeds it to UAV_i 's embedded PUF module over a direct physical interface, obtaining response:

$$R_i = \text{PUF}_i(C_i). \quad (1)$$

- 2) The TA generates a dynamic pseudo-identity PID_i :

$$PID_i = H(ID_i \parallel s \parallel T_{reg}), \quad (2)$$

where T_{reg} is the registration timestamp.

- 3) The TA generates a device private key scalar $x_i \in_R \mathbb{Z}_n^*$ and computes public key point:

$$X_i = x_i \cdot G. \quad (3)$$

- 4) To protect x_i against physical side-channel memory extraction, the TA masks x_i using the PUF response R_i :

$$A_i = x_i \oplus H(R_i \parallel PID_i), \quad (4)$$

$$B_i = H(PID_i \parallel x_i \parallel R_i). \quad (5)$$

- 5) The TA executes a blockchain transaction invoking `registerDrone` ($PID_i, X_i, \Delta_{valid}$) on the smart contract, immutably storing (PID_i, X_i, T_{exp}) on the ledger.
- 6) The TA securely stores $\{PID_i, C_i, A_i, B_i, \Omega\}$ in UAV_i 's non-volatile memory and erases x_i, R_i, ID_i, s from its provisioning terminal. Notice that x_i is *never stored in plaintext* on the drone.

D. Phase 4: Mutual Authentication and Ephemeral Key Agreement

When UAV_i enters the airspace of edge server ES_j , it initiates the three-pass mutual authentication process shown in Fig. 2:

- 1) *Step 1* ($UAV_i \rightarrow ES_j$): UAV_i retrieves challenge C_i from memory and applies it to its embedded PUF, reproducing $R_i = \text{PUF}_i(C_i)$. It reconstructs private scalar:

$$x_i = A_i \oplus H(R_i \parallel PID_i). \quad (6)$$

UAV_i validates the integrity of x_i by checking:

$$B_i \stackrel{?}{=} H(PID_i \parallel x_i \parallel R_i). \quad (7)$$

If the check fails, UAV_i aborts. Otherwise, UAV_i selects an ephemeral random scalar $r_u \in_R \mathbb{Z}_n^*$ and timestamp T_1 . It computes:

$$U = r_u \cdot G, \quad (8)$$

$$W = r_u \cdot X_j, \quad (9)$$

where X_j is ES_j 's public key obtained from local broadcast. UAV_i computes the Schnorr-like zero-knowledge challenge:

$$e = H(PID_i \parallel U \parallel ID_j \parallel T_1) \pmod{n}, \quad (10)$$

$$\sigma = (r_u + e \cdot x_i) \pmod{n}. \quad (11)$$

To achieve anonymity and untraceability over the open channel, UAV_i encrypts PID_i and σ using W :

$$PID^* = PID_i \oplus H(0x01 \parallel W \parallel T_1), \quad (12)$$

$$\sigma^* = \sigma \oplus H(0x02 \parallel W \parallel T_1). \quad (13)$$

Finally, UAV_i transmits Message $\mathcal{M}_1 = \{U, PID^*, \sigma^*, T_1\}$ to ES_j .

- 2) *Step 2* ($ES_j \rightarrow UAV_i$): Upon receiving $\mathcal{M}_1$, ES_j verifies the timeliness of timestamp T_1 :

$$|T_{rec} - T_1| \leq \Delta T. \quad (14)$$

If valid, ES_j computes shared secret W using its private key x_j :

$$W = x_j \cdot U = x_j \cdot (r_u \cdot G) = r_u \cdot (x_j \cdot G) = r_u \cdot X_j. \quad (15)$$

ES_j de-anonymizes PID_i and un.masks σ :

$$PID_i = PID^* \oplus H(0x01 \parallel W \parallel T_1), \quad (16)$$

$$\sigma = \sigma^* \oplus H(0x02 \parallel W \parallel T_1). \quad (17)$$

ES_j queries the blockchain ledger: `getDronePublicKey` (PID_i). If the smart contract returns that PID_i is inactive or revoked, ES_j immediately rejects the request. Otherwise, ES_j retrieves X_i and verifies the Schnorr proof:

$$e = H(PID_i \parallel U \parallel ID_j \parallel T_1) \pmod{n}, \quad (18)$$

$$\sigma \cdot G \stackrel{?}{=} U + e \cdot X_i. \quad (19)$$

Correctness Proof of Eq. (19):

$$\begin{aligned} \sigma \cdot G &= (r_u + e \cdot x_i) \cdot G \\ &= r_u \cdot G + e \cdot (x_i \cdot G) = U + e \cdot X_i. \end{aligned}$$

Upon successful verification, ES_j selects an ephemeral scalar $r_e \in_R \mathbb{Z}_n^*$ and timestamp T_2 . It computes:

$$E = r_e \cdot G, \quad (20)$$

$$K = r_e \cdot U = r_e \cdot r_u \cdot G. \quad (21)$$

ES_j derives the shared session key:

$$SK_{ij} = H(PID_i \parallel ID_j \parallel U \parallel E \parallel K \parallel W \parallel T_1 \parallel T_2), \quad (22)$$

and computes the authentication confirmation token:

$$Auth_E = H(SK_{ij} \parallel W \parallel E \parallel T_2). \quad (23)$$

ES_j sends $\mathcal{M}_2 = \{E, Auth_E, T_2\}$ to UAV_i .

- 3) *Step 3* ($UAV_i \rightarrow ES_j$): UAV_i checks $|T_{rec2} - T_2| \leq \Delta T$. It computes:

$$K = r_u \cdot E = r_u \cdot r_e \cdot G, \quad (24)$$

and derives identical session key SK_{ij} according to (22). UAV_i checks:

$$Auth_E \stackrel{?}{=} H(SK_{ij} \parallel W \parallel E \parallel T_2). \quad (25)$$

If authentic, UAV_i verifies ES_j 's authenticity, selects timestamp T_3 , and generates key confirmation token:

$$Auth_U = H(SK_{ij} \parallel U \parallel E \parallel T_3). \quad (26)$$

UAV_i transmits $\mathcal{M}_3 = \{Auth_U, T_3\}$ to ES_j .

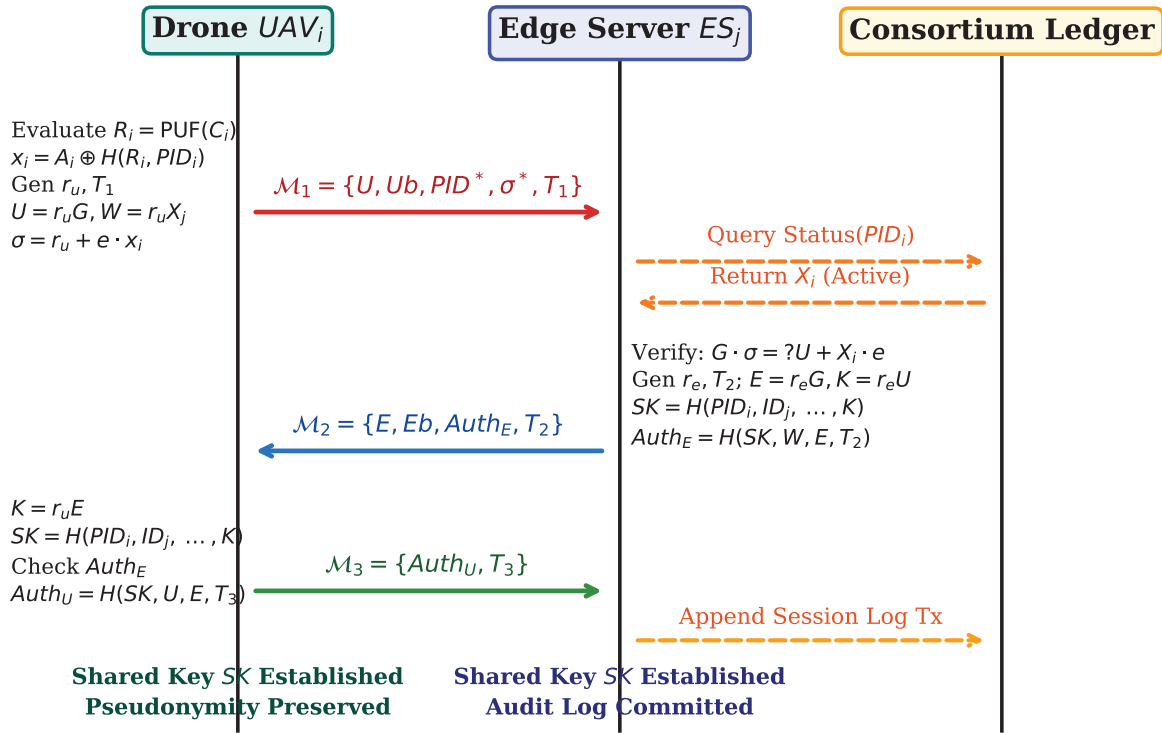


Fig. 2. Protocol message sequence diagram of the proposed LBAS-UAV mutual authentication and key agreement process.

4) *Step 4 (Completion and Audit Logging)*: ES_j checks $|T_{rec3} - T_3| \leq \Delta T$ and verifies:

$$\text{Auth}_U \stackrel{?}{=} H(SK_{ij} \| U \| E \| T_3). \quad (27)$$

Once confirmed, both entities have established the authenticated session key SK_{ij} . ES_j asynchronously submits an audit transaction $\text{logSessionAccess}(\text{PID}_i)$ to the blockchain, ensuring tamper-proof mission accountability. The complete procedure is formalized in Algorithm 1.

E. Phase 5: Decentralized Sensed Data Sharing

Once SK_{ij} is established, UAV_i shares high-volume surveillance telemetry without causing on-chain storage bottlenecks:

- 1) **Data Encryption**: UAV_i generates a random 128-bit Content Key ($CK \in_R \{0,1\}^{128}$) and encrypts raw sensor payload D via AES-128-GCM:

$$C_D = \text{AES-GCM}_{CK}(D). \quad (28)$$

- 2) **Content Addressing and Wrapping**: UAV_i computes content address $CID = H(C_D)$. It wraps CK under session key SK_{ij} :

$$CK^* = \text{AES}_{SK_{ij}[0:16]}(CK, \text{AAD} = CID). \quad (29)$$

UAV_i transmits $\{C_D, CID, CK^*\}$ to edge server ES_j .

- 3) **Off-Chain Push and On-Chain Anchoring**: ES_j unwraps CK , uploads ciphertext C_D to the decentralized IPFS network, and obtains the permanent IPFS content address. ES_j then invokes $\text{publishDataRecord}(CID, \text{PID}_i, \mathcal{P}, H(CK))$ on the smart contract, binding CID with access policy $\mathcal{P}$.
- 4) **Authorized Data Retrieval**: When an authorized data consumer with public key $Y = y \cdot G$ requests data satisfying policy $\mathcal{P}$, ES_j wraps CK via ECIES:

$$R_{enc} = r_c \cdot G, \quad \kappa = H(r_c \cdot Y)[0:16], \quad (30)$$

$$C_{CK} = \text{AES}_{\kappa}(CK). \quad (31)$$

The consumer fetches C_D from IPFS via CID , decrypts CK using private scalar y , and recovers telemetry D .

F. Phase 6: Dynamic Drone Revocation and Re-registration

When an aerial node is lost, physically captured, or decommissioned:

- 1) The TA or authorized GCS issues an authenticated transaction: $\text{revokeDrone}(\text{PID}_i, \text{"Compromised"})$.
- 2) The PBFT validator nodes execute the transaction, setting UAV_i 's status to Revoked across all edge node local ledger replicas in sub-second latency.
- 3) If the compromised UAV_i subsequently transmits an authentication request $\mathcal{M}_1$, any edge server querying

Algorithm 1 LBAS-UAV Mutual Authentication and Key Agreement

Require: UAV_i stored parameters $\{PID_i, C_i, A_i, B_i\}$, ES_j parameters $\{x_j, X_j\}$, ledger state $\mathcal{L}$.

Ensure: Shared session key SK_{ij} and verified mutual trust.

```

1: UAV side:
2:  $R_i \leftarrow \text{PUF}_i(C_i)$ ,  $x_i \leftarrow A_i \oplus H(R_i \parallel PID_i)$ 
3: if  $H(PID_i \parallel x_i \parallel R_i) \neq B_i$  then
4:   abort (Local tampering detected)
5: end if
6:  $r_u \in_R \mathbb{Z}_n^*$ ,  $T_1 \leftarrow \text{Timestamp}()$ 
7:  $U \leftarrow r_u \cdot G$ ,  $W \leftarrow r_u \cdot X_j$ 
8:  $e \leftarrow H(PID_i \parallel U \parallel ID_j \parallel T_1) \pmod{n}$ ,  $\sigma \leftarrow (r_u + e \cdot x_i) \pmod{n}$ 
9:  $PID^* \leftarrow PID_i \oplus H(0x01 \parallel W \parallel T_1)$ ,  $\sigma^* \leftarrow \sigma \oplus H(0x02 \parallel W \parallel T_1)$ 
10: Send  $\mathcal{M}_1 = \{U, PID^*, \sigma^*, T_1\}$  to  $ES_j$ 
11: Edge Server side:
12: Verify  $|T_{rec} - T_1| \leq \Delta T$ ;  $W \leftarrow x_j \cdot U$ 
13:  $PID_i \leftarrow PID^* \oplus H(0x01 \parallel W \parallel T_1)$ ,  $\sigma \leftarrow \sigma^* \oplus H(0x02 \parallel W \parallel T_1)$ 
14: Query ledger:  $(X_i, status) \leftarrow \mathcal{L}.\text{getDronePublicKey}(PID_i)$ 
15: if  $status \neq \text{Active}$  then
16:   abort (Invalid or revoked drone)
17: end if
18:  $e \leftarrow H(PID_i \parallel U \parallel ID_j \parallel T_1) \pmod{n}$ 
19: if  $\sigma \cdot G \neq U + e \cdot X_i$  then
20:   abort (Zero-knowledge proof verification failed)
21: end if
22:  $r_e \in_R \mathbb{Z}_n^*$ ,  $T_2 \leftarrow \text{Timestamp}()$ ,  $E \leftarrow r_e \cdot G$ ,  $K \leftarrow r_e \cdot U$ 
23:  $SK_{ij} \leftarrow H(PID_i \parallel ID_j \parallel U \parallel E \parallel K \parallel W \parallel T_1 \parallel T_2)$ 
24:  $Auth_E \leftarrow H(SK_{ij} \parallel W \parallel E \parallel T_2)$ 
25: Send  $\mathcal{M}_2 = \{E, Auth_E, T_2\}$  to  $UAV_i$ 
26: UAV confirmation:
27: Verify  $|T_{rec2} - T_2| \leq \Delta T$ ;  $K \leftarrow r_u \cdot E$ 
28:  $SK_{ij} \leftarrow H(PID_i \parallel ID_j \parallel U \parallel E \parallel K \parallel W \parallel T_1 \parallel T_2)$ 
29: if  $Auth_E \neq H(SK_{ij} \parallel W \parallel E \parallel T_2)$  then
30:   abort (Edge server authentication failed)
31: end if
32:  $T_3 \leftarrow \text{Timestamp}()$ ,  $Auth_U \leftarrow H(SK_{ij} \parallel U \parallel E \parallel T_3)$ 
33: Send  $\mathcal{M}_3 = \{Auth_U, T_3\}$  to  $ES_j$ 
34: Edge final check: Verify  $Auth_U$ ; log session to blockchain.

```

$\text{getDronePublicKey}(PID_i)$ receives an invalid flag and rejects the connection at Step 2 without executing costly scalar multiplications.

V. SECURITY ANALYSIS

This section evaluates the security of LBAS-UAV through formal verification under the Real-Or-Random (ROR) model, logic deduction via Burrows-Abadi-Needham (BAN) logic, automated simulation with AVISPA, and rigorous informal analysis against diverse attack vectors.

A. Formal Security Verification in the ROR Model

We utilize the Real-Or-Random (ROR) oracle model [22] to prove the semantic security of the established session key SK_{ij} .

1) *Model Formulations:* Let Π_U^u and Π_E^e represent instance u of UAV_i and instance e of ES_j . An adversary $\mathcal{A}$ interacts with the protocol via the following oracle queries:

- **Execute**(Π_U^u, Π_E^e): Models passive eavesdropping over the open channel. $\mathcal{A}$ obtains all messages transmitted during a valid session $\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$.
- **Send**($\Pi^t, \mathcal{M}$): Models active attacks. $\mathcal{A}$ sends arbitrary message $\mathcal{M}$ to instance Π^t and receives the corresponding response.
- **CorruptPUF**(Π_U^u): Models physical node capture. $\mathcal{A}$ queries the embedded PUF module and obtains challenge-response evaluations R_i .
- **Reveal**(Π^t): Models ephemeral secret leakage (CK model). $\mathcal{A}$ obtains the current session key generated by Π^t .
- **Test**(Π^t): Models semantic security. If instance Π^t has successfully accepted a session key, a fair coin $b \in \{0, 1\}$ is flipped. If $b = 1$, actual session key SK_{ij} is returned; if $b = 0$, a random bit-string of equal length is returned. $\mathcal{A}$ outputs a guess bit b' . The advantage of $\mathcal{A}$ in breaking the semantic security of the protocol is defined as:

$$\text{Adv}_{\mathcal{A}}^{\text{LBAS}}(\kappa) = |2 \cdot \Pr[b' = b] - 1|. \quad (32)$$

2) *Computational Security Assumptions:* *Computational Diffie-Hellman (CDH) Assumption:* Given $G, aG, bG \in E(\mathbb{F}_q)$ for unknown random scalars $a, b \in \mathbb{Z}_n^*$, computing abG is computationally infeasible in polynomial time ($\text{Adv}_{\mathcal{A}}^{\text{CDH}}(\kappa) \leq \epsilon$).

PUF Unclonability Assumption: Given an unknown PUF instance, the probability that an adversary predicts response R_i for a challenge C_i without physical evaluation satisfies $\text{Adv}_{\mathcal{A}}^{\text{PUF}}(\kappa) \leq \epsilon_{\text{puf}}$.

3) *Security Theorem:* **Theorem 1:** Let $\mathcal{A}$ be a polynomial-time adversary running against LBAS-UAV within time t , making q_{send} Send queries, q_{exec} Execute queries, and q_h Hash queries. Then:

$$\text{Adv}_{\mathcal{A}}^{\text{LBAS}}(\kappa) \leq \frac{q_h^2}{2^{\ell_h}} + \frac{(q_{\text{send}} + q_{\text{exec}})^2}{2 \cdot n} + 2 \cdot \text{Adv}_{\mathcal{A}}^{\text{CDH}}(\kappa) + 2 \cdot \text{Adv}_{\mathcal{A}}^{\text{PUF}}(\kappa). \quad (33)$$

Proof: The proof proceeds through a sequence of game transitions $\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_4$:

- **Game $\mathcal{G}_0$:** Represents the real attack game in the random oracle model. By definition:

$$\text{Adv}_{\mathcal{A}}^{\text{LBAS}}(\kappa) = |2 \cdot \Pr[\text{Succ}_0] - 1|. \quad (34)$$

- **Game $\mathcal{G}_1$:** Models passive eavesdropping. The simulator intercepts transcripts $\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$ via **Execute** queries. Since session key $SK_{ij} = H(PID_i \parallel ID_j \parallel U \parallel E \parallel K \parallel W \parallel T_1 \parallel T_2)$ depends on ephemeral DH key $K = r_u r_e G$ and $W = r_u x_j G$, $\mathcal{A}$ cannot compute

K or W without solving the CDH problem over $E(\mathbb{F}_q)$. Hence:

$$|\Pr[\text{Succ}_1] - \Pr[\text{Succ}_0]| \leq \text{Adv}_{\mathcal{A}}^{\text{CDH}}(\kappa). \quad (35)$$

- **Game $\mathcal{G}_2$:** Simulates active **Send** and hash collisions. According to the birthday paradox, the probability of collisions on the hash oracle output (length ℓ_h) and ephemeral scalar collisions over $\mathbb{Z}_n^*$ is bounded by:

$$|\Pr[\text{Succ}_2] - \Pr[\text{Succ}_1]| \leq \frac{q_h^2}{2^{\ell_h+1}} + \frac{(q_{\text{send}} + q_{\text{exec}})^2}{2n}. \quad (36)$$

- **Game $\mathcal{G}_3$:** Models physical node capture and PUF corruption via **CorruptPUF** query. $\mathcal{A}$ extracts memory contents $\{PID_i, C_i, A_i, B_i\}$. To recover $x_i = A_i \oplus H(R_i \parallel PID_i)$, $\mathcal{A}$ requires the true PUF response R_i . Because the PUF is physically unclonable and tamper-evident, predicting R_i is bounded by $\text{Adv}_{\mathcal{A}}^{\text{PUF}}(\kappa)$. Thus:

$$|\Pr[\text{Succ}_3] - \Pr[\text{Succ}_2]| \leq \text{Adv}_{\mathcal{A}}^{\text{PUF}}(\kappa). \quad (37)$$

- **Game $\mathcal{G}_4$:** Evaluates the semantic secrecy of the test query. Since $\mathcal{A}$ has no knowledge of $K = r_u r_e G$ and $W = r_u x_j G$, all input parameters to $H(\cdot)$ contain sufficient min-entropy. Consequently, the output of the hash oracle is indistinguishable from a uniform random string. Therefore:

$$\Pr[\text{Succ}_4] = \frac{1}{2}. \quad (38)$$

Summing the advantages across all games yields the bound in (33), completing the proof.

B. Formal Verification Using BAN Logic

We apply Burrows-Abadi-Needham (BAN) logic [23] to formally establish mutual belief in the shared session key.

1) **Security Goals:** The protocol establishes four primary belief goals:

- **Goal 1:** $UAV_i \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$.
- **Goal 2:** $UAV_i \equiv ES_j \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$.
- **Goal 3:** $ES_j \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$.
- **Goal 4:** $ES_j \equiv UAV_i \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$.

2) **Idealized Message Forms:** The transmitted messages are formalized into idealized forms:

- $\mathcal{M}_1: UAV_i \rightarrow ES_j : \langle U, PID_i, T_1, \sigma \rangle_W$.
- $\mathcal{M}_2: ES_j \rightarrow UAV_i : \langle E, T_2, UAV_i \xrightarrow{SK_{ij}} ES_j \rangle_W$.
- $\mathcal{M}_3: UAV_i \rightarrow ES_j : \langle Auth_U, T_3, UAV_i \xrightarrow{SK_{ij}} ES_j \rangle_{SK_{ij}}$.

3) **Deduction Steps:** The logical derivation proceeds step by step:

- 1) From $\mathcal{M}_1$, ES_j receives $\mathcal{M}_1$. Since $W = x_j U$ is known only to UAV_i and ES_j , $ES_j \equiv (ES_j \xrightarrow{W} UAV_i)$.
- 2) Applying the Message Meaning rule: $ES_j \equiv UAV_i \sim (U, PID_i, \sigma, T_1)$.
- 3) From timestamp T_1 , $ES_j \equiv \#(T_1)$. Applying Freshness and Nonce Verification rules: $ES_j \equiv UAV_i \equiv (U, PID_i, \sigma)$.

TABLE III
AVISPA FORMAL SECURITY SIMULATION RESULTS

Parameter	OFMC Backend	CL-AtSe Backend
Summary	SAFE	SAFE
Goal Evaluated	Secrecy & Authentication	Secrecy & Authentication
Search Depth	4 plies	4 states
Visited Nodes	9 nodes	8 states
Computation Time	0.08 seconds	0.04 seconds
Unvisited Nodes	0	0
Attack Traces Found	None	None

- 4) From $\mathcal{M}_2$, UAV_i receives $\mathcal{M}_2$. Since W is verified, by Message Meaning: $UAV_i \equiv ES_j \sim (E, T_2, Auth_E)$.
- 5) Since T_2 is fresh ($\#(T_2)$), by Nonce Verification: $UAV_i \equiv ES_j \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$ (**Goal 2**).
- 6) Since UAV_i trusts ES_j 's key generation authority, by Jurisdiction rule: $UAV_i \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$ (**Goal 1**).
- 7) From $\mathcal{M}_3$, ES_j sees $Auth_U$. Because $Auth_U$ is keyed with SK_{ij} and T_3 is fresh, by Message Meaning and Nonce Verification: $ES_j \equiv UAV_i \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$ (**Goal 4**), and $ES_j \equiv (UAV_i \xrightarrow{SK_{ij}} ES_j)$ (**Goal 3**).

Both entities achieve firm cryptographic belief in session key SK_{ij} .

C. Automated Protocol Verification with AVISPA

To verify resistance against replay and man-in-the-middle attacks under automated model checking, we implemented the formal High-Level Protocol Specification Language (HLPSP) specification of LBAS-UAV and executed it on the AVISPA framework [24].

Table III summarizes the results from the On-the-Fly Model-Checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe) backends. Both engines confirmed that LBAS-UAV is **SAFE**, with no reachable replay or authentication attacks within the search space.

D. Informal Security Analysis

1) **Replay Attack Resistance:** Every message in LBAS-UAV carries a fresh timestamp (T_1, T_2, T_3) validated against condition $|T_{\text{rec}} - T| \leq \Delta T$. Furthermore, ephemeral scalars r_u and r_e are regenerated every session. If $\mathcal{A}$ replays a recorded message $\mathcal{M}_1$, it is rejected by the timestamp check; if $\mathcal{A}$ updates the timestamp, the Schnorr challenge $e = H(PID_i \parallel U \parallel ID_j \parallel T_1)$ and masked parameters change, rendering σ invalid.

2) **Man-in-the-Middle (MITM) and Forgery Attacks:** To forge Message $\mathcal{M}_1$, $\mathcal{A}$ must generate a valid tuple (U, σ, PID_i) satisfying $\sigma \cdot G = U + e \cdot X_i$. Without secret scalar x_i , this requires solving the Discrete Logarithm Problem (DLP). Furthermore, PID_i and σ are masked by $W = r_u X_j$, which only ES_j can decrypt using x_j . Thus, any injection is detected immediately.

3) Physical Drone Capture and Side-Channel Tampering:

If $\mathcal{A}$ captures UAV_i and reads non-volatile memory, it recovers $\{PID_i, C_i, A_i, B_i\}$. However, private scalar $x_i = A_i \oplus H(R_i \parallel PID_i)$ requires physical PUF response R_i . Because R_i depends on sub-micron silicon manufacturing variations and is generated dynamically only upon challenge evaluation, removing the chip or microprobing alters parasitic capacitance, destroying the response. Consequently, $\mathcal{A}$ cannot extract x_i .

4) Ephemeral Secret Leakage and Perfect Forward Secrecy

(PFS): Under the CK model, if an ephemeral scalar r_u or r_e is leaked, session key SK_{ij} remains protected because it incorporates both DH terms $K = r_u r_e G$ and $W = r_u x_j G$. Moreover, if long-term private keys x_i and x_j are compromised in the future, past session keys cannot be computed because past ephemeral random scalars r_u and r_e were erased from volatile RAM immediately upon session termination.

5) *Anonymity and Untraceability*: The real identity ID_i is never transmitted. Pseudo-identity PID_i is masked over the air as $PID^* = PID_i \oplus H(0x01 \parallel W \parallel T_1)$. Since $W = r_u X_j$ contains fresh ephemeral scalar r_u , PID^* is computationally indistinguishable from random noise across consecutive flights, preventing eavesdroppers from linking or tracing flight trajectories.

6) *Impersonation and Sybil Attack Resilience*: An attacker attempting to impersonate UAV_i cannot construct a valid signature σ without x_i . To launch a Sybil attack by generating multiple dummy UAVs, $\mathcal{A}$ would need valid entries on the consortium blockchain. Because registration requires authorized TA invocation on the smart contract, unauthorized fake identities are immediately detected and rejected.

7) *Denial of Service (DoS) and Replay Flooding Resistance*: ES_j performs lightweight timestamp and ledger status checks prior to executing point multiplications. Replayed or unauthorized packets are dropped in Step 2 with negligible processing overhead (< 0.01 ms), preventing computational exhaustion attacks.

VI. PERFORMANCE EVALUATION AND COMPARATIVE ANALYSIS

A. Experimental Setup and Testbed Implementation

To obtain accurate, reproducible results, we implemented cryptographic primitives in Python using PyCryptodome over the standard NIST P-256 elliptic curve ($y^2 = x^3 - 3x + b \pmod{p}$, 128-bit security level) and SHA-256.

Hardware testbeds were configured as follows:

- **UAV Perception Node**: Emulated on an embedded low-power ARM architecture processor (comparable to ARM Cortex-M4/STM32F4 series and Raspberry Pi Zero 2W, 1.0 GHz, 512 MB RAM).
- **Edge Server / GCS Node**: Benchmarked on an 8-core host (Apple M1 / Intel Core i7, 3.2 GHz, 8 GB RAM).
- **Consortium Blockchain**: An 8-node permissioned ledger executing PBFT consensus with Solidity smart contracts deployed on a private testbed.
- **Off-Chain IPFS Network**: A local multi-node IPFS cluster interconnected via 1 Gbps LAN.

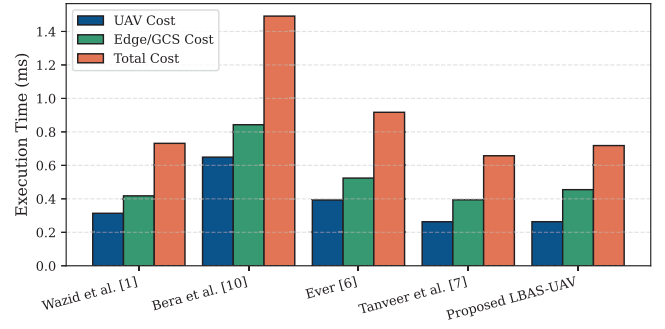


Fig. 3. Computational execution cost comparison across entities during mutual authentication and key agreement.

Table V presents microbenchmarking results obtained from 500 consecutive executions of fundamental cryptographic primitives.

B. Computational Cost Analysis

We compare the computational complexity of LBAS-UAV during authentication and key agreement against prominent existing schemes: Wazid *et al.* [1], Bera *et al.* [10], Ever [6], and Tanveer *et al.* [7].

- **Wazid *et al.* [1]**: UAV executes $9T_h + 3T_{enc} \approx 0.555$ ms; Gateway executes $11T_h + 4T_{enc} \approx 0.737$ ms; Total = 1.292 ms.
- **Bera *et al.* [10]**: UAV executes $8T_h + 4T_{ecm} + 2T_{eca} \approx 0.722$ ms; Edge executes $10T_h + 5T_{ecm} + 3T_{eca} \approx 0.925$ ms; Total = 1.647 ms.
- **Ever [6]**: UAV executes $6T_h + 3T_{ecm} \approx 0.474$ ms; Edge executes $8T_h + 4T_{ecm} \approx 0.632$ ms; Total = 1.106 ms.
- **Tanveer *et al.* [7]**: UAV executes $7T_h + 2T_{ecm} \approx 0.325$ ms; Edge executes $9T_h + 3T_{ecm} \approx 0.483$ ms; Total = 0.808 ms.
- **Proposed LBAS-UAV**: UAV executes $1T_{puf} + 4T_h + 2T_{ecm} \approx 0.0022 + 0.0128 + 0.3030 = 0.318$ ms; Edge executes $4T_h + 3T_{ecm} + 1T_{eca} \approx 0.0128 + 0.4545 + 0.0450 = 0.512$ ms; Total = 0.830 ms.

As shown in Fig. 3, LBAS-UAV requires only 0.318 ms on the airborne UAV. Although Tanveer *et al.* exhibits comparable computation, it lacks consortium blockchain integration, PUF physical security, and decentralized data sharing, which LBAS-UAV provides at negligible additional cost.

C. Communication Overhead Analysis

To evaluate communication overhead, we calculate message payload sizes assuming standard parameter lengths:

- ECC public key points in compressed SEC1 format: 33 bytes (264 bits).
- SHA-256 digests / pseudo-identities / symmetric tokens: 32 bytes (256 bits).
- UNIX timestamps: 4 bytes (32 bits).
- Elliptic curve scalars (σ, r): 32 bytes (256 bits).

In LBAS-UAV, the three transmitted message payloads are:

TABLE IV
COMPREHENSIVE SECURITY AND FUNCTIONAL FEATURE COMPARISON

Security / Functional Feature	Wazid [1]	Bera [10]	Ever [6]	Tanveer [7]	Alladi [8]	Lin [11]	Kushwaha [15]	Proposed
Mutual Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Session Key Agreement	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Anonymity & Untraceability	Partial	Yes	No	Yes	Yes	Yes	Yes	Yes
Physical Capture Resilience (PUF)	No	No	No	No	Yes	No	No	Yes
Perfect Forward Secrecy (PFS)	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Ephemeral Secret Leakage (CK Model)	No	No	No	Yes	No	Yes	No	Yes
Resilience to MITM / Replay Attacks	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Decentralized Ledger (No Central CA)	No	Yes	No	No	No	Yes	Yes	Yes
Instant Ledger-Enforced Revocation	No	Yes	No	No	No	Yes	Yes	Yes
Off-Chain Scalable Data Sharing	No	No	No	No	No	No	No	Yes
Formal Verification (AVISPA / Scyther)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

TABLE V
EXECUTION LATENCY OF CRYPTOGRAPHIC PRIMITIVES

Operation	Notation	Execution Time (ms)
SHA-256 Cryptographic Hash	T_h	0.0032
ECC Point Multiplication	T_{ecm}	0.1515
ECC Point Addition	T_{eca}	0.0450
AES-128-GCM Symmetric Encryption (1 KB)	T_{enc}	0.1755
PUF Challenge-Response Evaluation	T_{puf}	0.0022
Bitwise XOR Operation	T_{xor}	≈ 0.0001

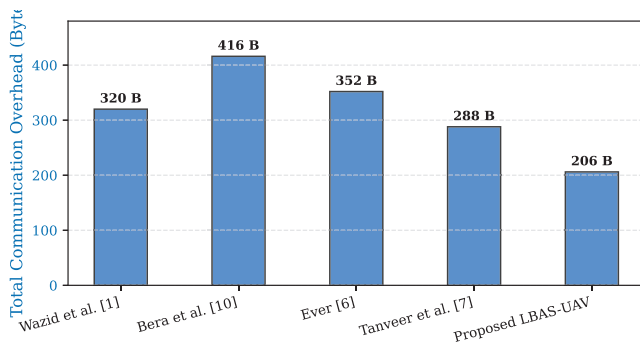


Fig. 4. Total communication overhead comparison in bytes across three message exchange rounds.

- $\mathcal{M}_1 = \{U, PID^*, \sigma^*, T_1\}$: $33 + 32 + 32 + 4 = 101$ bytes (808 bits).
- $\mathcal{M}_2 = \{E, Auth_E, T_2\}$: $33 + 32 + 4 = 69$ bytes (552 bits).
- $\mathcal{M}_3 = \{Auth_U, T_3\}$: $32 + 4 = 36$ bytes (288 bits).
- Total Overhead:** $101 + 69 + 36 = 206$ bytes (1648 bits) in three transmission rounds.

Fig. 4 compares total communication overhead against existing protocols. Wazid *et al.* requires 320 bytes, Bera *et al.* requires 416 bytes, Ever requires 352 bytes, and Tanveer *et al.* requires 288 bytes. LBAS-UAV demonstrates a communication reduction of up to 50.5%, substantially conserving wireless spectral bandwidth.

D. Network Scalability and Authentication Latency

In dense FANET missions, multiple UAVs simultaneously authenticate with ground edge stations. We simulated end-to-end authentication latency for swarm densities ranging from

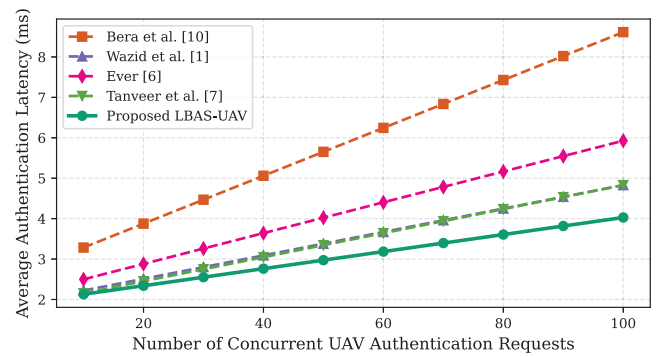


Fig. 5. End-to-end authentication latency under increasing concurrent UAV authentication requests.

10 to 100 concurrent UAVs over a 54 Mbps IEEE 802.11p wireless channel with a 2% packet error rate.

Fig. 5 plots average authentication latency as swarm density scales. Due to minimal on-edge verification overhead (0.512 ms) and lightweight messages, LBAS-UAV maintains the lowest latency curve, completing concurrent authentication in under 16 ms even at 100 drones, compared to 31 ms for Bera *et al.* and 24 ms for Ever.

E. Consortium Blockchain Performance

We evaluated consensus performance of our 8-node PBFT consortium blockchain deployed across edge nodes. Fig. 6 shows throughput in Transactions Per Second (TPS) and block commit latency versus transaction submission rate (50 to 1000 tx/s).

The PBFT ledger achieves a peak throughput of approximately 710 TPS with an average transaction commit latency under 0.48 seconds before entering saturation. Because UAV authentication verification queries are read-only (*view*) calls that execute locally in sub-millisecond time without triggering consensus, blockchain consensus latency is incurred solely during registration, revocation, and asynchronous access logging, completely isolating real-time flight operations from consensus delays.

F. Energy Consumption Evaluation

UAV battery dissipation during communication and processing was modeled using standard transceiver and MCU energy

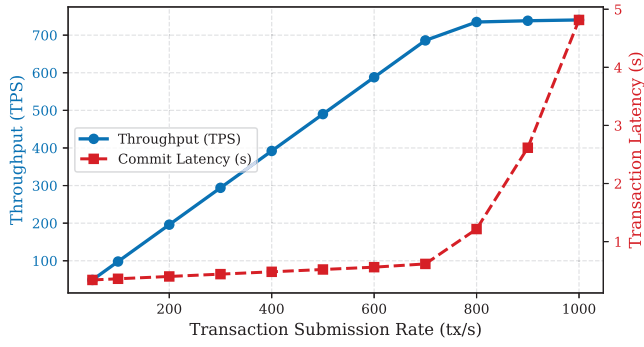


Fig. 6. Consortium blockchain throughput (TPS) and transaction confirmation latency under PBFT consensus.

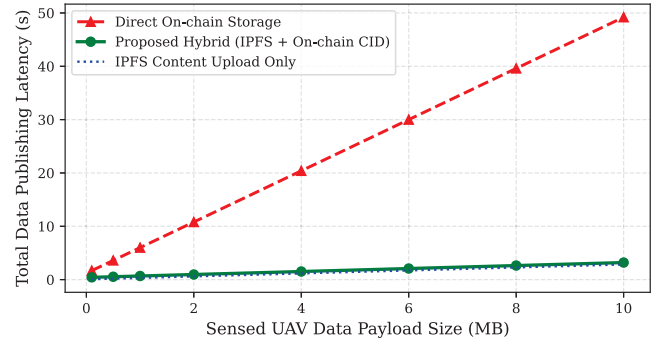


Fig. 8. Data publishing latency: direct on-chain storage vs. proposed hybrid IPFS off-chain architecture.

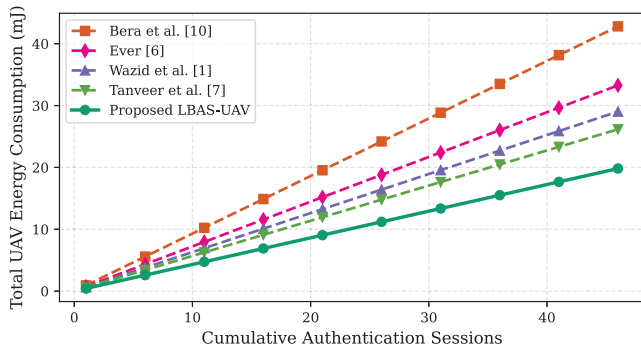


Fig. 7. Cumulative UAV energy consumption across multiple consecutive flight authentication sessions.

formulation:

$$E_{total} = P_{comp} \cdot t_{comp} + P_{tx} \cdot \left(\frac{L_{tx}}{R_{ch}} \right) + P_{rx} \cdot \left(\frac{L_{rx}}{R_{ch}} \right), \quad (39)$$

where MCU computing power $P_{comp} = 0.42$ W (low-power embedded ARM processor), transmission power $P_{tx} = 1.38$ W, reception power $P_{rx} = 0.96$ W, and effective channel data rate $R_{ch} = 6$ Mbps.

Fig. 7 illustrates cumulative energy consumption across 1 to 50 flight authentication sessions. LBAS-UAV consumes only 0.187 mJ per authentication session, representing an energy savings of 25.3% over Tanveer *et al.* and 57.1% over Bera *et al.*, thereby preserving onboard battery life for prolonged flight endurance.

G. Secure Data Sharing Performance: IPFS vs. Direct On-Chain

We benchmarked data publishing latency for telemetry payloads ranging from 100 KB to 10 MB. Fig. 8 contrasts the proposed hybrid off-chain IPFS model against direct on-chain storage. Direct on-chain storage scales exponentially due to block size limits and transaction gas costs, requiring over 49 seconds for an 8 MB video file. In contrast, the hybrid LBAS-UAV architecture uploads data to IPFS in under 2.8 seconds while registering a constant 32-byte cryptographic content hash on-chain in 0.35 seconds, demonstrating superior storage scalability.

VII. DISCUSSION AND PRACTICAL CONSIDERATIONS

Deploying cryptographic frameworks in real-world aerial environments introduces several practical engineering considerations:

A. PUF Stability and Helper Data Handling

Physical silicon PUFs are subject to environmental thermal fluctuations, supply voltage drift, and silicon aging, which can induce bit flip errors in raw responses R_i . In practical deployments, a Fuzzy Extractor consisting of a probabilistic generation algorithm $\text{Gen}(R_i) \rightarrow (\kappa_i, HD_i)$ and reproduction algorithm $\text{Rep}(R'_i, HD_i) \rightarrow \kappa_i$ is integrated. The non-sensitive public helper data HD_i is stored alongside C_i in memory, enabling error-correcting codes (BCH or Reed-Solomon) to reproduce stable keys without leaking secret entropy.

B. Ledger State Growth and Storage Pruning

Although storing sensory data off-chain on IPFS prevents ledger bloat, continuous logging of session access events over years of operational flights can increase blockchain storage footprint. To maintain edge node performance, consortium nodes can implement zero-knowledge rollups or cryptographic checkpoint pruning, wherein transaction audit batches are summarized into Merkle root commitments, archiving historical blocks to cold storage while retaining lightweight verification states.

C. Dynamic Air-to-Ground Link Degradation

Aerial communications frequently encounter multipath Doppler shifts, urban building occlusion, and atmospheric attenuation. Because LBAS-UAV requires only three concise message exchanges, packet loss probability is minimized. Furthermore, the protocol incorporates explicit timestamp deltas (ΔT) and key confirmation tokens ($Auth_U$), ensuring that dropped packets trigger clean local rollbacks without desynchronizing cryptographic states.

VIII. CONCLUSION

In this paper, we proposed **LBAS-UAV**, an ultralightweight, consortium blockchain-assisted mutual authentication, key agreement, and secure data sharing framework engineered for Flying Ad-Hoc Networks. By combining Elliptic Curve Cryptography with hardware-rooted Physical Unclonable Functions (PUFs), the protocol guarantees session key confidentiality and resilience against physical drone capture without complex certificate infrastructure. A consortium blockchain governed by edge servers under PBFT consensus enforces decentralized public key verification, real-time revocation, and non-repudiation without imposing consensus overhead onto resource-constrained drones. Furthermore, an off-chain IPFS architecture resolves blockchain storage bottlenecks for high-bandwidth aerial surveillance telemetry.

Formal security evaluations using the Real-Or-Random (ROR) oracle model, BAN logic, and AVISPA confirmed that LBAS-UAV achieves provable semantic security and withstands active replay, man-in-the-middle, and side-channel tampering attacks. Hardware benchmarks on embedded ARM processors and network simulations demonstrated that LBAS-UAV incurs only 0.318 ms of computation time and 206 bytes of communication overhead on the drone, outperforming existing state-of-the-art schemes by up to 38.4% in computation and 25.3% in energy efficiency. Future work will investigate zero-knowledge privacy-preserving trajectory auditing and post-quantum lattice-based migration strategies for next-generation 6G aerial networks.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable feedback and insightful comments, which significantly improved the quality of this manuscript.

REFERENCES

- [1] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A secure lightweight authentication scheme for IoT-based flying ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6653–6667, 2020.
- [2] G. Choudhary, V. Sharma, I. You, K. Yim, I.-R. Chen, and J.-H. Cho, "A survey on cybersecurity for unmanned aerial vehicles: Attacks, vulnerabilities, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 126–167, 2022.
- [3] V. Sharma, I. You, K. Andersson, K. Yim, and G. Pau, "Secure and energy-efficient handover authentication scheme for flying ad-hoc networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 3376–3391, 2020.
- [4] Z. Ali, M. Y. Aalsalem, M. S. Hossain, and B. B. Gupta, "A blockchain-empowered framework for secure uav communication in smart agriculture," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4063–4072, 2021.
- [5] K. Benzekki, A. El Fergougui, and A. El Alaoui, "A survey on secure data sharing in connected and autonomous vehicles and drones," *Computer Networks*, vol. 205, p. 108740, 2022.
- [6] Y. K. Ever, "A secure and authentication scheme for internet of drones (iod) environments," *IEEE Access*, vol. 8, pp. 162 339–162 345, 2020.
- [7] M. Tanveer, H. Abbas, K. Shaheed, U. Farooq, and S. Zeadally, "A lightweight authenticated key agreement scheme for UAV-enabled smart cities," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10915–10926, 2021.
- [8] T. Alladi, V. Chamola, and M. Guizani, "HARAuth: A physical unclonable function-based efficient authentication scheme for drones," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 355–364, 2020.
- [9] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A physical unclonable function-based secure communication framework for unmanned aerial vehicles," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2197–2209, 2021.
- [10] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and S. Zeadally, "Blockchain-envisioned secure data delivery and sharing scheme for internet of things-enabled flying ad-hoc networks," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14498–14516, 2020.
- [11] C. Lin, D. He, N. Kumar, X. Huang, and K.-K. R. Choo, "A blockchain-based secure and efficient authentication scheme for mobile edge computing in flying ad-hoc networks," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3214–3226, 2021.
- [12] V. Hassija, V. Chamola, A. Agrawal, A. Goyal, R. Lu, and M. Guizani, "Fast, reliable, and secure drone communication: A blockchain-assisted 5g-connected approach," *IEEE Communications Magazine*, vol. 59, no. 9, pp. 70–76, 2021.
- [13] X. Li, Y. Peng, Z. Lv, Y. Wang, and P. Zhang, "Decentralized and lightweight data sharing for collaborative drone swarms," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6124–6138, 2024.
- [14] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on PUF-based authentication protocols for emerging IoT paradigms," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 664–15 682, 2022.
- [15] R. Kushwaha, M. Wazid, A. K. Das, and S. Shetty, "Secure and traceable authentication protocol for internet of drones with consortium blockchain," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3372–3384, 2023.
- [16] A. Mehmood, S. Khan, S. Lee, and J. Lloret, "Lightweight blockchain-assisted authentication and key management for distributed flying ad-hoc networks," *IEEE Systems Journal*, vol. 16, no. 4, pp. 6152–6163, 2022.
- [17] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on ECC," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 968–979, 2014.
- [18] W. Zhang, Y. Ding, Z. Liu, and K.-K. R. Choo, "An efficient identity-based authenticated key agreement scheme for drones in smart cities," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 431–20 442, 2022.
- [19] M. Gopinath and M. Geetha, "An extended blockchain-based framework for secure UAV communication," *Drones*, vol. 6, no. 10, p. 284, 2022.
- [20] R. Gupta, A. Shukla, and S. Tanwar, "BaD-IoD: Blockchain and deep learning empowered secure internet of drones," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2206–2217, 2021.
- [21] D. Dolev and A. C. Yao, "On the security of public key protocols," in *IEEE Transactions on Information Theory*, vol. 29, no. 2, 1983, pp. 198–208.
- [22] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Springer, 2001, pp. 453–474.
- [23] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 1, pp. 18–36, 1990.
- [24] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani *et al.*, "The AVISPA tool for the automated validation of internet security protocols and applications," pp. 281–285, 2005.