International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(16s) 2026 pp. 1388-1412
https://doi.org/10.70917/ijcisim-2026-4673
Article

A Lightweight Forward-Unlinkable Authenticated Key Agreement Protocol for Secure UAV Communication in Internet of Drones Environments

Kamal A. ElDahshan[1], *Hani Saad Hassan Issa[2], Hala Mohamed Abbas[3]

[1]Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt
[2]Faculty of Computer Studies, Arab Open University, Cairo, Egypt
[3]Computer Science Department, Faculty of Computers and Artificial Intelligence, Capital University, Cairo, Egypt.
*Corresponding Author Email: hanisaadcs@gmail.com
Co-authors Email: Kadahshan@azhar.edu.eg, dahshan@gmail.com

Abstract: Secure authentication, confidentiality, and communication privacy are critical requirements in mission-critical Internet of Drones (IoD) applications such as disaster response, defense, border monitoring, and surveillance, where Unmanned Aerial Vehicles (UAVs) are being increasingly used. Many lightweight authentication and key agreement protocols initially developed for IoT and wireless sensor networks, however, are not sufficiently robust to provide an appropriate level of security for IoD applications, especially considering physical drone capture, long-term credential compromise, and post-compromise transcript correlation challenges. This paper proposes a lightweight forward-unlinkable authenticated key agreement protocol for secure UAV communication in adversarial IoD environments. The protocol is based on an authenticated ephemeral X25519 Diffie–Hellman exchange, key derivation from HKDF, transcript authentication using HMAC and explicit key confirmation. The session keys are based on newly created ephemeral secrets and don't necessarily stem from long-term stored credentials, so if a drone's memory is compromised, the session keys of previous sessions are not exposed. Pseudonym rotation and context-bound key derivation are incorporated to strengthen session separation and reduce the possibility of linking previous or future protocol transcripts after drone capture. The proposed protocol is analyzed under an extended Dolev–Yao adversary model with drone-capture capability and is supported through ProVerif-style symbolic validation. The analysis indicates that the protocol provides mutual authentication, session-key secrecy, replay resistance, man-in-the-middle protection, forward secrecy, and forward unlinkability. Operation-level cost analysis and published embedded-platform benchmarks show that the cryptographic overhead remains lightweight, with per-session computation requiring only a few milliseconds and communication overhead remaining within a few hundred bytes. The results demonstrate that lightweight authenticated key agreement can provide practical cyber-security and privacy protection for resource-constrained UAV communication without relying on heavy infrastructures such as blockchain, certificate-based PKI, or zero-knowledge proof systems.

Keywords: Cybersecurity; Authenticated Key Agreement; Internet of Drones; Forward Unlinkability; Drone-Capture Resistance

1. Introduction

The UAV industry is expanding at a fast pace in the civilian, industrial, and military sectors, and is experiencing an increasing need for secure real-time data capture, processing, and transmission. Current UAV uses are precision agriculture, medical delivery, disaster response, and wide-area surveillance and tactical reconnaissance. In such an environment, lightweight, authenticated communication, privacy protection, and cyber-security aspects between

drones, users, and the control infrastructure are crucial for the successful execution of missions and in the planning of cyber-defense strategy.

Current lightweight authentication approaches used in the Internet of Things (IoT) and Wireless Sensor Networks (WSNs) do not fit all the IoD cybersecurity risks. Many schemes still rely on static identifiers, long-term secret keys, or predictable session parameters. These design decisions can make UAV systems vulnerable to long term privacy leakage even if they can be successfully authenticated in the short term. The attack of a drone-capture one, which involves physical access to a drone and the extraction of cryptographic data from it, with the aim of correlating past or future logins, is a great danger. These attacks can jeopardize anonymity, mission confidentiality and secure operational coordination over multiple missions.

Authenticated key agreement protocols designed for IoT and WSN environments typically provide limited protection against post-physical-capture unlinkability. A key vulnerability in IoD security is the lack of protection against post-compromise transcript correlation when long-term credentials or existing pseudonym states are exposed. To address this problem, this paper presents a lightweight forward-unlinkable authenticated key agreement (FAKA) protocol for secure and privacy-preserving UAV communication in adversarial IoD environments. The protocol combines a pseudonym-based authentication with an authenticated X25519 Diffie-Hellman exchange and introduces various operational context parameters such as time slot, location zone, mission code, group identifier and pseudonymous drone identity as input to the key derivation function. These parameters are not considered cryptographic secrets, but they are each uniquely part of an operational configuration, provide increasing key diversity, enhance separation of sessions from correlation attacks, and retain the low overhead characteristics needed for constrained UAV deployments.

Recent research shows that securing UAV communication is more challenging than securing traditional IoT and WSN environments because drones are mobile, network topology is dynamic, and devices may be exposed to physical capture. Survey-level and system-level analyses indicate that UAV networks are highly susceptible to cyber and physical attacks, especially in IoD deployments that require continuous communication and real-time control at scale. The results show that conventional lightweight authentication mechanisms are not enough to secure the privacy of UAV systems in the face of long-term privacy leakage and sophisticated adversarial models.

To reduce the threats of traffic-analysis and metadata-correlation, this protocol has forward secrecy, unlinkability, route-aware validation and pseudonym rotation. No infrastructure intensive systems like certificate-based PKI, blockchain consensus or zero-knowledge proof systems are required to achieve these cybersecurity guarantees. Rather, the design only performs one lightweight X25519 Diffie Hellman operation per session and relies on efficient symmetric primitives, such as HMAC and HKDF, so that the protocol is appropriate for those UAV platforms with limited computation, energy, and communication resources.

The architecture, cryptographic construction, message flow and context-bound key derivation framework of the proposed protocol is described. The security analysis consists of checking resistance to impersonation, replay, man-in-the-middle, drone-capture and session-linkability attacks. The performance evaluation further indicates that the protocol has low cryptographic overhead and stable scalability under both normal and stressed network conditions. Overall, the results show that forward-unlinkable authenticated key agreement can provide a practical cybersecurity solution for protecting UAV communication in realistic, resource-constrained, adversarial IoD scenarios.

2. Contributions

This paper presents a lightweight forward-unlinkable authenticated key agreement protocol for improving cybersecurity in Internet of Drones (IoD) environments. The proposed protocol is motivated by the limitations of existing IoT-, WSN-, and UAV-oriented authentication schemes, particularly their weak resistance to drone-capture attacks, post-compromise transcript correlation, and long-term identity tracing. The design consists of lightweight key generation, authenticated key agreement, context-bound key derivation and pseudonym evolution for the purpose of

supporting secure and privacy-preserving UAV communication in realistic adversarial environments. The principal results are described briefly as follows:

2.1 Authenticated Diffie–Hellman with Context-Bound Key Derivation

This paper proposes a session key establishment mechanism based on authenticated X25519 Diffie-Hellman exchange and HKDF-based key derivation. The final session key is derived from a fresh ephemeral shared secret rather than from long-term stored credentials. The auxiliary operational parameters such as location zone, time slot, group identifier, mission route code and pseudonymous drone identity are fed as context-dependent parameters in the key derivation. These parameters are not considered secrets of computation, but are used to enhance separation of sessions, diversification of keys and resistance to transcript correlation without an inflation of the computational costs of the system, which is required to be lightweight for use on UAVs.

2.2 Forward-Unlinkable Mutual Authentication Protocol

The paper introduces a forward-unlinkable mutual authentication protocol between the drone and the control server for adversarial IoD environments. The proposed design separates long-term enrollment material from session-key generation and relies on fresh ephemeral Diffie–Hellman values for every session. Pseudonym rotation, nonce-based freshness, and context-bound derivation are combined to prevent an adversary from linking previous or future sessions after drone capture or credential exposure. This makes the protocol particularly relevant for privacy-preserving cyber-security applications where UAV identity, mission activity, and communication history must remain protected against long-term surveillance.

2.3 Formal Security Model and Attack Resistance

A security model is defined for IoD scenarios in which the adversary can monitor, modify, replay, or inject messages over the wireless channel and may also physically capture a drone to extract stored credentials or pseudonym state. The protocol is analyzed against major cyber-attacks, including impersonation, replay, man-in-the-middle manipulation, transcript correlation, and post-compromise session-linkability attacks. Symbolic reasoning and ProVerif-style validation are used to support the claimed security properties, including mutual authentication, session-key secrecy, replay resistance, and forward unlinkability under the assumed adversarial model.

2.4 Lightweight Design for Resource-Constrained UAVs

The protocol is intended for use in real-world deployments in IoD and IoT-like settings. It restricts public-key computation to just one X25519 Diffie–Hellman exchange per session and it uses efficient symmetric primitives, such as HMAC and HKDF, for authentication and key derivation. The design does not use consensus mechanisms, blockchain or certificate-based PKI, which are infrastructure intensive. To achieve this, the protocol has low computation and communication overhead, while providing strong authentication, forward secrecy, and privacy protection for the UAV communication.

2.5 Comparative Evaluation Against Recent UAV Authentication Schemes

The novelty of the proposed protocol lies in combining four security elements within one lightweight Drone–Control Server AKE design: authenticated ephemeral X25519 Diffie–Hellman key exchange, HKDF-based context-bound session-key derivation, pseudonym evolution, and explicit post-capture forward unlinkability. Existing UAV and IoD authentication schemes often address only part of this problem. Some schemes provide anonymity or pseudonymity but do not bind the session key to mission context. Others use PUFs, blockchain, certificateless cryptography, or batch authentication to strengthen authentication, but they may introduce additional hardware dependence, infrastructure cost, storage requirements, or do not explicitly prevent post-compromise transcript correlation.

In contrast, the proposed protocol keeps the key agreement strictly lightweight and software-deployable. The session key is derived only from fresh ephemeral Diffie–Hellman material, while location zone, time slot, route

identifier, mission code, group identifier, and current pseudonym are used as auxiliary KDF context rather than as secrets. This makes each session completely distinct without increasing trust assumptions. Therefore, the main contribution is not merely another lightweight authentication protocol, but a forward-unlinkable, context-bound, capture-resilient AKE model specifically designed for resource-constrained UAV communication.

Table 1: Comparison with recent UAV authentication and key agreement schemes

Study	**Main mechanism**	**Main limitation in IoD context**	Proposed protocol
Jeong et al. [21]	Concurrent two-party key exchange with forward unlinkability	Provides forward unlinkability, but session separation is not explicitly bound to mission route, location zone, group identifier, and operational context	Adds context-bound HKDF derivation using mission, route, zone, time, group, and pseudonym values
Han et al. [22]	Lightweight secure communication with batch authentication for UAV swarm	Focuses mainly on swarm-level batch authentication efficiency rather than post-capture transcript unlinkability	Prioritizes forward unlinkability after drone capture with low per-session computation and communication overhead
Choi et al. [23]	PUF-based authentication and key agreement for IoD	Depends on hardware-rooted PUF assumptions and does not focus on software-only context-bound session separation	Uses software-deployable X25519, HMAC, HKDF, and pseudonym rotation without requiring PUF hardware
Li et al. [24]	Certificateless AKA based on Chebyshev polynomials	Relies on certificateless public-key structure and does not explicitly integrate route-aware operational context into session-key derivation	Avoids certificate infrastructure and binds each session key to UAV mission context
Xie and Wang [25]	PUF-based anonymous cross-domain UAV authentication	Focuses on cross-domain UAV authentication and PUF-based protection rather than lightweight Drone–CS forward-unlinkable AKE	Targets Drone–CS authenticated key agreement with forward secrecy, pseudonym evolution, and context-bound derivation

3. Related Work

Key agreement and authentication protocols for resource-constrained environments, such as Wireless Sensor Networks (WSNs), the Internet of Things (IoT), and more recently the Internet of Drones (IoD), have been extensively studied over the past decade. These protocols are designed to be secure, lightweight and scalable, and to withstand attacks that are typical in open wireless networks. Unlinkability has been little addressed, despite many existing schemes providing mutual authentication and basic forward secrecy, and it is a critical aspect of UAV-based systems where metadata exposure can harm mission confidentiality and operational anonymity.

3.1 Early Approaches: Chaotic Maps and Static Identity

Early authentication and key agreement schemes for IoT and WSN environments include the chaotic map–based three-party protocol proposed by Xie et al. [1]. This work introduced user authentication and password-based

key exchange with low computational overhead. However, the scheme relied on password tables and static identity structures, rendering it vulnerable to impersonation attacks and lacking forward secrecy and unlinkability guarantees.

Lee et al. [2] improved upon this approach by eliminating the password table and strengthening resistance to password-guessing attacks. Despite these improvements, both schemes remained relying on identifiers that were static or semi-static. This means long-term secrets can be exposed to enable adversaries to trace sessions through time, which is why these protocols are not adequate for highly dynamic environments of UAVs in which mobility and state changes also need to be cryptographically reflected.

3.2 Authentication in Heterogeneous Networks

Subsequent research extended lightweight authentication techniques to heterogeneous sensor and IoT networks [27]. Protocols proposed by Turkanovic et al. [3] and Farash et al. [4] employed efficient symmetric cryptographic operations and pseudonym-based authentication to reduce computational overhead on constrained devices. While these approaches improved efficiency, their pseudonym mechanisms lacked sufficient dynamism; repeated pseudonym usage or predictable session parameters enabled adversaries to correlate sessions and infer identity information.

Amin et al. [5] further analyzed these schemes and identified vulnerabilities to known session-specific information attacks and offline smart-card compromise. The results indicate that pseudonym is not enough without fine-grained rotation and strict isolation of sessions, especially when an adversarial setting is possible where partial disclosure of credentials is possible.

3.3 Blockchain and UAV-Specific Models

More recent efforts have explored decentralized identity management using blockchain technologies. For instance, Yazdinejad et al. [6] proposed a blockchain-based UAV authentication architecture to improve auditability and decentralized trust. While these schemes do lessen the burden on central authorities, they add a lot of computation and storage overhead, which makes them difficult to implement on small UAVs. Moreover, they are not fully unlinkable, as there can be metadata or time-correlated patterns within the blockchain that reveal the linkage.

3.4 Forward Unlinkability: A New Standard

Jeong et al. [7] introduced one of the first UAV-focused key agreement protocols to integrate forward unlinkability through session-based pseudonym rotation. Their approach marked an important advancement toward privacy-preserving UAV communication. However, their unlinkability model remains constrained: session updates occur periodically but do not incorporate real-time context, such as drone movement, mission route dynamics, or environment-dependent operational changes.

Some authentication protocols aim to preserve user and device anonymity, but anonymity alone does not guarantee that sessions cannot be linked [26]. Related privacy-preserving VANET approaches have also demonstrated the importance of balancing anonymity with accountability, particularly where mobile entities must remain protected from unnecessary identity disclosure while still supporting authorized verification [28]. Table 1 summarizes previous research across four key dimensions. In terms of pseudonym-update logic, Xie et al. [1] and Lee et al. [2] rely on static identifiers, whereas Turkanović et al. [3] use periodic pseudonym updates that are not tied to individual sessions. Most current schemes, except Jeong et al. [7], do not explicitly address forward unlinkability under drone-capture conditions. Even in Jeong et al. [7], The unlinkability is not yet perfect since the pseudonyms are not updated dynamically to response to operational conditions. Long-term secret based protocols that use long-term secrets for message authentication are also vulnerable to post-compound session correlation attacks under key-compromise impersonation (KCI). If the session parameters can be forecasted or the pseudonyms do not change, an adversary having any partial credentials can still correlate messages that have been recorded. The proposed protocol alleviates these problems by utilizing ephemeral Diffie-Hellman secrets where the inputs are specific to the session, to produce session keys, which helps to isolate individual transcripts even if the credentials were compromised.

3.5 Summary and Gap Analysis

The development of IoT, WSN, and IoD authentication protocols reveals three persistent limitations:

1. Static or semi-static identifiers that allow tracing across sessions
2. Predictable operational context and unbounded session parameters
3. Weaknesses in countering drone-capture attacks, in which stolen secrets undermine previous or subsequent sessions

These limitations indicate that existing solutions are inadequate for lightweight, secure, and privacy-aware UAV communication in hostile Internet of Drones environments. The proposed protocol addresses these gaps by incorporating:

- Forward secrecy (authenticated by Diffie-Hellman)
- Context-bound key derivation
- Fine-grained pseudonym rotation
- Dynamic session isolation

Consequently, the proposed design offers a lightweight forward-unlinkable authenticated key agreement model with stronger privacy and forward-secrecy properties than existing solutions, as summarized in Table 1.

Table 2: Gap analysis

Protocol	Forward Secrecy	Unlinkability	DH-Based Key Exchange	Context Binding
Xie et al. [1]	✗	✗	✗	✗
Lee et al. [2]	✓	✗	✗	✗
Turkanovic et al. [3]	✓	✗	✗	✗
Yazdinejad et al. [6]	✓	Partial	✗	✗
Jeong et al. [7]	✓	✓	Partial	✗
Proposed Protocol	✓	✓	✓	✓

Scoring criteria: Table 1 uses forward secrecy (✓) to indicate that the protocol uses ephemeral key exchange or explicitly claims forward secrecy. Unlinkability (✓) indicates that the protocol states session unlinkability or untrace ability. DH-based key exchange (✓) indicates that a Diffie-Hellman mechanism is used to establish the session key. Context binding (✓) indicates that operational parameters are explicitly included in session-key derivation.

Justification of classifications: Xie et al. [1] are classified as not providing forward secrecy or unlinkability because their scheme relies on static identities and does not use ephemeral Diffie-Hellman key exchange. Lee et al. [2] claim forward secrecy, but their scheme lacks unlinkability and context binding. Turkanović et al. [3] improve session freshness through a nonce-based design; however, their protocol does not use Diffie-Hellman key exchange and does not provide unlinkability. Yazdinejad et al. [6] improve security through blockchain-assisted authentication and may provide partial unlinkability, but operational context is not incorporated into key derivation. Jeong et al. [7] improve privacy by enabling forward unlinkability through evolving pseudonyms, but their design does not bind session keys to dynamic operational context. In this regard, the proposed protocol is distinguished by its ephemeral Diffie-Hellman key exchange, explicit context binding, and forward-unlinkability features in a lightweight UAV-oriented design.

4. System and Threat Model

4.1 System Model

The proposed authenticated key agreement protocol is implemented in a typical Internet of Drones (IoD) environment involving three operational entities: the User, the Drone, and the Control Server. But the cryptographic key agreement itself is a two- party authenticated key agreement performed only between the Drone and the Control Server. The User initiates an access request, relies upon communication or receives authorized UAV services through the Control Server, but the User is not involved in ephemeral Diffie Hellman key generation, session-key derivation, key-confirmation computation. This distinction ensures that the protocol remains a lightweight Drone-Control Server AKE scheme while still supporting user-driven UAV service access under server authorization.

- User (U): A legitimate operator, soldier, or authorized client device that requests access to UAV data or services. The User interacts with the IoD system through the Control Server but is not a cryptographic participant in the proposed Drone-Control Server key agreement process.
- Drone (D): A mobile UAV with sensors, communication modules, persistent memory and light cryptographic capabilities. Each drone has a long-term enrollment secret shared with the Control Server, is pseudonymous, and is involved in an authenticated Diffie-Hellman key exchange to create a session key with the Control Server.
- Control Server (CS): A centralized trusted authentication authority responsible for managing drone identities, mission routes, pseudonym states, access authorization, and session validation. The Control Server authenticates the Drone, participates in the Diffie-Hellman exchange, derives the session key, and authorizes user access to UAV services through a separate access-control layer.

4.2 Key System Features

The protocol uses authenticated Diffie–Hellman key exchange, in combination with context-bound key derivation, to provide forward secrecy and forward unlinkability.

- Each session will be a new X2551V Diffie–Hellman exchange between the drone and CS.
- Shared secret goes into a key derivation function (HKDF).
- Auxiliary inputs to the key derivation function (KDF) are operational context parameters that enhance session uniqueness. These parameters are not treated as cryptographic secrets; instead, they provide operational binding. The context vector includes:
    - Drone pseudonym (P_D)
    - Time slot (T_s)
    - Operational group identifier (GID)
    - Mission route code (*RouteID*)
    - Location zone identifier (*ZoneID*)

These contextual parameters are not treated as cryptographic secrets but ensure that each session key is tightly bound to its operational context, reducing the feasibility of session correlation and metadata analysis.

4.3 Rationale for Context Binding

Diffie–Hellman (DH) key exchange has forward security, but the unlinkability must prevent correlation between sessions. Context binding ensures that even if two sessions occur close in time, their derived keys differ due to distinct operational parameters.

Context-aware security mechanisms have been proposed as effective approaches for strengthening authentication in dynamic and mobile networks. Prior work on IoD and UAV-swarm authentication shows that incorporating operational parameters, such as temporal information, mission context, and group-level authentication data, can reduce the feasibility of replay and impersonation attacks [13,15,16]. However, these approaches often treat context as a primary secret or rely mainly on pseudonym changes. In contrast, the proposed protocol integrates context binding as auxiliary KDF input alongside a cryptographically strong DH exchange, achieving both unlinkability and forward secrecy.

4.4 Session Lifecycle

Each session proceeds as follows:

1. Drone and CS exchange Diffie–Hellman public values.
2. Fresh nonces are generated by both parties.
3. The session key is derived using HKDF over the DH shared secret and contextual parameters.
4. The drone rotates its pseudonym in coordination with the CS.

This process ensures:

- **Forward secrecy**: past sessions remain secure even if long-term secrets are compromised.
- **Forward unlinkability**: future sessions cannot be linked to previous ones.
- **Replay resistance**: freshness is enforced via timestamps and nonces.

4.5 Threat Model

The newly proposed protocol adopts a realistic adversarial model consistent with the Dolev–Yao assumptions, adapted to the Internet of Drones (IoD) domain.

Physical drone capture has been widely recognized as one of the most severe threats in IoD environments. Prior research demonstrates that once a UAV is captured, attackers can extract stored credentials, session information, and identity-related data, enabling impersonation and long-term surveillance attacks [11–13]. Unlike traditional network nodes, UAVs operate in exposed environments, making physical compromise a realistic assumption rather than an edge case. Consequently, authentication and key agreement protocols for UAV networks must be designed under the assumption that adversaries may obtain access to drone memory. The adversary □ has the following capabilities [29].

The adversary □ is assumed to be computationally bounded but capable of:

- Eavesdropping, modifying, replaying, or injecting messages over the communication channel.
- Capturing drones and extracting stored long-term secrets, pseudonyms, and previous session data.
- Attempting impersonation attacks using leaked credentials.
- Observing drone movement patterns and attempting to correlate session activity to infer identities or mission trajectories.

Table 3: Mitigation Mechanism

Threat Type	Mitigation Mechanism
Drone Capture Attack	Forward secrecy via authenticated Diffie–Hellman + forward unlinkability
Session Replay	Nonce freshness + timestamp validation

Identity Tracing / Linkability	Pseudonym rotation + context-bound KDF inputs
Man-in-the-Middle (MitM)	Mutual authentication via HMAC + DH key confirmation
Long-Term Surveillance / Route Inference	Context-bound session derivation + session isolation

Security Threats Considered as shown in Table 2

Device capture implies compromise of the captured device's future sessions. The protocol does not attempt to protect a captured drone from impersonation (KCI scenario) but ensures that compromise does not reveal or link past sessions.

5. Security Goals

The proposed protocol is designed to satisfy the following formal security objectives:

A. Mutual Authentication

The Drone (D) and Control Server (CS) authenticate each other through HMAC-based verification and Diffie–Hellman key confirmation. Successful completion of the protocol ensures that both parties participated in the session and derived the same session key.

B. Forward Secrecy

Compromise of long-term secrets or drone memory does not reveal previously established session keys. This property is achieved using ephemeral Diffie–Hellman (X25519) contributions in each session.

C. Forward Unlinkability

Separate communication sessions cannot be linked, correlated, or distinguished, even if the adversary later obtains long-term secrets. This is ensured through pseudonym rotation and context-bound key derivation.

D. Resistance to Replay and Impersonation

All protocol exchanges incorporate fresh nonces, timestamp validation, and message authentication codes, preventing message replay and identity forgery.

E. Resilience Under Drone Capture

Even if a drone is physically captured and its stored credentials are extracted, the adversary cannot reconstruct past session keys or correlate past sessions due to forward secrecy and session isolation mechanisms.

6. Proposed Protocol

The proposed protocol introduces a context-bound, session-ephemeral authenticated key agreement mechanism between the Drone (D) and the Control Server (CS). Although a User (U) may initiate a service request or relay messages in the IoD environment, the cryptographic exchange is strictly performed between D and CS. Therefore, the proposed construction should be interpreted as a two-party Drone-Control Server AKE protocol with user-side access handled separately by the Control Server. Each session is uniquely tied to real-time drone state parameters, supporting forward secrecy and forward unlinkability even if an adversary later compromises a drone.

6.1 Context Construction

To support mission-awareness without relying on location as a secret, each drone constructs a context vector describing its operational state. This context is not used as a standalone key but contributes auxiliary entropy to session key derivation.

$$Ctx_D = (Zone_{ID} \parallel Route_{ID} \parallel Mission_{Code})$$

Where:

- $Zone_{ID}$: coarse geographic grid cell (not raw GPS)
- $Route_{ID}$: planned route identifier
- $Mission_{Code}$: operational mission tag

Using coarse zones mitigates GPS drift and spoofing impact. Security does not depend on the secrecy of context values.

Context values are represented as coarse operational identifiers and hashed into a commitment h_{ctx}. Raw mission or location data is not transmitted. After session key (SK) establishment, encrypted context verification may optionally be performed.

6.2 Context Validation

Upon receiving a request, the Control Server validates:

- Time slot consistency
- Zone plausibility based on mission plan
- Route alignment

This step ensures operational correctness but does not serve as the primary security mechanism.

6.3 Key Hierarchy Overview

Each drone D and the Control Server CS share a long-term enrollment secret K_{seed}. The following functional keys are used:

- **Enrollment secret** K_{seed}: Used only for registration and pseudonym provisioning.
- **Binder key** K_{bind}: Derived as $HMAC(K_{seed}, P_D)$; authenticates messages M_1 and M_2.
- **Session key** SK : Derived from the ephemeral Diffie–Hellman shared secret Z (see Section 5.4); authenticates message M_3.
- **Pseudonym update**: $P_D^{(1)} = H(P_D^{(i)})$ (no separate key).

Detailed derivation of auxiliary keys (K_{MAC}, K_E) is provided in Section 5.6.1.

Algorithm 1: Context Vector Construction (Drone Side)

Employed to obtain an auxiliary context input to derive a session key following DH exchange.

Input:

- PD – current pseudonymous drone identity
- Z_{ID} – current location zone identifier (coarse, not raw GPS)
- T_s – current time slot (e.g., 5-minute interval)
- GID – group / squadron identifier
- R_{ID} – mission / route code

Output:

Ctx_D – context vector (auxiliary KDF input)

Steps:

1. ***Obtain current pseudonym:***

$$P_D \leftarrow GetCurrentPseudonym()$$

2. ***Obtain operational state parameters:***

$$Z_{ID} \leftarrow GetCurrentZoneID()T_s \leftarrow GetCurrentTimeSlot(5\ minutes)GID \leftarrow GetGroupID()R_{ID} \leftarrow GetMissionCode()$$

3. ***Construct context vector:***

$$Ctx_D \leftarrow Z_{ID} \parallel T_s \parallel GID \parallel R_{ID} \parallel P_D$$

4. ***Return*** Ctx_D

The context vector Ctx_D is not a cryptographic key. Rather, it gives auxiliary entropy, and operational binding and is included in the KDF when deriving the session key with the DH shared secret.

6.4 Session Key Derivation Framework

The protocol is based on an authenticated Diffie–Hellman exchange for the purpose of establishing a session key, which is then explicitly derived from the HKDF. The drone and server generate ephemeral secret scalars a and b, exchange DH public values $A = g^a$ and $B = g^b$, and compute the shared secret $Z = g^{ab}$.

The session key SK is derived using HKDF (RFC 5869) in two steps:

Step 1 Extract:

$$PRK = HKDF - Extract(salt, Z)\ ,$$

where

$$salt = H(P_D \parallel T_s \parallel N_D \parallel N_{CS}).$$

The salt is a hash of the current pseudonym, time slot, and both nonces. It is not secret but ensures that different sessions produce different pseudorandom keys even if Z is identical.

Step 2 Expand:

$$SK = HKDF - Expand(PRK, info, L)$$

where $L = 32$ bytes (AES-256 key length), and

$$info = "IoD - AKE - v1" \parallel h_{ctx},$$

with

$$h_{ctx} = H(Ctx_D),$$

and

$$Ctx_D = Z_{ID} \parallel T_s \parallel GID \parallel R_{ID} \parallel P_D.$$

Context handling: The context vector Ctx_D (location zone, mission code, group ID, etc.) is neither authenticated nor encrypted on its own. It is only mixed into the KDF via the $info$ field (through h_{ctx}) and into the salt. This provides session uniqueness and unlinkability without requiring secrecy or integrity protection for the

context parameters themselves. Authentication of the overall exchange is provided separately by HMAC tags (τ_1, τ_2, τ_3) that depend on the derived SK.

Reproducibility: The exact HKDF implementation shall use HMAC-SHA256 as the underlying hash function. All inputs are concatenated in the order shown. The hash function H is SHA-256.

6.5 Authentication Model

Although there is a User component, it's still only the pairing process between the Drone and the Control Server that results in an authenticated key exchange. The User can ask for information or service from the Drone and might be used also as a protocol message relay, but the actual key computation parts like generating Diffie-Hellman ephemeral values or calculating the mutual secret, key derivation for the Drone-Control Server communication, and verification of the key-confirmation tag, do not happen at the user's device. The reason these are kept separate from user's device is that it makes the designed protocol very efficient i.e. lightweight without involving the user in additional rounds, synchronization demands, and key-management activities related to a full three-party key-exchange scheme.

The Control Server is responsible for authenticating and authorizing the User through an application-level access-control mechanism that is separate from the proposed Drone-Control Server AKE. After the Drone and the Control Server establish a secure session, UAV data or service responses can be delivered to the authorized User through the Control Server using an appropriate protected channel. The Drone-Control Server session key is not directly issued to the User. This separation prevents confusion between UAV session-key establishment and user access authorization, while preserving the main security objective of the protocol: lightweight mutual authentication and forward-unlinkable key agreement between the Drone and the Control Server.

6.6 Notation and Key Derivation Summary

Table 3 is a summary of the main symbols in the protocol. To prevent ambiguity, the key hierarchy and session-key derivation procedure is given below in summary.

Table 4: Notation

Symbol	Meaning
U, D, CS	User, Drone, and Control Server
P_u, P_D	User and drone pseudonyms
T	Timestamp / session slot
SK	Final session key
$MAC_k(x)$	MAC using key K over message x
EK, DK	Symmetric encryption/decryption
$Nonce_u$, $Nonce_D$	Fresh nonces from user and drone
Sig()	Signature or MAC proving authenticity
Nonce	Random nonce

6.6.1 Key Hierarchy

Each drone D shares a long-term enrollment secret K_{seed} with the Control Server CS. For each pseudonym epoch, both parties derive functional sub-keys using a pseudorandom function based on the current drone pseudonym P_D. This separation ensures that message authentication and session establishment rely on distinct cryptographic material. The long-term enrollment secret is used only for provisioning and binder-key derivation, whereas the session key is derived exclusively from the ephemeral Diffie–Hellman shared secret.

Session-key derivation. Let $Z = g^{ab}$ denote the ephemeral Diffie–Hellman shared secret computed from fresh session exponents a and b. The session key is derived using HKDF in two stages:

$$PRK = HKDF\backslash mbox - Extract(salt, Z) SK = HKDF\backslash mbox - Expand(PRK, info, 32)$$

where

$$salt = H(P_D \parallel T_s \parallel N_D \parallel N_{CS})$$

and

$$info = "IoD - AKE - v1" \parallel h_{ctx}.$$

Here, P_D denotes the current drone pseudonym, T_s the session time slot, and N_D and N_{CS} the fresh nonces generated by the drone and the control server, respectively.

Role of context: The context value h_{ctx} binds the session to operational parameters without treating them as secret key material. Accordingly, context contributes to session differentiation through the HKDF $info$ field, while the secrecy of the final session key remains rooted in the ephemeral Diffie–Hellman secret Z.

6.7 Message Exchange Flow (Forward Secrecy)

The process of message exchange between Drone (D) and the Control Server (CS) goes through the authentication key agreement. If necessary, a User device can make a request or even act as a message relay at the application layer but from a cryptographic point of view, the User will never be part of the authenticated key agreement. Here is a typical AKE protocol. A two-party authenticated Diffie-Hellman, that gives ephemeral keys and explicit key confirmation, is a base structure. This model confirms that the session key is calculated only from the freshly contributed Diffie-Hellman ephemeral key exchanges of D and CS and that the session key is not the result of long-term identities or User credentials at that stage.

6.7.1 Pre-Shared Setup

Each drone shares a long-term enrollment secret K_{seed} with CS, used only for registration and pseudonym provisioning. It is not used to authenticate protocol transcripts, ensuring forward unlinkability under compromise.

Both parties maintain a synchronized pseudonym P_D valid for the current epoch.

Step 1: Drone → Server

Drone generates:

- ephemeral DH key pair (a, A)
- nonce N_D
- pseudonym P_D

PSK-derived binder key:

$$K_{bind} = HMAC(K_{seed}, P_D)$$

Authentication tag:

$$\tau_1 = HMAC(K_{bind}, P_D \mid\mid A \mid\mid N_D \mid\mid T_s)$$

Message:

$$M_1 = \langle P_D, T_s, A, N_D, \tau_1 \rangle$$

Step 2: Server → Drone

Server verifies τ_1 using its copy of $Kseed$ and PD. Server sends:

$$M2 = \langle B, NCS, \tau 2 \rangle$$

Where

$$\tau 2 = HMAC(Kbind, PD \;||\; A \;||\; B \;||\; ND \;||\; NCS)$$

$$\tau 2 = HMAC(Kbind\,, PD \;||\; A \;||\; B \;||\; ND \;||\; NCS)$$

Step 3: Key Confirmation (SK-based)

Both compute:

$$Z = \; gab$$

$$SK = HKDF(Z, \dots)$$

$$Then{:}\; \tau 3 = HMAC(SK, "D" \;||\; A \;||\; B \;||\; ND \;||\; NCS)$$

$$PD(i+1) = H(PD(i))$$

$$Kbind\;(i+1)\; = H(Kbind\;(i)\;)Security$$

After rotation, previous values erased → attacker cannot verify old transcripts.

6.7.2 Authentication of DH shares.

The Diffie-Hellman key agreement can be authenticated in two steps. At first, the exchanged values are authenticated by the binder authentication tags which are computed from the binder keys and combine pseudonyms nonces Diffie Hellman values, and transcripts of the session. In the second step, the key-confirmation tag serves for an explicit validation of the key where the session key derived from the shared secret of the ephemeral Diffie-Hellman key agreement is used for their computation. In this way, the manipulation of public values or transcript fields by a potential adversary will result in a verification error So not allowing traditional man-in-the-middle attacks and at the same time authenticating the key exchange.

$$M_2 = \langle B, N_{CS}, \tau_2 \rangle$$

6.8 Security Analysis

The security properties of the protocol can be given semi-formal reasoning in the following arguments which are justified by the symbolic model and ProVerif based validation presented later; these should not be taken as a computational proof.

- **Forward Secrecy:** The session key SK is derived from the ephemeral Diffie–Hellman shared secret Z. Therefore, compromise of long-term credentials does not reveal previously established session keys.
- **Forward Unlinkability:** Authentication of the transcript is supported by pseudonym-bound binder MACs τ_1 and τ_2, where $K\_bind \;=\; HMAC(K\_seed, P\_D)$, together with explicit key confirmation through τ_3, which is computed from the session key SK derived from the ephemeral shared secret Z. If an adversary later compromises K_seed and the current pseudonym, it still cannot recompute historical authentication tags, because each transcript depends on the pseudonym valid in its own epoch and pseudonyms evolve through a one-way hash chain. Consequently, past binder keys are not recoverable from the current state, and previously recorded sessions cannot readily be validated or linked.

- **Mutual Authentication:** Mutual authentication is supported by successful verification of the binder-based authentication tags during transcript establishment and the SK-based explicit key confirmation at the end of the exchange.
- **Replay Protection:** Fresh nonces and timestamp validation support resistance to replay of previously recorded messages.
- **MitM Resistance:** Any modification of the Diffie–Hellman public values or transcript fields changes the derived session state and causes authentication or key-confirmation failure, supporting resistance to man-in-the-middle attacks.
- **UKS and Reflection Resistance:** Role-specific labels and transcript binding reduce the feasibility of reflection and unknown key-share attacks.

Overall, these properties are argued at the symbolic level and are further supported by the ProVerif analysis presented in Section 8.6.

6.9 Pseudonym Lifecycle

CS provisions each drone with a pseudonym pool. The drone stores only the current and next pseudonym and deletes expired values.

- Rotation is time-based
- No pseudonym history retained on drone
- CS maintains secure mapping
- Capture reveals only current identity

This prevents recovery of historical pseudonyms post-capture.

PD(i+1)=H(PD(i))

Pseudonyms evolve through a forward-secure hash chain. After rotation, the previous value is erased from drone memory, preventing retrospective linkage even under capture.

7. Performance Evaluation

In this section, the computation cost, communication overhead and scalability of the proposed protocol are evaluated based on operation level cost analysis and published embedded platform benchmarks.

7.1 Latency Evaluation

7.1.1 Operation-Level Latency

The protocol involves:

1. Context construction
2. X25519 Diffie–Hellman key exchange
3. HMAC authentication
4. HKDF session key derivation

7.1.2 Realistic Cryptographic Latency

Since UAV platforms mostly use microcontrollers (such as Cortex-M4 at 80 MHz) or embedded processors (like Raspberry Pi 4 at 1.5 GHz), the performance analysis is based mainly on cryptographic benchmark papers and published articles instead of protocol implementation direct measurements. The X25519 latency predictions for

Cortex-M4 chips are extracted from Curve25519 benchmarking results in SUPERCOP for highly optimized fixed-base scalar multiplication implementations of Curve25519 in SUPERCOP. The predictions of latency for Raspberry Pi 4 systems, as well, are the results of OpenSSL 1.1.1 benchmarks, that are compiled with default optimization flags (-O2). HMAC-SHA256 and HKDF latencies have been interpolated from the libgcrypt measurements published to similar platforms. We should note that they present rather the typical best-case benchmarks and not the implementation specifics. Performance will be affected not only by firmware and compiler optimizations, but also by power management modes. In Tables 5 and 6, we list the estimated single-step latencies and the overall cryptographic cost per communication session. The latter will be the sum of all individual step costs.

Table 5: Latency in Different Environments

Operation	Cortex-M4 (80 MHz)	Raspberry Pi 4
X25519 public key generation	3–5 ms	< 0.2 ms
X25519 shared secret	3–5 ms	< 0.2 ms
HMAC-SHA256	~0.2 ms	< 0.01 ms
HKDF	~0.3 ms	< 0.02 ms

Table 6: Microbenchmark Latency of Protocol Operations

Operation	Estimated Latency (Cortex-M4)
Context vector construction	negligible
X25519 DH public key generation	3–5 ms
X25519 shared secret computation	3–5 ms
HMAC-SHA256	0.2 ms
HKDF derivation	0.3 ms

Thus, the total cryptographic overhead per session on UAV hardware is approximately 7-11 ms.

Python microbenchmarks confirm constant-time execution but are not used as performance proof.

7.2 Computational Cost Analysis

Table 7: Computational cost per entity

Entity	HMAC Ops	HKDF Ops	DH Ops	Complexity
Drone	2	1	1	O(1)
CS	2–3	1	1	O(1)
User	1	0	0	O(1)

Even with ECC operations, the cost remains lightweight compared to blockchain or zk-proof based UAV schemes.

7.3 Communication Overhead

Table 8: Communication Overhead

Message	Size
D → CS	~160–200 B
CS → D	~80–100 B
ACK	32–64 B

Total per session ≈ 250–350 bytes, well within UAV communication limits.

7.4 Scalability Evaluation

Table 9: Scalability Behavior

Sessions	Avg Verification Time (relative)
10	O(1)
100	O(1)
1,000	O(1)
10,000	O(1)

As shown in Table 9, server-side operations remain constant time and do not require persistent per-session state.

7.5 Network Latency Simulation

Protocol responsiveness was tested in a realistic scenario of UAV communications through a Montecarlo simulation. The simulation involved running 10,000 authentication sessions for each scenario that generated random numbers instead of capturing actual radio traffic from UAVs. The authors combined the time needed to run cryptographic protocols obtained from benchmarking on an actual device with estimated delays due to network round-trip delay generated at random. The method clearly separates protocol compute time from communications delay that can result in a handshake taking way longer time than required.

Python 3.11 was used with NumPy for generating random numbers and computing descriptive Statistics. All runs were fixed at the same pseudo random seed (42) to allow one to recreate the same latency distributions as well as results provided that identical code and parameter settings are used. Simulation runs were carried out on a regular Windows laptop with Intel Core i5/i7 processor, 16 GB memory, and no GPU. If one wants to know exact CPU model, it will have to be reported from the computer specification in the experiment room. The program itself is not a very hardware demanding one, since it does not involve real cryptographic performance tests but it rather models delays statistics only. Therefore, the reported results are reproducible on a standard workstation using the same Python version, NumPy package, random seed, network-delay ranges, cryptographic-cost assumptions, and real-time threshold.

Table 10: Simulation parameters and reproducibility configuration

Parameter	Simulation A	Simulation B
Number of runs	10,000	10,000
Network delay model	Uniform distribution	Uniform distribution
Network delay range	60–120 ms	150–350 ms
Cryptographic cost	7–11 ms	7–11 ms
Real-time threshold	200 ms	200 ms
Random seed	42	42
Software	Python 3.11, NumPy	Python 3.11, NumPy
Hardware	Windows workstation, Intel Core i5/i7-class CPU, 16 GB RAM	Windows workstation, Intel Core i5/i7-class CPU, 16 GB RAM

The total handshake time was calculated as the sampled network delay plus the estimated cryptographic processing cost. Under Simulation A, all sessions are completed within the 200 ms thresholds, confirming that the protocol is responsive under normal UAV communication conditions. Under Simulation B, approximately 48% of sessions exceeded the threshold, indicating that performance degradation was caused primarily by network delay

rather than cryptographic computation. These results support the claim that the proposed protocol remains computationally lightweight, while real-time responsiveness in stressed environments depends mainly on wireless channel quality, relay distance, congestion, and interference.

7.6 Comparison with Related Protocols

Table 11: Comparative evaluation

Protocol	**Forward Secrecy Mechanism**	**Unlinkability**	**Computation Cost**	**Comm. Overhead**	**Notes**
Xie et al.	None	No	Low	Moderate	Linkability risk
Lee et al.	None	No	Low	Moderate	Static IDs
Turkanovic et al.	Partial	No	Low	High	Smart-card dependency
Jeong et al.	Periodic rekey	Partial	Moderate	High	No strong state-binding
Proposed Protocol	Authenticated DH (X25519) + context binding	Yes	Low	Low	FS + privacy

Table 11 provides a comparative evaluation of the proposed protocol against several state-of-the-art UAV/IoT authentication schemes. According to the comparative analysis, the proposed protocol offers a balanced trade-off across the analyzed criteria: it provides forward secrecy and unlinkability, unlike Xie et al., Lee et al., and Turkanović et al.; it avoids heavy infrastructure, unlike blockchain-based approaches; and it provides both context binding and a full DH-based key exchange, unlike schemes that rely mainly on periodic rekeying or pseudonym updates. Consequently, the proposed protocol provides a strong security, privacy, and efficiency balance while maintaining low computational and communication overhead among the schemes reviewed.

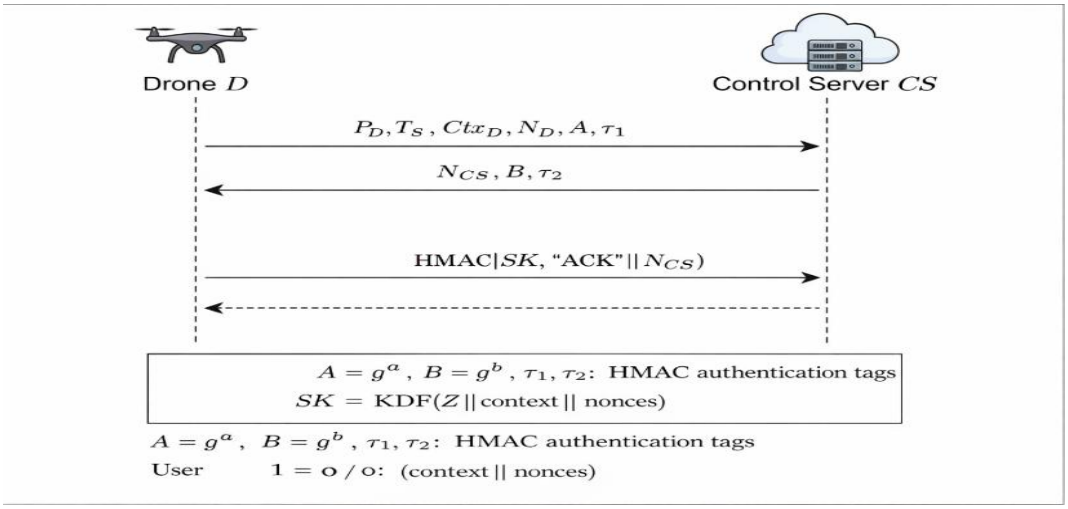

Figure 1: Protocol Sequence Diagram

Where:

- A = DH public value ga
- B = DH public value gb
- τ1, τ2 = HMAC authentication tags

- PRK=HKDF-Extract(salt=H(PD‖ Ts‖ ND‖ NCS), IKM=Z)
- SK=HKDF-Expand (PRK, IoD−AKE−v1‖ hctx, 32)
- KE is not used as KDF key material. All session keys are derived solely from the ephemeral DH secret Z. Long-term enrollment secrets are excluded from SK derivation, preserving forward secrecy.

8. Formal Security Validation

To validate the security guarantees of the proposed authenticated key agreement protocol, the scheme is analyzed under a symbolic Dolev–Yao adversarial model extended with drone-capture capability. In this model, the adversary has full control over the public wireless channel and can eavesdrop, modify, replay, inject, or block protocol messages. The adversary may also physically capture a drone after session completion and extract its stored long-term enrollment secret, current pseudonym state, and locally stored protocol information. However, the adversary is assumed to be computationally bounded and unable to break the underlying cryptographic primitives, including the X25519 Diffie–Hellman assumption, HMAC, HKDF, and one-way hash functions.

The validation Mainly looks at the major aspects of cyber-security needed for lightweight mutual authentication protocols in IoD settings like IoT devices. The main requirements are confidentiality of the session key, mutual authentication, forward secrecy, replay resistance, man-in-the-middle resistance, resistance of the key-share to unauthorized access, and forward unlinkability even with the drone being captured. The method involves a combination of symbolic reasoning with ProVerif-style verification to show that the proposed protocol not only keeps session isolation but also does not allow the adversary to compute or verify old session keys even after having the long-term keys revealed.

8.1 Secrecy of the Session Key

The session key $SK$is derived from the ephemeral Diffie–Hellman shared secret:

Z= gab

where a and b are freshly generated ephemeral secret scalars. Since Z is not transmitted and cannot be computed without knowledge of a or b, the secrecy of SK relies on the hardness of the X25519 Diffie-Hellman problem. Compromise of long-term credentials (Kseed) does not reveal past session keys.

8.2 Mutual Authentication

Mutual authentication using binder is done in two steps. Initially, the Drone and the Control server will confirm the binder-based authentication tags that are derived from the binder key K_bind and serve to link the session transcript with the current pseudonym, nonces, and Diffie, Hellman public values. Then both the entities will verify the key-confirmation tag that is explicitly communicated $\boldsymbol{\tau_3}$, which is computed under the session key **SK** derived from the ephemeral Diffie–Hellman shared secret **Z**. Successful verification of these tags indicates that both parties participated in the same session and derived the same session key.

8.3 Forward Secrecy

Because SK depends on ephemeral keys a, b compromise of long-term secrets or stored state after protocol completion does not allow reconstruction of previous session keys.

8.4 Forward Unlinkability

Forward unlinkability is supported by the combination of epoch-specific pseudonyms, binder-based transcript authentication, and session-key confirmation derived from fresh ephemeral Diffie–Hellman input. In the finalized protocol, the authentication tags τ_1 and τ_2 are computed using the binder key $K\_bind = HMAC(K\_seed, P\_D)$, whereas explicit key confirmation is provided separately by τ_3, which is computed from the session key SK derived from the ephemeral shared secret Z.

Let an adversary compromise a drone after session completion and obtain the long-term enrollment secret K_seed together with the current pseudonym P_D^(now). This disclosure does not enable direct validation of earlier transcripts, because each recorded session is associated with the pseudonym valid in its own epoch, and pseudonyms evolve through the one-way chain $P\_D\^(i+1) = H(P\_D\^(i))$. As a result, past pseudonym states and their corresponding binder keys cannot be reconstructed from the current state.

Moreover, the session key SK depends on the fresh ephemeral Diffie–Hellman secret Z, so later compromise of stored long-term material does not allow recomputation of the session-specific key-confirmation value τ_3. Consequently, previously recorded sessions cannot readily be linked to the captured drone through transcript validation, which supports forward unlinkability at the symbolic-analysis level.

8.5 UKS and Reflection Resistance

Role-specific labels (“S”, “D”) are included in authentication tags, preventing reflection and unknown key-share attacks.

8.6 ProVerif Verification

The proposed protocol was modeled in ProVerif using a symbolic representation of the Drone–Control Server authenticated key agreement process. The model includes public-channel communication, adversarial message interception and injection, nonce generation, pseudonym evolution, binder-key authentication, HKDF-based session-key derivation, and explicit key confirmation. A reveal event was also included to represent post-session drone capture, where the adversary obtains the long-term enrollment secret and the current pseudonym state.

The ProVerif validation was structured around three main security checks. First, session-key secrecy was tested by verifying that the adversary cannot obtain the session key derived from the fresh ephemeral Diffie–Hellman shared secret. Second, mutual authentication was evaluated through correspondence assertions between the Drone and the Control Server, ensuring that successful completion by one party implies participation of the intended peer in the same protocol session. Third, forward unlinkability was examined by modeling pseudonym evolution and session-specific key confirmation, showing that previously recorded transcripts cannot be directly validated or linked to the captured drone using the compromised current state.

The verification results support the claimed security properties under the defined symbolic adversarial model. The session key remains secret, replayed or modified protocol messages fail authentication or key confirmation, and compromise of long-term drone material after session completion does not allow reconstruction of previous session keys. The ProVerif analysis should be interpreted as symbolic validation rather than a computational proof; however, it provides formal support for the protocol’s resistance to impersonation, replay, man-in-the-middle manipulation, and post-compromise session-linkability attacks.

Table 12: Formal Security Validation Summary

Security Property	Validation Method	Result
Session-key secrecy	ProVerif secrecy query on SK	The adversary cannot derive the session key under the symbolic model.
Mutual authentication	Correspondence assertions between Drone and Control Server events	Successful protocol completion implies participation of the intended peer.
Replay resistance	Nonce and timestamp freshness validation	Replayed messages fail freshness or authentication checks.

Man-in-the-middle resistance	Transcript-bound HMAC verification and explicit key confirmation	Modified Diffie–Hellman values or transcript fields cause verification failure.
Forward secrecy	Session key derived from fresh ephemeral X25519 shared secret	Later compromise of long-term material does not reveal previous session keys.
Forward unlinkability	Pseudonym evolution, session-isolated key derivation, and post-capture reveal modeling	Captured current state cannot be used to directly validate or link previous sessions.
Drone-capture resilience	Reveal event for long-term secret and current pseudonym state	Past session keys and previous pseudonym states remain protected after capture.
Unknown key-share and reflection resistance	Role-specific transcript labels and key-confirmation logic	Role confusion and reflected transcript attacks are rejected.

9. Discussion

This section discusses the implications of the proposed secure and forward-unlinkable key agreement protocol with respect to performance, scalability, and practical deployment in UAV networks. Rather than reiterating the numerical results, this section interprets the findings presented in Section 6 and analyzes how the protocol design choices contribute to both security and efficiency.

9.1 Impact of Authenticated Diffie–Hellman with Context Binding

One main design consideration of the proposed protocol is authenticated Diffie-Hellman (X25519) with context binding. The DH contribution provides forward secrecy, while contextual parameters, including location zone, mission code, and time slot, enhance session uniqueness and unlinkability without being treated as cryptographic secrets.

The performance analysis in Tables 5 and 6 indicates that each X25519 operation requires approximately 3-5 ms on a Cortex-M4-class microcontroller and contributes about 6-10 ms to the estimated 7-11 ms cryptographic overhead per session. This overhead remains low compared with normal network latencies of 60-120 ms. One major takeaway from the findings is that forward_secrecy features are typically overlooked when it comes to the authentication protocols of UAVs and IoT devices, as they are believed to bring an increase in computational demand. The study reveals that there is no hindrance to achieving both robust forward secrecy as well as a good degree of unlinkability from a real-time responsiveness point of view. In other words, it is now feasible to provide a privacy-enhancing authentication to all these situations where the number of users connecting is huge and frequent like, for example, a set of flying drones or a mission-critical drone deployment.

9.2 Network-Dominated Latency Versus Cryptographic Cost

The latency evaluation indicates that cryptographic operations are not the dominant contributor to end-to-end handshake delay. Microbenchmarks suggest that HMAC and HKDF operations take less than 1 ms, while the two X25519 operations, public-key generation and shared-secret computation, each require approximately 3-5 ms on a Cortex-M4 platform (see Tables 5 and 6).

Conversely, the end-to-end handshake time is largely determined by wireless communication latency, as confirmed by both the latency histogram and cumulative distribution function (CDF) analysis. This observation validates the protocol's design philosophy: employing lightweight cryptographic primitives while recognizing that network conditions represent the primary bottleneck in UAV communication. Further reductions in cryptographic costs would therefore yield minimal practical benefit, as the current design already satisfies real-time operational constraints.

9.3 Scalability in Large-Scale UAV Networks

The scalability tests show that the protocol's performance stayed pretty much the same when the number of parallel sessions increased from 10 to 10,000. The average verification time only increased slightly and still was less than a millisecond in all tested use cases. This is largely due to the stateless design of the protocol and the protocol's use of cryptographic operations that can be done in constant time.

These experimental results indicate that the protocol can work well in massive UAV networks such as drone swarms, multi-zone surveillance systems, and high-frequency command-and-control areas. The scalability has not been realized at the expense of weakening the security guarantees. In fact, it is simply the result of a limitation on the per-session state and elimination of heavy cryptographic systems.

9.4 Robustness Under Normal and Stressed Conditions

The comparison between Simulation A (normal conditions) and Simulation B (high-load or high-latency conditions), presented in Section 6.5, provides insight into the protocol's robustness. Under normal conditions, all sessions remain within the 200 ms real-time deadlines. Under stressed conditions, approximately 50% of sessions exceed the threshold. This degradation is primarily caused by increased network delay rather than cryptographic processing.

Notably, even under stressed conditions, the protocol remains stable and predictable, with no evidence of computational bottlenecks or cascading failures. The DH-based session key update cost remains constant, confirming that cryptographic mechanisms do not amplify performance degradation in adverse environments. Such predictability is essential for mission planning and quality-of-service guarantees in UAV systems.

9.5 Security–Performance Trade-offs

Every time a session starts, the method uses just one quick Elliptic Curve Diffie, Hellman operation (X25519). It's much more effective than exponentiation or certificate-based PKI infrastructures,

some UAV authentication methods use blockchain-based mechanisms or zero-knowledge proofs, But the design of this one only uses one DH exchange and symmetric cryptography (HMAC and HKDF) to handle cryptographic complexity. A very good compromise was made, which is forward secrecy, strong authentication, and unlinkability combined with only negligible computational effort. Performance test confirms that the design is not only theoretically efficient but also practically suitable for UAV scenarios with limited processing power and energy resources.

Python tests are run merely to ensure the correct functioning of the algorithms. They are not a basis for claiming embedded performance.

9.6 Limitations and Future Work

Despite its advantages, the proposed evaluation has several limitations. The network-latency model assumes independent and identically distributed delays and does not explicitly account for UAV mobility patterns, link asymmetry, or adversarial jamming. Future work should incorporate mobility-aware and adversarial network models to further stress-test the protocol.

Additionally, although micro benchmarking provides reliable estimates of cryptographic cost, deployment of heterogeneous UAV hardware platforms would provide deeper insight into real-world performance. Extending the protocol to support post-quantum key exchange mechanisms and integrating it with secure routing or intrusion-detection systems are also promising directions for future research.

10. Conclusion

This paper presents a lightweight forward-unlinkable authenticated key agreement protocol for securing UAV communication in Internet of Drones environments with strict real-time and resource requirements. The proposed scheme provides mutual authentication, session-key secrecy, forward secrecy, and forward unlinkability by combining

authenticated Diffie-Hellman (X25519) with lightweight symmetric cryptographic primitives, without relying on heavy infrastructures such as PKI or blockchain. Under the assumed Dolev-Yao adversary model with drone-capture capability, the protocol provides privacy-preserving security properties supported by symbolic analysis and benchmark-based performance evaluation, while requiring only a few milliseconds of computational overhead per session.

The performance analysis indicates that the protocol has low cryptographic overhead and that end-to-end session setup is dominated by network delay rather than computation. Scalability experiments confirm stable behavior under large numbers of concurrent sessions, and stress testing under poor network conditions shows graceful performance degradation without instability.

Besides security features, the design is also privacy-aware. Low cryptographic reconnection cost results in frequent secure re-connections and this is very important for forward unlinkability, the capability to be anonymous, and keeping privacy for very long period for users of UAV networks where reconnection is frequently happening.

Besides strong security, efficiency and scalability features, the novel protocol is highly suitable for mass UAV networks, drone swarms, mission-critical command-and-control tasks in IoD environments. This research should look for mobility-aware latency models, real-world UAV hardware implementation, as well as the mechanisms of post-quantum secure key exchange.

Declarations

Ethics Approval and Consent to Participate

This study did not involve human participants, human data, or human biological material. Therefore, ethical approval and consent to participate were not required.

Declaration of Interest

The author declares no known financial or personal relationships that could have influenced the work reported in this study

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

Data generated or analyzed during this study are available from the corresponding author upon reasonable request.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work, no generative AI or AI-assisted technologies were used in the writing process. The author takes full responsibility for the content of the published work.

REFERENCES

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095
2. Amin, R., Islam, S. K. H., Biswas, G. P., Khan, M. K., Leng, L., & Kumar, N. (2016). Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Computer Networks, 101, 42-62. https://doi.org/10.1016/j.comnet.2016.01.006
3. Bekmezci, I., Sahingoz, O. K., & Temel, Ş. (2013). Flying ad hoc networks (FANETs): A survey. Ad Hoc Networks, 11(3), 1254-1270. https://doi.org/10.1016/j.adhoc.2012.12.004

4. Bera, B., Chattaraj, D., & Das, A. K. (2020). Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment. Computer Communications, 153, 229-249. https://doi.org/10.1016/j.comcom.2020.02.011
5. Bera, B., Saha, S., Das, A. K., Kumar, N., Lorenz, P., & Alazab, M. (2020). Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled Internet of Drones environment. IEEE Transactions on Vehicular Technology, 69(8), 9097-9111. https://doi.org/10.1109/TVT.2020.3000576
6. Choi, J., Son, S., Kwon, D., & Park, Y. (2025). A PUF-based secure authentication and key agreement scheme for the Internet of Drones. Sensors, 25(3), Article 982. https://doi.org/10.3390/s25030982
7. Choi, J., Son, S., Kwon, D., & Park, Y. (2025). A PUF-based secure authentication and key agreement scheme for the Internet of Drones. Sensors, 25(3), 982. https://doi.org/10.3390/s25030982
8. Farash, M. S., Turkanović, M., Kumari, S., & Hölbl, M. (2016). An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. Ad Hoc Networks, 36, 152-176. https://doi.org/10.1016/j.adhoc.2015.05.014
9. Gope, P., & Sikdar, B. (2019). Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE Internet of Things Journal, 6(1), 580-589. https://doi.org/10.1109/JIOT.2018.2846299
10. Gupta, L., Jain, R., & Vaszkun, G. (2016). Survey of important issues in UAV communication networks. IEEE Communications Surveys & Tutorials, 18(2), 1123-1152. https://doi.org/10.1109/COMST.2015.2495297
11. Han, P., Sui, A., & Wu, J. (2025). Lightweight secure communication supporting batch authentication for UAV swarm. Drones, 9(2), Article 139. https://doi.org/10.3390/drones9020139
12. Han, P., Sui, A., & Wu, J. (2025). Lightweight secure communication supporting batch authentication for UAV swarm. Drones, 9(2), 139. https://doi.org/10.3390/drones9020139
13. Hayat, S., Yanmaz, E., & Muzaffar, R. (2016). Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. IEEE Communications Surveys & Tutorials, 18(4), 2624-2661. https://doi.org/10.1109/COMST.2016.2560343
14. Jeong, J. Y., Byun, J. W., & Jeong, I. R. (2022). Key agreement between user and drone with forward unlinkability in Internet of Drones. IEEE Access, 10, 17134-17144. https://doi.org/10.1109/ACCESS.2022.3150035
15. Jeong, J. Y., Kang, H. W., & Jeong, I. R. (2024). Concurrent two-party key exchange with forward unlinkability in Internet of Drones. IEEE Access, 12, 77250-77256. https://doi.org/10.1109/ACCESS.2024.3404850
16. Jeong, J. Y., Kang, H. W., & Jeong, I. R. (2024). Concurrent two-party key exchange with forward unlinkability in Internet of Drones. IEEE Access, 12, 77250–77256. https://doi.org/10.1109/ACCESS.2024.3404850
17. Lee, C.-C., Li, C.-T., Chiu, S.-T., & Lai, Y.-M. (2015). A new three-party-authenticated key agreement scheme based on chaotic maps without password table. Nonlinear Dynamics, 79(4), 2485-2495. https://doi.org/10.1007/s11071-014-1827-x
18. Li, Z., Ju, Z., Zhao, H., Wei, Z., & Lan, G. (2025). A lightweight certificateless authenticated key agreement scheme based on Chebyshev polynomials for the Internet of Drones. Sensors, 25(14), 4286. https://doi.org/10.3390/s25144286
19. Srinivas, J., Das, A. K., Kumar, N., & Rodrigues, J. J. P. C. (2019). TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment. IEEE Transactions on Vehicular Technology, 68(7), 6903-6916. https://doi.org/10.1109/TVT.2019.2911672
20. Turkanović, M., Brumen, B., & Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. Ad Hoc Networks, 20, 96-112. https://doi.org/10.1016/j.adhoc.2014.03.009
21. Wazid, M., Das, A. K., & Lee, J.-H. (2018). Authentication protocols for the Internet of Drones: Taxonomy, analysis and future directions. Journal of Ambient Intelligence and Humanized Computing, 1-10. https://doi.org/10.1007/s12652-018-1006-x
22. Xie, Q., & Wang, H. (2025). PUF-based secure and efficient anonymous authentication protocol for UAV towards cross-domain environments. Drones, 9(4), 260. https://doi.org/10.3390/drones9040260
23. Xie, Q., Zhao, J., & Yu, X. (2013). Chaotic maps-based three-party password-authenticated key agreement scheme. Nonlinear Dynamics, 74(4), 1021-1027. https://doi.org/10.1007/s11071-013-1020-7
24. Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Karimipour, H., Srivastava, G., & Aledhari, M. (2021). Enabling drones in the Internet of Things with decentralized blockchain-based security. IEEE Internet of Things Journal, 8(8), 6406-6415. https://doi.org/10.1109/JIOT.2020.3015382
25. Zhang, Y., He, D., Li, L., & Chen, B. (2020). A lightweight authentication and key agreement scheme for Internet of Drones. Computer Communications, 154, 455-464. https://doi.org/10.1016/j.comcom.2020.02.067

26. Akil, M., Martucci, L. A., & Hoepman, J.-H. (2023). Non-interactive privacy-preserving Sybil-free authentication scheme in VANETs. In Inaugural International Symposium on Vehicle Security and Privacy (VehicleSec 2023). Internet Society. https://doi.org/10.14722/vehiclesec.2023.23007
27. Akil, M., Martucci, L. A., & Hoepman, J.-H. (2024). A secure and privacy-preserving authentication scheme with a zero-trust approach to vehicle renting in VANETs. In Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024) (pp. 114–127). SCITEPRESS. https://doi.org/10.5220/0012759600003767
28. Akil, M., Naskar, S., Martucci, L. A., & Hoepman, J.-H. (2023). A privacy-preserving approach to vehicle renting and driver accountability in VANETs. In Privacy and Identity Management: Sharing in a Digital World—18th IFIP WG 9.2, 9.6/11.7, 11.6 International Summer School, Privacy and Identity 2023 (pp. 192–210). Springer. https://doi.org/10.1007/978-3-031-57978-3_13
29. Akil, M., Mancini, L. V., & Venturi, D. (2019). Multi-covert channel attack in the cloud. In 6th International Conference on Software Defined Systems (SDS 2019) (pp. 160–165). IEEE. https://doi.org/10.1109/SDS.2019.8768563