Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.083880

ARTICLE

A Lightweight Time-Indexed Secure Communication Framework with Intrusion Detection Modeling for Resource-Constrained UAV Swarm Networks

Li-Woei Chen[1], Kun-Lin Tsai[2,*], Fang-Yie Leu[3], Chao-Tung Yang[3,4,5] and Wei-Zong Liang[2]

[1]Department of Computer and Information Sciences, Chinese Military Academy, Kaohsiung, Taiwan
[2]Department of Electrical Engineering, Tunghai University, Taichung, Taiwan
[3]Department of Computer Science, Tunghai University, Taichung, Taiwan
[4]Research Center for Smart Sustainable Circular Economy, Tunghai University, Taichung, Taiwan
[5]Department of Medical Research, Kuang Tien General Hospital, Taichung, Taiwan

*Corresponding Author: Kun-Lin Tsai. Email: kltsai@thu.edu.tw

Received: 12 April 2026; Accepted: 11 June 2026; Published: 27 July 2026

ABSTRACT: Unmanned aerial vehicle (UAV) swarm networks are increasingly deployed in surveillance, disaster response, and intelligent transportation systems, where secure and efficient communication is critical under resource-constrained environments. However, conventional public-key-based security mechanisms introduce excessive computational overhead, while standalone intrusion detection systems are insufficient to defend against dynamic and multi-vector attacks in swarm networks. To address these challenges, in this paper, a lightweight time-indexed secure communication framework with intrusion detection modeling (TSCID) is proposed for resource-constrained UAV swarm networks. The proposed TSCID integrates a time-indexed session key derivation mechanism with lightweight authenticated encryption to ensure confidentiality, integrity, replay resistance, and session-key isolation with low computational cost. A security and communication-overhead analysis under standard symmetric-key cryptographic assumptions is conducted to evaluate the practicality and lightweight characteristics of the proposed framework. To enhance resilience against network-level attacks, an intrusion detection module based on deep neural networks is incorporated and optimized through structured pruning, enabling real-time anomaly detection on edge-class UAV devices. Experimental results demonstrate that TSCID reduces communication latency by up to 35% and energy consumption by nearly 30% compared with conventional public-key-based security mechanisms, while the lightweight intrusion detection model achieves over 92% detection accuracy with less than 4% false positives. Analytical and experimental results confirm that the proposed framework provides an efficient and secure solution for real-time UAV swarm communication under strict resource constraints.

KEYWORDS: UAV swarm networks; time-indexed secure communication; lightweight cryptography; intrusion detection system; structured pruning; resource-constrained networks

1 Introduction

Unmanned aerial vehicles (UAVs) have been widely deployed in surveillance, disaster response, precision agriculture, and intelligent transportation systems due to their flexibility and rapid deployment capability. Recently, UAV swarm networks have attracted significant attention because cooperative operation among multiple UAVs enables wider coverage, higher mission efficiency, and improved robustness compared with single-UAV systems. In swarm environments, UAV nodes communicate frequently with each other and

with a ground control station (GCS), forming a highly dynamic wireless network that requires reliable and secure communication mechanisms.

Despite abovementioned advantages, UAV swarm networks face critical security challenges. UAV platforms are typically resource-constrained in terms of computation capability, memory size, and battery capacity, making it difficult to deploy conventional public-key–based security mechanisms. Conventional cryptographic protocols can provide strong confidentiality and authentication, but they often require considerable computational cost and communication latency, which may significantly degrade the performance of real-time swarm operations. Moreover, the dynamic topology and open wireless environment make UAV swarms vulnerable to various cyberattacks, including replay attacks, spoofing, man-in-the-middle attacks, and denial-of-service attacks.

Intrusion detection systems (IDS) have been introduced to enhance the resilience of UAV networks against malicious behaviors. Recent studies have applied machine learning [1,2] and deep neural network models [3] to detect abnormal traffic patterns with high accuracy. However, standalone IDS approaches are often limited in their ability to provide comprehensive protection, because they operate independently from the communication protocol and cannot prevent attacks at the cryptographic level. In addition, deep learning–based IDS models usually require huge computational resources, which makes direct deployment on edge-class UAV devices challenging.

Accordingly, there is a need for an integrated security framework that simultaneously ensures lightweight secure communication and real-time intrusion detection while maintaining low computational overhead suitable for resource-constrained UAV platforms. Such a framework should provide strong security guarantees, efficient key management, and scalable performance for resource-constrained UAV swarms, while allowing intrusion detection to operate in coordination with the communication protocol.

To reach the above-mentioned goals, in this paper, a lightweight Time-indexed Secure Communication framework with Intrusion Detection modeling (TSCID) is proposed for resource-constrained UAV swarm networks. The proposed framework integrates a time-indexed session key derivation scheme with lightweight authenticated encryption to ensure confidentiality, integrity, replay protection, and session-key isolation with minimal computational cost. A mathematical model of the time-indexed key schedule and packet authentication process is developed to formally describe the communication mechanism and analyze security properties.

In addition to the secure communication layer, the proposed TSCID also develops an intrusion detection module based on deep neural networks to monitor network traffic and identify abnormal behaviors. To enable deployment on edge-class UAV devices, the IDS model is optimized using structured pruning techniques to reduce parameter size and computational complexity while maintaining detection accuracy. The integration of time-indexed communication and intrusion detection modeling forms a dual-layer security architecture that enhances both cryptographic protection and behavioral anomaly detection.

Unlike conventional studies that focus exclusively on either lightweight secure communication or standalone intrusion detection mechanisms, the proposed TSCID framework adopts a deployment-oriented cross-layer integration strategy for resource-constrained UAV swarm environments. The primary contribution of this work lies not in the development of fundamentally new cryptographic primitives or proof methodologies, but rather in the coordinated integration of lightweight authenticated communication, time-indexed session-key management, and behavioral anomaly detection within a unified architecture optimized for real-time UAV swarm operation under constrained computational and energy budgets.

The main contributions of this paper are summarized as follows:

1. A lightweight cross-layer security framework TSCID is developed for resource-constrained UAV swarm networks by integrating time-indexed secure communication and intrusion detection mechanisms within a unified deployment-oriented architecture.

2. A lightweight communication and authentication mechanism based on time-indexed session-key derivation and AEAD-based packet protection is designed to provide low-overhead secure communication with replay protection and session-level key isolation suitable for real-time UAV swarm operation.
3. A security analysis under standard symmetric-key cryptographic assumptions is conducted to evaluate confidentiality, integrity, authentication, replay resistance, and session-key isolation properties of the proposed framework.
4. An IDS framework operating on authenticated traffic is integrated with a structured-pruning-based optimization strategy to improve anomaly detection reliability while reducing computational complexity for edge-class UAV deployment.
5. Experimental evaluations on Raspberry Pi platforms demonstrate the feasibility of the proposed framework in terms of communication latency, energy efficiency, and lightweight intrusion detection performance under resource-constrained environments.

The remainder of this paper is organized as follows. Section 2 reviews related work on secure communication, lightweight cryptography, and intrusion detection in UAV and resource-constrained networks. Section 3 formulates the system model and problem definition, including the network architecture, threat model, and security requirements. Section 4 presents the proposed TSCID framework, focusing on the time-indexed secure communication mechanism. Section 5 develops the mathematical modeling and provides a comprehensive security and complexity analysis of the proposed scheme. Section 6 introduces the intrusion detection modeling and its optimization via structured pruning for resource-constrained UAV devices, forming a dual-layer security architecture. Section 7 evaluates the proposed framework through experimental results, including performance and detection effectiveness. Finally, Section 8 concludes this paper.

2 Related Studies

Secure communication in UAV networks has been widely studied due to the increasing deployment of UAVs in mission-critical applications [4,5]. Traditional approaches mainly rely on public-key infrastructure (PKI) [6] and certificate-based authentication [7], which provide strong security but may introduce significant computational and communication overhead, making them unsuitable for resource-constrained UAV platforms [8,9]. To improve efficiency, recent studies have explored lightweight cryptographic schemes based on symmetric-key primitives. For example, Cecchinato et al. [10] demonstrated the feasibility of lightweight Advanced Encryption Standard (AES)-based encryption for real-time UAV multimedia transmission. Alexan et al. [11] investigated secure communication mechanisms in UAV-assisted reconnaissance scenarios. These approaches reduce computational cost; however, many of them still rely on static or infrequently updated keys, which may expose UAV networks to replay attacks and key compromise in dynamic swarm environments. More advanced security solutions, such as trust-based communication models and distributed frameworks, have also been proposed to enhance system resilience [12,13]. However, these methods may also increase system complexity and communication overhead, limiting their applicability in large-scale UAV swarms. In particular, the lack of efficient and scalable dynamic key management mechanisms remains a key limitation.

Lightweight cryptography has emerged as a key enabler for resource-constrained environments. Different from conventional cryptographic algorithms, lightweight schemes are designed to minimize computational complexity, memory usage, and energy consumption while maintaining security. For example, lightweight AES implementations have been applied in UAV communication scenarios to reduce processing overhead while preserving data confidentiality [3,10]. However, such approaches still inherit structural limitations from conventional cryptographic designs and may not fully meet the stringent

efficiency requirements of highly dynamic UAV swarm networks. More recently, dedicated lightweight authenticated encryption schemes have been developed to address these limitations. In particular, the Ascon family [14], selected as the NIST standard for lightweight cryptography, provides efficient authenticated encryption with associated data and hash functions tailored for constrained environments [15]. Despite these advances, existing lightweight cryptographic approaches often focus solely on encryption efficiency and lack integration with adaptive key management mechanisms.

In order to enhance the security of UAV and IoT networks by identifying malicious activities and abnormal communication patterns, IDS has been adopted in UAV systems [16–18]. Recent studies have focused on applying deep learning techniques to improve IDS effectiveness. Deep neural networks (DNNs) have demonstrated superior capability in learning complex traffic patterns and detecting various cyberattacks, such as denial-of-service, spoofing, and man-in-the-middle attacks. For example, Hassler et al. [19] proposed a cyber-physical IDS for UAV systems, while Ihekoronye et al. [20] introduced an explainable IDS framework for UAV networks. Mohammed and Fourati [21] introduced a taxonomy of UAV-related IDS datasets and provides practical guidelines for selecting appropriate datasets under different operational scenarios. Graph-based and swarm-level intrusion detection methods have been explored to capture the distributed nature of UAV networks, further improving detection accuracy and scalability [22]. Despite these advances, most existing IDS solutions are designed as standalone systems and operate independently from communication protocols. As a result, they are difficult to prevent attacks at the cryptographic level and may introduce additional processing overhead. Furthermore, the lack of tight integration between secure communication mechanisms and intrusion detection models limits the overall effectiveness of system-level security.

Deploying IDS on resource-constrained platforms requires careful consideration of computational efficiency and memory usage. Although deep learning–based IDS models have demonstrated high detection accuracy, their complexity often makes them unsuitable for direct deployment on edge devices with limited processing capability and energy budget. Model compression and pruning have been widely studied to reduce the computational overhead of deep neural networks. Early work by Han et al. [23] introduced network pruning techniques to remove redundant connections and reduce model size without significantly affecting accuracy. Similarly, Luo et al. [24] proposed filter-level pruning methods to further optimize convolutional neural networks. More recent studies have developed structured and adaptive pruning approaches to improve efficiency while maintaining robust performance, enabling deep learning models to operate effectively in resource-constrained environments [25,26]. Existing optimization techniques are generally applied independently of communication protocols and do not consider the interaction between secure communication and intrusion detection processes. This motivates the integration of structured pruning–based IDS optimization within the proposed TSCID framework, enabling real-time anomaly detection with reduced computational cost for edge-level UAV systems.

Recent studies have witnessed extensive explorations into enhancing the resilience of cooperative aerial swarms from both network routing and machine learning perspectives. For network-layer data protection, ref. [27] introduced LDST-UAVS, a lightweight and traceable secure data transmission protocol designed for infrastructure-less, multi-hop UAV environments. The protocol orchestrates a scalable aggregate verification mechanism derived from biorthogonal codes, allowing untrusted routing nodes along the path to proximally isolate data injection attacks and track malicious sources with linear time complexity. Moving beyond static cryptographic boundaries to adaptive threat intelligence, ref. [28] presented a privacy-preserving security framework that integrates Federated Learning with explainable AI techniques to defend critical cyber-physical infrastructures. Their approach utilizes decentralized machine learning to detect airborne sensor attacks while uniquely implementing SHAP analysis on the aggregated global model, providing

operators with granular interpretations regarding how specific telemetry variables dictate the swarm's runtime reconfiguration and anomaly classification.

Although prior studies have investigated lightweight secure communication and intrusion detection independently, limited attention has been devoted to the coordinated interaction between authenticated communication protocols and behavioral anomaly detection in resource-constrained UAV swarm environments. Existing IDS approaches typically analyze raw or partially validated traffic, while secure communication frameworks often operate without considering their interaction with higher-level anomaly detection processes. This motivates the proposed TSCID framework, which integrates lightweight authenticated communication and IDS-based behavioral analysis within a unified deployment-oriented architecture.

3 System Models and Problem Formulation

3.1 UAV Network Model

The considered system is a UAV swarm network consisting of a GCS and a set of UAV nodes denoted as $\mathcal{N} = \{n_1, n_2, \ldots, n_m\}$. The GCS is responsible for mission coordination, command dissemination, and data aggregation, while UAV nodes cooperate through peer-to-peer communication to perform distributed tasks. The swarm forms a dynamic wireless network with a mesh-like topology, where communication links are established among UAVs and between UAVs and the GCS. The communication topology is represented as a dynamic graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of communication links between nodes. Each node $n_i \in \mathcal{N}$ is equipped with minimal computational capability C_i, memory capacity M_i, and energy budget E_i. And all communications and security mechanisms must satisfy the constraint in Eq. (1).

$$\begin{cases} 0 < C_i \le C_{sec}\left(n_i\right)_k \\ 0 < M_i \le M_{sec}\left(n_i\right)_k \\ 0 < E_i \le E_{sec}\left(n_i\right)_k \end{cases} \quad \forall n_i \in \mathcal{N} \tag{1}$$

where C_{sec}, M_{sec}, and E_{sec} denote the computational, memory, and energy cost of the security mechanism, respectively, and k denotes the k^{th} transmission. Since each UAV is equipped with limited onboard resources, including constrained computational capability, memory, and battery capacity, and these resources deplete with every transmission, it is essential to develop a lightweight secure communication mechanism.

3.2 Communication and Threat Models

Time is divided into discrete intervals of length Δ, and a time index T is defined as $T = \lfloor t/\Delta \rfloor$. At each time index T, a node n_i sends a message $m \in \mathcal{M}$ to node n_j in the form of a packet $p_{t_\alpha} = \langle ID_i, ID_j, \mathrm{T}, ctr_i, C, \tau \rangle$, where ctr_i is a monotonically increasing counter, and (C, τ) denote the ciphertext and authentication tag, respectively. Accordingly, the communication process can be modeled as a mapping shown in Eq. (2).

$$f_{comm}: \mathcal{M} \times \mathcal{N} \times \mathcal{N} \longrightarrow \mathcal{P} \tag{2}$$

where $\mathcal{P}$ denotes the set of valid transmitted packets. The communication objective is to ensure secure and efficient message delivery under dynamic network conditions.

The UAV swarm operates in an open and potentially adversarial wireless environment. The current threat model primarily focuses on communication-layer adversaries. It is assumed that adversaries can eavesdrop, intercept, replay, modify, and inject packets into the communication channel, but they do not physically compromise UAV hardware or directly extract long-term secret keys stored in legitimate UAV

nodes or the GCS during protocol operation. Therefore, the proposed framework does not explicitly consider physical node capture, hardware tampering, or large-scale swarm compromise scenarios. Under this threat model, an adversary $\mathcal{A}$ is assumed to have the capability to perform the following operations:

- Eavesdropping attacks: the adversary passively observes transmitted packets to extract sensitive information, which can be modeled as $\mathcal{A}(p) \rightarrow p$.
- Replay attacks: previously transmitted packets are resent to disrupt system operation, modeled as $\mathcal{A}(p, T) \rightarrow \mathcal{A}(p, T')$, where $T' \neq T$.
- Spoofing attacks: the adversary impersonates a legitimate UAV node n_i by generating forged packets, and sends to UAV node n_j or GCS $\mathcal{G}$, modeled as $\mathcal{A}_{n_i}(p) \rightarrow n_j/\mathcal{G}$.
- Man-in-the-middle (MITM) attacks: the adversary intercepts and modifies communication between nodes, modeled as $\mathcal{A}(p) \rightarrow \tilde{p}$.
- Denial-of-service (DoS) attacks: the adversary injects excessive or malicious traffic into the network to degrade system performance, modeled as $\mathcal{A}(\varnothing) \rightarrow \{p_1, p_2, \ldots, p_k\}$.

3.3 Security Requirements

To ensure reliable and secure UAV swarm operation, the communication framework must satisfy confidentiality, integrity, authentication, replay protection, and session-key isolation. Confidentiality indicates that the transmitted data should be protected from unauthorized access, and integrity denotes messages must not be altered during transmission. Authentication means communicating entities must be verified as legitimate nodes. Replay protection represents previously transmitted messages should not be accepted again. Session-key isolation means that compromise of one session key should not directly reveal other session keys corresponding to different communication intervals under the defined threat assumptions. The abovementioned five models are defined as follows.

- Confidentiality: $\Pr[\mathcal{A}(\mathrm{C}) \rightarrow \mathrm{m}] \approx 0$
- Integrity: $\Pr[\mathrm{Verify}(\tilde{p}) = \mathrm{true} \ \wedge \tilde{p} \neq p] \approx 0$
- Authentication: $\Pr[\mathcal{A}$ impersonates $n_i] \approx 0$
- Replay Protection: $\forall p = (T, ctr_i)$, $p$is accepted only once
- session-key isolation: $K_T \perp K_{T'}, \forall T \neq T'$

These properties ensure that communication remains secure even under adversarial conditions.

3.4 Problem Formulation

Given the above system and threat model, the objective is to design a secure and efficient communication framework for UAV swarm networks that satisfies both security and resource constraints. Specifically, the problem can be formulated as follows: how to develop a lightweight communication mechanism that (i) provides strong security guarantees including confidentiality, integrity, authentication, and replay protection, (ii) supports efficient and scalable key management under dynamic network conditions, and (iii) integrates intrusion detection capabilities with minimal additional computational overhead. This can be formulated as an optimization problem:

$$\text{Minimize } \mathrm{O} = \alpha \cdot C_{sec} + \beta \cdot E_{sec} + \gamma \cdot L_{comm}$$
$$\text{subject to } Security(p) \geq S_{\min} \text{ and } Latency \leq L_{\max}$$

where $\cdot C_{sec}$, E_{sec}, and L_{comm} denote computational cost, energy consumption, and communication latency, respectively, and α, β, γ are weighting factors. $S_{\min}$ is the minimal security requirement shown in Section 3.3, and $L_{\max}$ is the allowable maximal communication latency. The challenge is to achieve an optimal balance

between security strength and system efficiency in resource-constrained UAV swarm environments. To address this problem, a unified framework is required that combines lightweight cryptographic design, dynamic key evolution, and efficient intrusion detection suitable for real-time UAV swarm environments. This motivates the proposed TSCID framework described in the following section.

4 Proposed TSCID Framework

4.1 TSCID Framework

The proposed framework is designed to provide comprehensive security for UAV swarm networks by integrating a lightweight secure communication protocol with an intrusion detection mechanism, forming a dual-layer security architecture. Its primary objective is to ensure end-to-end protection of control messages and mission-critical data, while preserving the low computational and energy footprint required by resource-constrained UAV platforms.

At the communication layer, the system employs the TSCID protocol. TSCID leverages a dynamic session key mechanism derived from master keys shared with the GCS and a synchronized time index, enabling the establishment of pairwise session keys between UAV nodes. This ensures that each communication interval is protected by a distinct key, preventing key reuse and limiting the impact of potential compromise. Ascon-Hash is used for key derivation, while Ascon-AEAD provides authenticated encryption. Compared with traditional PKI-based approaches, TSCID achieves significant reductions in latency and energy consumption while maintaining strong cryptographic guarantees.

On top of this secure communication foundation, the framework incorporates a deep neural network–based IDS, which monitors authenticated UAV network traffic and detects attacks such as denial-of-service, spoofing, and man-in-the-middle intrusions. By operating on validated packets, the IDS avoids processing unauthenticated or corrupted data, thereby improving both efficiency and detection reliability. To accommodate limited onboard resources, pruning techniques are applied to reduce the computational complexity of the IDS without significantly sacrificing detection accuracy, making it suitable for deployment on UAV edge devices.

The interaction of these modules is illustrated in Fig. 1. Communication begins with TSCID, which secures message delivery through encryption and authentication. The resulting traffic is then analyzed by the IDS to identify abnormal patterns at the behavioral level. By combining these layers, the framework ensures that security is preserved across both communication and network monitoring dimensions.

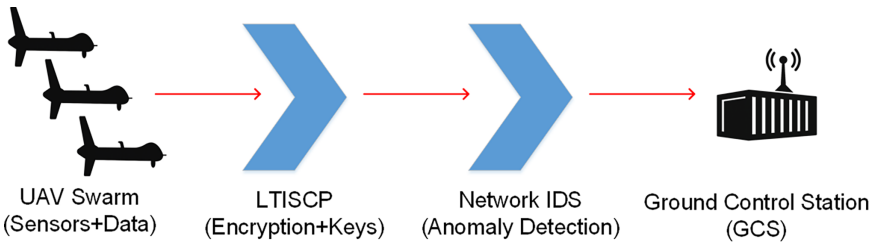

Figure 1: Workflow of the proposed framework.

4.2 Three-Party Time-Indexed Secure Communication Protocol

In this paper, a time-index communication protocol is developed to meet security requirements for three-party communication among UAV U_i, UAV U_j, and GCS $\mathbb{G}$. Table 1 defines the notation used in the protocol.

Table 1: Notations of three-party communication protocol.

Notations	Descriptions
K_i	Pre-shared long-term master key between U_i and $\mathbb{G}$
K_j	Pre-shared long-term master key between U_j and $\mathbb{G}$
T	Current time index, which is defined as $T = \lfloor t/\Delta \rfloor$
Δ	Time slot duration
r_i, r_j, r_g	Random nonces generated by U_i, U_j, and $\mathbb{G}$
ctr_i, ctr_j	per-node monotonic counters
$\mathrm{H}(\cdot)$	Secure hash function
AEAD. $\mathrm{Enc}_K(N, A, M)$	AEAD encryption under key K, nonce N, associated data A, plaintext M
AEAD. $\mathrm{Dec}_K(N, A, C)$	AEAD decryption under key K, nonce N, associated data A, ciphertext C
$\|$	Data/Message concatenation
$\perp$	Data/Packet rejection

To prevent replay attacks, this paper defines the allowable clock drift as δ; thus, the valid time window is defined as

$$T' \in [T - \delta, T + \delta] \tag{3}$$

The synchronization tolerance parameter δ is introduced to accommodate practical clock drift and transmission delay variations among UAV nodes. In the current framework, time synchronization is assumed to be supported by existing infrastructure mechanisms such as GPS timing, periodic beacon synchronization, or GCS-assisted coordination. The proposed work primarily focuses on lightweight secure communication and does not explicitly optimize distributed synchronization protocols under GPS-denied or highly intermittent network environments.

The proposed protocol employs a two-layer key mechanism. The first layer is the UAV-GCS Long-term Keys K_i and K_j, and the second layer is the pairwise session key $SK_{ij}^{(T)}$ established with the assistance of the GCS and used for communication between U_i and U_j within the time index T. To ensure that a unique key is used for each time interval, the GCS generates the session key as Eq. (4).

$$SK_{ij}^{(T)} = \mathrm{H}(K_i \| K_j \| ID_i \| ID_j \| \mathrm{T} \| r_g) \tag{4}$$

In Eq. (4), r_g generated by $\mathbb{G}$ represents a fresh nonce for the current session. The inclusion of the time index T ensures that different time intervals correspond to distinct session keys, thereby maintaining cryptographic freshness.

The protocol can be divided into three phases, including session establishment, mutual authentication, and secure data transmission. The proposed protocol is designed based on three key considerations. First, time-indexed key evolution is introduced to ensure that each communication interval uses a distinct key, thereby reducing the impact of key compromise. Second, the involvement of the GCS enables efficient and secure distribution of pairwise session keys without requiring expensive public-key operations. Third, the use of lightweight AEAD ensures both confidentiality and integrity while maintaining low computational overhead suitable for resource-constrained UAV nodes.

Phase 1: Session establishment

Step 1: U_i creates a session establishment request message $M_1 = (ID_i, ID_j, T_1, r_i, ctr_i)$, and encrypts M_1 by using K_i, $C_1 = \text{AEAD.Enc}_{K_i}(N_{i1}, A_1, M_1)$, where $A_1 = (ID_i, ID_j, T_1)$.

Step 2: U_i sends C_1 to $\mathbb{G}$. After decryption, $\mathbb{G}$ checks whether (1) T_1 falls within the valid time window; (2) ctr_i has not been repeated; and (3) tag is correct or not. Once the verification is successful, the protocol proceeds to the next step.

Step 3: $\mathbb{G}$ generates a fresh nonce r_g, and calculates $SK_{ij}^{(T)} = \text{H}(K_i||K_j||ID_i||ID_j||\text{T}_1||r_g)$.

Step 4: $\mathbb{G}$ generates a message $M_2 = (ID_i, ID_j, T_2, r_i, r_g, SK_{ij}^{(T)})$, and encrypts M_2 by using K_j, $C_2 = \text{AEAD.Enc}_{K_j}(N_{g2}, A_2, M_2)$, where $A_2 = (ID_i, ID_j, T_2)$.

Step 5: $\mathbb{G}$ sends C_2 to U_j. After decryption, U_j checks whether (1) T_2 falls within the valid time window; (2) r_i is a fresh nonce; and (3) M_2 is from legal $\mathbb{G}$. Upon successful verification, the protocol proceeds to the next step.

Step 6: Similarly, U_j constructs a request message M_1' following the same format as M_1, and sends to $\mathbb{G}$, in which the r_j is included, and performs similar steps with Step 1 to Step 2.

Step 7: $\mathbb{G}$ generates a message $M_3 = (ID_i, ID_j, T, r_j, r_g, SK_{ij}^{(T)})$, and encrypts M_3 by using K_i, $C_3 = \text{AEAD.Enc}_{K_i}(N_{g3}, A_3, M_3)$, where $A_3 = (ID_i, ID_j, T_3)$.

Step 8: $\mathbb{G}$ sends C_3 to U_i. After decryption, U_i checks whether (1) T_3 falls within the valid time window; (2) r_j is a fresh nonce; and (3) M_3 is from legal $\mathbb{G}$. Once the verification is successful, both U_i and U_j have the same session key $SK_{ij}^{(T)}$.

Phase 2: Mutual authentication

Step 9: U_j generates r_j', and calculates $M_4 = (ID_j, ID_i, T_4, r_j', ctr_j)$. Then encrypts M_4 by using $SK_{ij}^{(T)}$, $C_4 = \text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{j4}, A_4, M_4)$, where $A_4 = (ID_j, ID_i, T_4, r_i)$.

Step 10: U_j sends C_4 to U_i. U_i decrypts C_4, and checks whether (1) T_4 falls within the valid time window; (2) r_i is identical to the value generated in step 1. Once the verification is successful, the protocol proceeds to the next step.

Step 11: U_i sends r_j' back along with its own confirmation information $M_5 = (ID_i, ID_j, T_5, r_j', ack_i, ctr_i')$, where $ack_i = H(r_i||r_j||SK_{ij}^{(T)}||T_5)$. U_i encrypts M_5 by using $SK_{ij}^{(T)}$, $C_5 = \text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{i5}, A_5, M_5)$, where $A_5 = (ID_i, ID_j, T_5)$.

Step 12: U_i sends C_5 to U_j. U_j decrypts C_5, and checks whether (1) T_5 falls within the valid time window; (2) r_j' is identical to the value generated in step 9; and (3) ack_i is successfully verified using the same parameters. Upon successful verification, the mutual authentication process is done.

Phase 3: Secure data transmission

Upon successful authentication, U_i and U_j utilize the session key $SK_{ij}^{(T)}$ to transmit data within the time interval T. For any message m, U_i sends the following to U_j:

The data message structure is defined as Eq. (5).

$$M_{data} = (ID_i, ID_j, T_x, ctr_i^{(d)}, m) \tag{5}$$

The corresponding ciphertext is generated using $SK_{ij}^{(T)}$, and is defined as Eq. (6).

$$C_{data} = \text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{id}, A_{data}, m) \tag{6}$$

where the A_{data} is defined as:

$$A_{data} = (ID_i, ID_j, T_x, ctr_i^{(d)}) \tag{7}$$

Before decryption, the receiver must perform the following checks: (1) whether T_x falls within the valid time window; (2) whether the counter $ctr_i^{(d)}$ has not been previously used; and (3) whether the authentication tag is valid. If any of the above checks fail, the system outputs data rejection $\perp$.

The security of the proposed protocol relies on three main factors. First, the use of AEAD ensures that any modification to the ciphertext or associated data will be detected during decryption. Second, the inclusion of time index T and counter ctr prevents replay attacks by enforcing freshness of each packet. Third, the session key $SK_{ij}^{(T)}$ is derived using both long-term keys and fresh nonce r_g, ensuring key uniqueness across sessions and time intervals.

Compared with conventional PKI-based approaches, the proposed protocol reduces computational overhead by relying only on symmetric-key primitives and hash functions. Each communication round requires a constant number of hash and encryption operations, resulting in $O(1)$ computational complexity per message. This makes the protocol suitable for real-time UAV swarm communication under strict resource constraints.

5 Modeling and Security Analysis

5.1 Mathematical Model of TSCID

To formally analyze the proposed TSCID framework, we model its key components, including time indexing, key evolution, and secure packet generation. Time is divided into discrete intervals of duration Δ, and the time index is defined as $T = \lfloor t/\Delta \rfloor$. Each pair of communicating UAV nodes establishes a session key at time index T, i.e., $SK_{ij}^{(T)} = \mathrm{H}(K_i||K_j||ID_i||ID_j||\mathrm{T}||r_g)$, as shown in Eq. (4), where K_i and K_j denote long-term keys shared with the GCS, and r_g is a fresh nonce generated by the GCS. The secure packet generation process is defined as:

$$\mathrm{C} = Enc_{SK_{ij}^{(T)}}(m) \tag{8}$$

$$\tau = Auth_{SK_{ij}^{(T)}}(C, A) \tag{9}$$

where $A = (ID_i, ID_j, T, ctr_i)$ is associated data. Thus, a transmitted packet can be represented as:

$$\mathrm{p} = (\mathrm{C}, \tau, \mathrm{T}, ctr_i) \tag{10}$$

This formulation ensures that each packet is uniquely bound to its sender, receiver, and time index, providing a foundation for secure communication.

5.2 Communication and Computation Overhead

The efficiency of the proposed TSCID framework is analyzed in terms of computational complexity and communication overhead. From the protocol design, each session establishment involves a constant number of symmetric encryption and hash operations. Specifically, the GCS performs one hash operation to generate the session key and two AEAD encryptions to distribute the key to UAV nodes. Each UAV performs at most two AEAD decryptions and one encryption during authentication. Therefore, the computational complexity per communication session can be expressed as $O_{comp} = O(1)$, indicating constant-time complexity independent of the network size.

For data transmission, each message requires one AEAD encryption at the sender and one AEAD decryption at the receiver, resulting in minimal computational overhead. In terms of communication overhead, each packet includes additional metadata associated with authenticated communication and replay protection:

$$|p| = |C| + |\tau| + |T| + |r_g| + |ctr_i| \tag{11}$$

where C denotes the encrypted payload, τ represents the authentication tag, T is the time index, r_g is the random nonce, and ctr_i represents the packet counter associated with replay protection.

The proposed framework introduces additional communication overhead associated with time indices, nonce values, authentication tags, and packet-verification metadata required for lightweight authenticated communication. The approximate overhead components introduced by the proposed framework are summarized in Table 2. However, the overhead remains relatively small because the framework relies primarily on lightweight symmetric-key operations and compact AEAD-based packet protection mechanisms without certificate exchange or expensive asymmetric-key payload structures. Table 2 summarizes the approximate additional packet overhead introduced by the proposed TSCID framework. The overall overhead mainly originates from the time index, random nonce, authentication tag, and replay-protection counter required for authenticated communication and replay resistance. A comparison of the approximate communication overhead introduced by different secure communication architectures is summarized in Table 3. Compared with PKI-based schemes that require certificate exchange and public-key operations, the proposed approach significantly reduces message size and transmission latency, making TSCID suitable for real-time UAV swarm environments with strict resource constraints.

Table 2: Additional packet overhead of the proposed TSCID framework.

Field	Description	Approx. Size
T	Time Index	4 bytes
r_g	Random Nonce	8–16 bytes
τ	AEAD Authentication Tag	16 bytes
ctr_i	Packet Counter	4 bytes
Total	**Additional Overhead**	**32–40 bytes**

Table 3: Packet overhead comparison.

Scheme	Additional Payload Components	Approx. Overhead
PKI-based	Certificate, Signature, Public Key	>200 bytes
ECC-based lightweight	ECC Public Parameters, Nonce, Auth Data	60–120 bytes
Proposed TSCID	Time Index T, Random Nonce r_g, AEAD Authentication Tag	24–48 bytes

Although the proposed framework achieves constant per-session cryptographic complexity from the perspective of individual UAV nodes, centralized GCS-assisted session establishment may introduce scalability bottlenecks under large-scale swarm deployments with frequent pairwise session requests. Therefore, the current framework is primarily intended for resource-constrained and moderate-scale UAV swarm environments. Distributed trust coordination, hierarchical GCS architectures, and decentralized session-management mechanisms will be investigated in future work to improve scalability under large-scale swarm conditions.

The current evaluation primarily focuses on lightweight deployment feasibility and communication efficiency rather than large-scale swarm scalability optimization.

5.3 Security Analysis

This section analyzes the security properties of the proposed TSCID framework under the defined adversary model and standard cryptographic assumptions commonly adopted in lightweight symmetric-key secure communication protocols. The analysis focuses on evaluating confidentiality, integrity, authentication, replay resistance, and session-key isolation characteristics of the proposed lightweight communication mechanism in resource-constrained UAV swarm environments.

Consistent with the threat model defined in Section 3.2, the adversary is assumed to have full control over the communication channel, including the capability to eavesdrop, intercept, replay, modify, and inject packets. However, the adversary is not assumed to physically compromise legitimate UAV nodes, the GCS, or their stored long-term secret keys during protocol operation. Therefore, the security guarantees of the proposed framework are established under the assumption that long-term secret keys stored in legitimate UAV nodes and the GCS remain protected throughout protocol execution.

In addition, the following standard cryptographic assumptions are adopted throughout the analysis: (i) the hash function H(·) behaves as a secure pseudorandom function (PRF); (ii) Ascon-AEAD provides confidentiality (IND-CPA) and ciphertext integrity (INT-CTXT); and (iii) all nonces are generated uniformly at random and are not reused under the same session key.

Assumption 1: *The hash function H(·) used in session-key derivation is modeled as a secure pseudorandom function. In addition, the GCS-generated nonce r_g is fresh for every session establishment.*

Theorem 1: *(Inter-Session Key Freshness and Independence) For any two sessions s and s′ established between UAV nodes U_i and U_j, if their corresponding protocol inputs differ in either the time index or the GCS-generated nonce,* i.e., $(T_s, r_g^{(s)}) \neq (T_{s'}, r_g^{(s')})$, *then the derived session keys $SK_{ij}^{(T)}$ and $SK_{ij}^{(T')}$ are computationally independent, provided that H(·) is modeled as a secure pseudorandom function.*

Proof of Theorem 1: According to the protocol, the session key generated by the GCS for session s is

$SK_{ij}^{(T_s)} = \mathrm{H}(K_i||K_j||ID_i||ID_j||\mathrm{T}_s||r_g^s)$, and for session s′, $SK_{ij}^{(T_{s'})} = \mathrm{H}(K_i||K_j||ID_i||ID_j||\mathrm{T}_{s'}||r_g^{s'})$.

Let $X_s = K_i||K_j||ID_i||ID_j||\mathrm{T}_s||r_g^s$, and $X_{s'} = K_i||K_j||ID_i||ID_j||\mathrm{T}_{s'}||r_g^{s'}$.

Because the protocol enforces freshness through the time index T and the nonce r_g, we have

$X_s \neq X_{s'}$ whenever $(T_s, r_g^{(s)}) \neq (T_{s'}, r_g^{(s')})$.

Under the assumption that H(·) behaves as a PRF, the outputs $\mathrm{H}(X_s)$ and $\mathrm{H}(X_{s'})$ on distinct inputs are computationally indistinguishable from two independent random strings of the same length. Therefore, the session keys $SK_{ij}^{(T)}$ and $SK_{ij}^{(T')}$ are computationally independent.

Hence, knowledge of one session key does not enable an adversary to infer any useful information about the other session key except with negligible probability. This proves session-key freshness and inter-session independence. □

Assumption 2: *Ascon-AEAD is IND-CPA secure, and the nonce used in AEAD encryption is never reused under the same session key. The confidentiality of the proposed protocol follows from the secrecy of the derived session key and the IND-CPA security of Ascon-AEAD.*

Theorem 2: *(Confidentiality of Payloads) Assume that Ascon-AEAD is IND-CPA secure, the long-term keys K_i and K_j are secret, and the nonces used in AEAD encryption are not reused under the same session key.*

Then the proposed TSCID protocol ensures confidentiality of transmitted payloads against any polynomial-time adversary.

Proof of Theorem 2: In the proposed protocol, the payload m transmitted between UAV nodes U_i and U_j is protected as

$$C_{data} = \text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{id}, A_{data}, m)$$

where $A_{data} = (ID_i, ID_j, T_x, ctr_i^{(d)})$, and the session key is defined as $SK_{ij}^{(T)} = \text{H}(K_i||K_j||ID_i||ID_j||\text{T}||r_g)$. Since K_i and K_j are long-term secret keys shared only with the GCS, and r_g is a fresh nonce generated by the GCS for each session establishment, an adversary that does not compromise these long-term secrets cannot compute $SK_{ij}^{(T)}$. By Theorem 1, session keys corresponding to different sessions are computationally independent. Now consider an adversary $\mathcal{A}$ observing ciphertexts exchanged over the public channel. Because the payload is encrypted using Ascon-AEAD under key $SK_{ij}^{(T)}$, and because the nonce N_{id} is assumed to be fresh under the same session key, the ciphertext enjoys IND-CPA confidentiality. Hence, for any two equal-length messages m_0 and m_1 chosen by $\mathcal{A}$, the ciphertexts $\text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{id}, A_{data}, m_0)$ and $\text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{id}, A_{data}, m_1)$ are computationally indistinguishable to $\mathcal{A}$. Therefore, the adversary cannot recover the plaintext m from the observed ciphertext C_{data} except with negligible probability. Equivalently, $\Pr[\mathcal{A}(C_{data}) \to m] \approx 0$. Hence, the proposed protocol preserves payload confidentiality. □

Assumption 3: *Ascon-AEAD is INT-CTXT secure, and the session key $SK_{ij}^{(T)}$ is known only to legitimate participants of the session.*

Theorem 3: *(Integrity and Packet-Origin Authenticity) Assume that Ascon-AEAD is INT-CTXT secure, the long-term keys K_i and K_j are secret, and the session key $SK_{ij}^{(T)}$ is known only to legitimate parties. Then the proposed TSCID protocol guarantees message integrity and source authenticity for all accepted packets.*

Proof of Theorem 3: In the proposed protocol, each transmitted payload m is protected as

$$C_{data} = \text{AEAD.Enc}_{SK_{ij}^{(T)}}(N_{id}, A_{data}, m)$$

where the associated data is defined as $A_{data} = (ID_i, ID_j, T_x, ctr_i^{(d)})$

The receiver accepts a packet only if the following conditions are simultaneously satisfied:

(i) the time index T_x falls within the valid time window,
(ii) the counter $ctr_i^{(d)}$ has not been previously used, and
(iii) the AEAD authentication tag is successfully verified.

To prove integrity, consider an adversary $\mathcal{A}$ that intercepts a valid ciphertext and attempts to modify either the ciphertext body or the associated data. Since Ascon-AEAD provides ciphertext integrity under the INT-CTXT notion, any forged ciphertext-tag pair that was not legitimately generated under $SK_{ij}^{(T)}$ will be rejected except with negligible probability. Therefore, if $\mathcal{A}$ modifies C_{data} or A_{data}, the resulting packet $\tilde{p}$ cannot pass verification except with negligible probability. Hence, $\Pr[\text{Verify}(\tilde{p}) = 1 \wedge \tilde{p} \neq p] \approx 0$.

Next, we prove source authenticity. In the proposed protocol, only legitimate nodes that possess the valid session key $SK_{ij}^{(T)}$ can generate a valid AEAD ciphertext and tag pair under that key. Since $SK_{ij}^{(T)}$ is established through GCS-assisted key distribution and protected by the long-term secrets K_i and K_j, an adversary without access to these keys cannot create a valid encrypted packet that will be accepted as originating from U_i. In addition, the sender identity ID_i is cryptographically bound in the associated data A_{data}. Thus, any attempt to impersonate a legitimate sender by altering the identity field or forging a new

packet will fail verification except with negligible probability. Therefore, every accepted packet must have been generated by an entity possessing the correct session key and using the correct sender identity. Hence, the proposed TSCID protocol guarantees both message integrity and source authenticity. □

Assumption 4: *The GCS is trusted, the long-term keys K_i and K_j are not compromised, and Ascon-AEAD prevents ciphertext forgery under unknown session keys.*

Theorem 4: *(Mutual Authentication). Assume that the long-term keys K_i and K_j are known only to the legitimate UAV nodes and the GCS, the GCS is trusted, and Ascon-AEAD is secure against ciphertext forgery. Then, upon successful completion of the authentication phase, the proposed TSCID protocol achieves mutual authentication between U_i and U_j.*

Proof of Theorem 4: The proof follows from the two-stage design of the protocol: secure session-key distribution and nonce-based challenge–response authentication.

First, during the session establishment phase, the GCS generates the session key $SK_{ij}^{(T)}$, and delivers it separately to U_i and U_j through AEAD-protected channels under the long-term keys K_i and K_j, respectively. Because only legitimate U_i and U_j share K_i and K_j with the GCS, only these legitimate nodes can decrypt the corresponding messages and recover the same session key.

Second, during the authentication phase, U_j generates a fresh nonce $\mathrm{r}_{j'}$ and sends $M_4 = (ID_j, ID_i, T_4, r'_j, ctr_j)$ encrypted under $SK_{ij}^{(T)}$. After decrypting this message, U_i must return a valid response containing the same nonce $\mathrm{r}_{j'}$ together with the confirmation value $ack_i = \mathrm{H}(r_i||r_j||SK_{ij}^{(T)}||T_5)$, and protect the response again using AEAD under $SK_{ij}^{(T)}$.

We now show authenticity in both directions.

(1) Authentication of U_j to U_i:

When U_i successfully decrypts C_4 and verifies that the recovered nonce $\mathrm{r}_{j'}$ is fresh and the ciphertext is valid under $SK_{ij}^{(T)}$, U_i concludes that the sender must know the correct session key. Since only legitimate U_j and U_i possess this session key after GCS-assisted establishment, and U_i itself did not generate C_4, the sender must be U_j. Therefore, U_i authenticates U_j.

(2) Authentication of U_i to U_j:

When U_j successfully decrypts C_5 and verifies that the returned nonce $\mathrm{r}_{j'}$ matches the one generated in Step 9, and that the confirmation value ack_i is correct, U_j concludes that the responder must know both the fresh challenge and the correct session key $SK_{ij}^{(T)}$. Because an adversary without $SK_{ij}^{(T)}$ cannot generate a valid AEAD-protected response, and because the response binds the fresh nonce and session-specific parameters, the responder must be U_i. Therefore, U_j authenticates U_i.

Hence, successful completion of the authentication phase implies that U_i and U_j both possess the same fresh session key and have verified each other's liveness and legitimacy. Therefore, the proposed TSCID protocol achieves mutual authentication. □

Lemma 1: *(Replay Resistance) The proposed TSCID protocol is resistant to replay attacks.*

Proof of Lemma 1: To prove replay resistance, we first define the packet acceptance rule used by the receiver. Let p denote an incoming protected packet. The receiver accepts p only if

$$\mathrm{Accept}(p) = I_{time}(p) \cdot I_{fresh}(p) \cdot I_{auth}(p) = 1 \tag{12}$$

where

- $I_{time}(p) = 1$ if the time index carried by p lies within the valid time window, and 0 otherwise;

- $I_{fresh}(p) = 1$ if the nonce or counter value carried by p has not been used previously in the corresponding session or time interval, and 0 otherwise;
- $I_{auth}(p) = 1$ if the authentication tag of p is successfully verified under the appropriate session key, and 0 otherwise.

Thus, a packet is accepted only when it simultaneously satisfies time validity, freshness, and authentication correctness.

Now consider an adversary $\mathcal{A}$ that records a previously valid protocol message p and later replays it in an attempt to make the receiver accept it again. We analyze two cases.

- Case 1: Replay outside the valid time window.

Suppose $\mathcal{A}$ replays p after the original communication interval has expired. Since the replayed packet still carries its original time index T, the receiver evaluates $I_{time}(p)$. Because T no longer lies within the allowable interval $[T_{now} - \delta, T_{now} + \delta]$, we have $I_{time}(p) = 0$. Hence, $\text{Accept}(p) = I_{time}(p) \cdot I_{fresh}(p) \cdot I_{auth}(p) = 0$, and the replayed packet is rejected.

- Case 2: Replay within the valid time window.

Suppose $\mathcal{A}$ replays p while its time index is still within the allowable interval. In this case, the time-validity test may still succeed, i.e., $I_{time}(p) = 1$. However, the receiver also checks freshness. For data packets, freshness is enforced through the monotonic counter $ctr_i^{(d)}$. For session-establishment and authentication messages, freshness is enforced through the corresponding nonces and counters. Since the replayed packet contains a nonce or counter value that has already been observed, the receiver sets $I_{fresh}(p) = 0$. Therefore, $\text{Accept}(p) = I_{time}(p) \cdot I_{fresh}(p) \cdot I_{auth}(p) = 0$, and the replayed packet is rejected.

In both cases, a replayed packet cannot satisfy the acceptance rule. Therefore, any adversarial attempt to reuse an earlier valid message is prevented either by failure of the time-validity condition or by failure of the freshness condition. Hence, the proposed TSCID protocol is resistant to replay attacks. □

Assumption 5: *The long-term keys K_i and K_j are not compromised, the GCS generates a fresh nonce r_g for each session establishment, and H(·) is modeled as a secure PRF.*

Theorem 5: *(Inter-Interval Session-Key Independence and Compromise Containment) Under the assumption that long-term keys K_i and K_j remain secure during protocol operation, assume that the long-term keys K_i and K_j remain secret, the GCS-generated nonce r_g is fresh for every session establishment, and the hash function H(·) behaves as a secure pseudorandom function (PRF). Then compromise of one session key $SK_{ij}^{(T)}$ does not reveal any useful information about past or future session keys $SK_{ij}^{(T')}$ for $T' \neq T$.*

Proof of Theorem 5: In the proposed TSCID protocol, the session key for communication between U_i and U_j at time index T is derived as

$$SK_{ij}^{(T)} = \mathrm{H}(K_i||K_j||ID_i||ID_j||\mathrm{T}||r_g).$$

Consider another session established at time index T', where $T' \neq T$, with fresh GCS nonce r'_g. The corresponding session key is

$$SK_{ij}^{(T')} = \mathrm{H}(K_i||K_j||ID_i||ID_j||\mathrm{T}'||r'_g).$$

Let

$$X_T = K_i||K_j||ID_i||ID_j||\mathrm{T}||r_g,$$
$$X_{T'} = K_i||K_j||ID_i||ID_j||\mathrm{T}'||r'_g.$$

Because either the time index differs, i.e., $T' \neq T$, or the GCS nonce is freshly generated for each session, we have

$X_T \neq X_{T'}$.

Under the PRF assumption on $\mathrm{H}(\cdot)$, the outputs $\mathrm{H}(X_T)$ and $\mathrm{H}(X_{T'})$ on distinct inputs are computationally indistinguishable from two independent random strings. Therefore, knowledge of $SK_{ij}^{(T)} = \mathrm{H}(X_T)$ does not allow an adversary to compute or distinguish $SK_{ij}^{(T')} = \mathrm{H}(X_{T'})$ except with negligible probability. Hence, compromise of the session key corresponding to one communication interval does not affect the secrecy of session keys used in earlier or later intervals. In other words,

$$SK_{ij}^{(T)} \perp SK_{ij}^{(T')}, \forall T' \neq T. \tag{13}$$

Therefore, the proposed protocol achieves session-key isolation across communication intervals. □

Remark 1: *(Role of Intrusion Detection) The IDS does not directly enhance the cryptographic guarantees of the proposed protocol. Instead, it provides an additional layer of defense by detecting anomalous traffic patterns and network-level attacks that cannot be prevented solely by encryption and authentication mechanisms. Formally, the IDS operates as* $y = f_{IDS}(\Psi(p))$, *where* $\Psi(p)$ *extracts behavioral features from authenticated packets. By analyzing only validated traffic, the IDS avoids processing malformed or adversarial inputs and improves detection reliability. Consequently, the overall framework follows a defense-in-depth strategy: cryptographic mechanisms ensure confidentiality, integrity, authentication, and replay protection, while IDS enhances resilience against behavioral and system-level threats.*

6 Intrusion Detection Modeling and Optimization

6.1 IDS Design Overview

While the proposed TSCID protocol provides strong cryptographic guarantees such as confidentiality, integrity, and authentication, it cannot prevent attacks that manifest at the behavioral level, such as traffic flooding, abnormal communication patterns, or coordinated DoS attacks. These types of attacks often generate structurally valid packets that successfully pass cryptographic verification but exhibit anomalous communication patterns over time. To address these limitations, an IDS is integrated as a complementary security mechanism.

The IDS operates on authenticated traffic generated by TSCID. Specifically, only packets that successfully satisfy the verification condition defined in Section 4, i.e., $\mathrm{Accept}(p) = 1$, are forwarded to the IDS module. This design ensures that the IDS processes only valid and structurally correct packets, thereby reducing noise, eliminating malformed inputs, and improving detection reliability.

From a system perspective, the integration of TSCID and IDS forms a two-stage processing pipeline. First, incoming packets are validated through cryptographic verification, including authentication, freshness, and integrity checks. Only after this verification stage are packets passed to the IDS for behavioral analysis. This separation ensures that cryptographic security and anomaly detection operate in a complementary and non-overlapping manner.

Formally, the IDS is defined as a classification function $y = f_{IDS}(x)$, where $x \in \mathbb{R}^d$ is a feature vector extracted from network traffic, and $y \in \{\text{normal, attack}\}$ is the classification output. The feature vector x is derived from validated packets p through a feature extraction function $\Psi(\cdot)$, i.e., $x = \Psi(p)$, which maps packet-level and temporal information into a structured representation suitable for learning-based detection.

The overall workflow of the proposed system is illustrated in Fig. 2. As shown in the figure, data generated by UAV node U_i is first encrypted using the TSCID protocol and transmitted through the network. Upon reception, packets undergo a verification process to evaluate whether Accept $(p) = 1$. Only validated packets are forwarded to the feature extraction module, followed by the IDS model for classification. The IDS outputs a decision indicating whether the traffic is normal or malicious, which can then trigger subsequent actions such as forwarding, filtering, or mitigation.

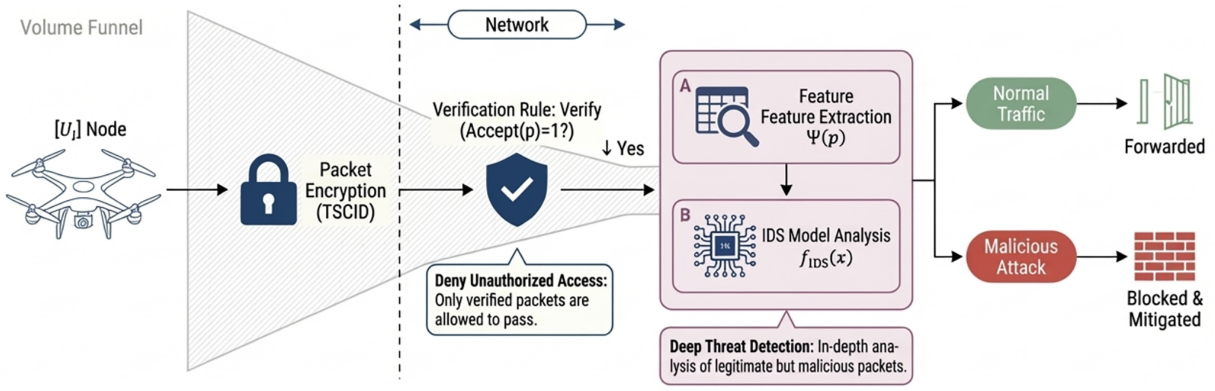

Figure 2: Integration of TSCID protocol and IDS module. Only authenticated packets that satisfy the acceptance rule are forwarded to the IDS for behavioral analysis.

This design ensures a clear separation between cryptographic verification and behavioral analysis, allowing the IDS to operate on trusted inputs only. As a result, the proposed framework achieves a defense-in-depth architecture that combines lightweight cryptographic protection with intelligent anomaly detection.

Different from conventional standalone IDS architectures that analyze raw network traffic directly, the proposed framework performs anomaly detection only after cryptographic verification and packet authentication. This coordinated interaction between secure communication and IDS processing reduces malformed traffic propagation into the detection pipeline and improves detection reliability under resource-constrained UAV environments.

The proposed IDS framework is primarily designed for lightweight deployment on resource-constrained UAV-edge platforms and may also be coordinated through the GCS depending on computational-resource availability and mission requirements. In practical UAV swarm environments, anomaly detection can therefore be performed either locally on UAV nodes for real-time monitoring or partially coordinated through the GCS for additional swarm-level analysis and threat management.

Since IDS processing is performed only after packet authentication and replay verification, malformed or unauthenticated packets can be filtered before entering the IDS analysis pipeline, thereby reducing unnecessary processing overhead and improving resource-utilization efficiency under constrained UAV-edge environments.

6.2 Feature Extraction Model

The effectiveness of the IDS critically depends on the quality of the extracted features. In the proposed framework, feature extraction is designed to capture both packet-level characteristics and temporal communication behaviors, enabling the detection of both instantaneous anomalies and long-term attack patterns.

The current IDS evaluation primarily focuses on validating the lightweight IDS architecture and structured-pruning effectiveness under representative authenticated-traffic assumptions using the

CIC-IDS2017 dataset. Although the proposed TSCID framework conceptually integrates authenticated communication and behavioral anomaly detection, the public dataset does not contain traffic generated directly from a fully deployed TSCID protocol stack. Therefore, the reported IDS performance mainly reflects the effectiveness of the lightweight IDS component operating after packet-authentication assumptions rather than a fully emulated end-to-end UAV swarm communication pipeline.

(1) Packet-Level Features

i. Packet-level features describe the intrinsic properties of individual packets. These include:
ii. Packet size ($|p|$), reflecting transmission payload characteristics
iii. Header-related information, such as sender ID (ID_i) and receiver ID (ID_j)
iv. Time index T, inherited from the TSCID protocol

These features allow the IDS to identify abnormal packet structures or irregular communication endpoints.

(2) Temporal Features

To capture dynamic communication behaviors, temporal features are extracted based on packet arrival patterns. Let p_t denote the packet observed at time step t_i. The temporal features include:

- Inter-arrival time: $\Delta_t = t_i - t_{i-1}$
- Packet rate within a window: $R_t = N_w/w$

where N_w denotes the number of packets within a sliding window of size w.
These features are particularly effective in detecting flooding attacks and abnormal traffic bursts.

(3) Statistical Features

To model longer-term behavior, statistical features are computed over a sliding window:

$$\mu_t = 1/w \sum_{k=t-w+1}^{t} p_k \tag{14}$$

$$\sigma_{t^2} = 1/w \sum_{k=t-w+1}^{t} (p_k - \mu_t)^2 \tag{15}$$

where μ_t and σ_{t^2} represent the mean and variance of packet sizes within the window, respectively. These features enable the detection of subtle anomalies that may not be evident from individual packets.

(4) Time-Indexed Feature Aggregation

Unlike conventional IDS designs, the proposed framework leverages the time index T introduced by TSCID to organize feature extraction. Specifically, features are aggregated within each time interval T:

$$x_T = \Psi(\{p_k | k \in \text{ interal } T\}) \tag{16}$$

This time-indexed aggregation provides a natural segmentation of network behavior, aligning feature extraction with the cryptographic session boundaries.

(5) Final Feature Representation

To construct a comprehensive representation of network behavior, the extracted features are grouped into three categories:

i. Packet-level feature vector: $x_{pkt} \in \mathbb{R}^{d_1}$, consisting of features derived from individual packets, such as packet size, sender/receiver identities, and time index.
ii. Temporal feature vector: $x_{temp} \in \mathbb{R}^{d_1}$, capturing dynamic traffic patterns, including inter-arrival time and packet rate within a sliding window.
iii. Statistical feature vector: $x_{stat} \in \mathbb{R}^{d_1}$, representing aggregated statistics such as mean and variance of packet sizes over a time window.

The final feature vector is constructed by concatenating these components:

$$x = [x_{pkt}, x_{temp}, x_{stat}] \in \mathbb{R}^d \tag{17}$$

where $d = d_1 + d_2 + d_3$.

6.3 Deep Learning–Based Detection Model

In previous sections, the IDS is abstractly modeled as a function $y = f_{IDS}(x)$. In this section, we instantiate $f_{IDS}(\cdot)$ using a parameterized deep neural network, denoted as $y = f_{IDS}(x;\theta)$, where $x \in \mathbb{R}^d$ is the input feature vector, θ represents the learnable model parameters, and $y \in \{0,1\}$ represents the classification result, with 0 indicating normal traffic and 1 indicating malicious activity.

(1) Model Architecture

The function $f_{IDS}(\cdot)$ is realized using a DNN composed of L layers. The forward propagation is defined as:

$$z^{l+1} = \sigma\left(W^l z^l + b^l\right), \text{ for } l = 1, \ldots, L \tag{18}$$

where $z^l = x$ is the input feature vector, W^l and b^l are the weight matrix and bias vector of layer l, and $\sigma(\cdot)$ is a nonlinear activation function. The final output layer produces a probability score

$$\hat{y} = \sigma_{out}(W^l z^l + b^l) \tag{19}$$

where $\hat{y} \in [0,1]$ represents the likelihood that the input corresponds to an attack.

(2) Design Rationale

The use of a DNN is motivated by the heterogeneous nature of the feature space. Specifically, x_{pkt} captures structural packet information, x_{temp} captures dynamic temporal behavior, and x_{stat} captures long-term statistical patterns. These feature components exhibit nonlinear relationships that are difficult to model using traditional rule-based or shallow learning methods. The multi-layer structure of the DNN enables hierarchical feature abstraction, allowing the model to learn complex correlations across different feature types.

(3) Training Objective

The model is trained using labeled data to minimize a binary cross-entropy loss:

$$L(\theta) = -\sum_{i=1}^{N} y_i \log \widehat{y_i} + (1 - y_i) \log(1 - \widehat{y_i}) \tag{20}$$

where N is the number of training samples, y_i is the ground truth label, and $\widehat{y_i}$ is the predicted probability. The parameters θ are optimized using gradient-based methods such as stochastic gradient descent (SGD) or Adam.

(4) Decision Rule

During inference, the classification result is obtained using a threshold τ:

$$y = \begin{cases} 1, & \text{if } \hat{y} \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where $\tau \in (0,1)$ is typically set to 0.5 or tuned based on the desired trade-off between false positives and false negatives.

(5) Integration with TSCID Features

The input feature vector x is constructed from validated packets, i.e., $x = \Psi(p)$, where $\text{Accept}(p) = 1$. This ensures that the DNN operates on trusted inputs only. Furthermore, the inclusion of the time index T in x enables the model to capture time-correlated attack patterns aligned with the TSCID session structure.

To clearly describe the operational flow of the proposed IDS, the detection procedure is summarized in Algorithm 1. The algorithm takes validated packets generated by the TSCID protocol as input, extracts multi-level traffic features, and performs attack classification using the trained DNN model.

Algorithm 1: This is an algorithm caption Online IDS Inference Procedure

Input: validated packet stream P, trained model $f_{IDS}(x;\theta)$, window size w, threshold τ
Output: detection labels y

Initialize buffer B $\leftarrow \varnothing$
for each packet p_t in P **do**
  if $\text{Accept}(p_t) = 0$ **then**
    Blocked
  end if
  Update sliding window buffer B with p_t
  $x_{pkt} \leftarrow \Psi_{pkt}(p_t)$
  $x_{temp} \leftarrow \Psi_{temp}(B)$
  $x_{stat} \leftarrow \Psi_{stat}(B)$
  $x \leftarrow [x_{pkt}, x_{temp}, x_{stat}]$
  $\hat{y} \leftarrow f_{IDS}(x;\theta)$
  $\hat{y} \leftarrow 1$ if $\hat{y} \geq \tau$ else 0
  Output y
  if $y = 1$ **then**
    Trigger mitigation
  end if
end for

6.4 Model Optimization via Pruning

To further improve deployment efficiency on resource-constrained UAV platforms, the proposed IDS model is optimized using a deployment-oriented structured pruning strategy. The pruning process incorporates both model-parameter characteristics and authenticated traffic-feature behavior associated with the TSCID communication framework to reduce IDS computational complexity while maintaining practical anomaly-detection performance under UAV-edge deployment constraints.

As described in Section 6.1, the IDS operates exclusively on authenticated packets, which exhibit structured and predictable patterns due to time-indexed communication and protocol constraints. This property allows the pruning process to incorporate both model-parameter importance and authenticated traffic-feature characteristics during IDS optimization.

Let the IDS input be $x = [x_{pkt}, x_{temp}, x_{stat}]$. Instead of treating all features equally, the proposed method evaluates the contribution of each feature group to the model output. Specifically, a feature-aware importance score is defined as:

$$S_i = \|w_i\|_2 \cdot \mathbb{E}[|x_i|] \tag{22}$$

where w_i represents the weights associated with feature x_i, and $\mathbb{E}[|x_i|]$ captures the average activation magnitude over the dataset. This formulation jointly considers model sensitivity and traffic-feature activation characteristics to support lightweight IDS optimization under authenticated UAV communication environments.

Furthermore, due to the time-indexed nature of TSCID, temporal features x_{temp} play a key role in detecting replay and DoS attacks, while statistical features x_{stat} capture long-term behavioral anomalies. Therefore, a group-wise pruning constraint is introduced to prevent over-pruning of critical feature categories:

$$\left\|\theta'_{temp}\right\|_0 \geq \eta_1, \|\theta'_{stat}\|_0 \geq \eta_2 \tag{23}$$

where η_1 and η_2 are minimum retention thresholds. This ensures that essential temporal and statistical information is preserved.

From a structural perspective, pruning is applied at the neuron level in fully connected layers, removing entire neurons associated with low-importance scores. This results in a compact model that is both memory-efficient and hardware-friendly. The overall optimization procedure is described as Algorithm 2.

Algorithm 2: Traffic-aware structured pruning for IDS

Input: Model $f_{IDS}(x;\theta)$, dataset $\mathcal{D}$, pruning ratio ρ
Output: Optimized model $f_{IDS}(x;\theta')$
Initialize $\theta' \leftarrow \theta$
for each neuron/filter i **do**
    Compute importance score $S_i = \|w_i\|_2 \cdot \mathbb{E}[|x_i|]$
end for
Partition parameters into feature groups (pkt, temp, stat)
Within each group, rank units based on S_i
Remove lowest ρ\rho$\rho$ fraction of units while satisfying $\left\|\theta'_{temp}\right\|_0 \geq \eta_1$ and $\|\theta'_{stat}\|_0 \geq \eta_2$
Update model structure
Fine-tune model using dataset $\mathcal{D}$
Repeat until convergence
Return θ'

6.5 Integration with TSCID Framework

The optimized IDS module is tightly integrated with the proposed TSCID framework to form a unified dual-layer security architecture. As described in previous sections, TSCID provides cryptographic protection through time-indexed key management and authenticated encryption, ensuring that all transmitted packets satisfy confidentiality, integrity, authentication, and replay protection. Building upon this secure communication foundation, the IDS operates as a second-layer defense that analyzes traffic behavior and detects attacks that cannot be mitigated at the cryptographic level.

In the integrated framework, the IDS is deployed after the packet verification stage. Specifically, only packets that satisfy the condition $Accept(p) = 1$ are forwarded to the IDS for further analysis, ensuring that the model processes only valid and structurally correct inputs. This design not only reduces computational overhead but also improves detection reliability by eliminating malformed or adversarial traffic prior to inference.

The interaction between TSCID and IDS can be viewed as a sequential processing pipeline. Let p denote an incoming packet. The overall system function can be expressed as

$$y = f_{IDS}(\Psi(p)) \text{ if } Accept(p) = 1, \tag{24}$$

and the packet is rejected otherwise. Here, $\Psi(p)$ represents the feature extraction function defined in Section 6.2, and $y \in \{\text{normal, attack}\}$ denotes the detection result. This formulation highlights the complementary roles of the two components: TSCID enforces cryptographic correctness, while IDS captures behavioral anomalies.

Furthermore, the lightweight design of the IDS, enabled by traffic-aware structured pruning, ensures that the integration does not introduce significant latency or energy overhead. The reduced model complexity allows real-time inference on UAV edge devices, maintaining system responsiveness even under dynamic network conditions. This is particularly important for UAV swarm environments, where timely detection of attacks such as denial-of-service and spoofing is critical for mission reliability.

The integration of the optimized IDS with TSCID establishes a defense-in-depth mechanism that combines cryptographic security and intelligent anomaly detection. This unified framework enhances robustness against both protocol-level and behavioral attacks while preserving the efficiency required for resource-constrained UAV systems.

6.6 Computational Efficiency Analysis

To evaluate the suitability of the proposed TSCID framework with the optimized IDS for resource-constrained UAV platforms, we analyze its computational complexity and system-level efficiency. The overall system consists of two main components: (i) lightweight secure communication based on symmetric cryptographic primitives, and (ii) IDS-based anomaly detection with pruning-based optimization.

For the communication layer, TSCID relies only on hash functions and authenticated encryption operations, both of which have constant-time complexity with respect to the message size. For each packet transmission, the sender and receiver perform a fixed number of hash and AEAD operations, resulting in an overall computational complexity of $O(1)$ per message. This is significantly more efficient than conventional public-key–based approaches, which typically involve expensive modular exponentiation or elliptic curve operations.

For the IDS component, let P denote the number of model parameters and C denote the number of arithmetic operations required for inference. The original model has complexity $O(P)$ and $O(C)$. After applying the proposed traffic-aware structured pruning with pruning ratio ρ, the complexity is reduced to $O((1-\rho)P)$ and $O((1-\rho)C)$, which directly leads to proportional reductions in memory usage and inference latency. This reduction is particularly important for UAV platforms, where both computational capacity and energy budget are limited.

From a system-level perspective, the integration of TSCID and IDS forms a sequential pipeline in which only authenticated packets are processed by the IDS. Let $\alpha \in [0,1]$ denote the fraction of packets that pass cryptographic verification. Then, the effective IDS workload is reduced to $\alpha \cdot C$, yielding an overall complexity of $O(1) - O(\alpha(1-\rho)C)$, where the first term corresponds to communication overhead and the second term corresponds to IDS inference. Since $\alpha < 1$ in adversarial environments, this design further reduces unnecessary computation by filtering invalid packets before IDS processing.

In addition, the structured pruning strategy ensures that the model maintains regular architecture after compression, enabling efficient execution on embedded processors without incurring irregular memory

access overhead. This is advantageous compared with unstructured pruning methods, which often require sparse computation support.

The proposed framework achieves a favorable balance between security and efficiency. By combining lightweight cryptographic operations with a compressed IDS model, the system supports real-time secure communication and anomaly detection on resource-constrained UAV devices, while maintaining low latency and energy consumption.

7 Experimental Evaluation

In this section, we evaluate the performance of the proposed TSCID framework with the optimized IDS model in terms of detection accuracy, computational efficiency, and system-level effectiveness. The experiments are designed to validate the intrusion detection capability, effectiveness of the proposed pruning strategy, and overall system efficiency under resource-constrained environments.

7.1 Experimental Setup

To evaluate the performance of the proposed TSCID framework with the optimized intrusion detection system, experiments are conducted under a resource-constrained environment that emulates practical UAV deployment conditions. The evaluation focuses on detection accuracy, computational efficiency, and the effectiveness of the proposed pruning strategy.

All experiments were conducted on Raspberry Pi 4B platforms equipped with a quad-core ARM Cortex-A72 processor operating at 1.5 GHz and 4 GB RAM running Raspberry Pi OS. The proposed framework was implemented in C using GCC 10.2.1. Communication latency and cryptographic execution time were measured using high-resolution system timers under repeated experimental trials to reduce measurement variance. This setup is selected to reflect the limited computational capability and energy constraints typical of UAV edge devices. Communication latency and cryptographic execution time are measured using high-resolution system timers under repeated experimental trials to reduce measurement variance and improve measurement reliability.

For intrusion detection evaluation, the widely used CIC-IDS2017 dataset is adopted as a representative network intrusion benchmark to evaluate the lightweight IDS architecture under authenticated-traffic assumptions. Although CIC-IDS2017 does not explicitly model UAV-specific swarm communication dynamics, selected attack categories, including DoS, spoofing-related anomalies, and MITM-like traffic manipulation behaviors, are used to approximate representative communication-layer attack patterns relevant to UAV swarm environments. In particular, spoofing-related and MITM-like behaviors are approximated using abnormal packet-manipulation and flow-interruption patterns extracted from CIC-IDS2017 traffic flows. These mappings are intended to provide representative communication-layer characteristics for lightweight IDS evaluation rather than exact UAV-specific attack implementations.

The dataset preprocessing pipeline includes feature normalization, attack-label mapping, train/test partitioning, and extraction of feature vectors as defined in Section 6.2, including packet-level features x_{pkt}, temporal features x_{temp}, and statistical features x_{stat} associated with authenticated traffic behavior. The data is normalized and divided into training and testing sets using a standard ratio.

The lightweight baselines mainly represent symmetric-key authenticated communication architectures employing lightweight hash-based authentication and low-overhead packet protection mechanisms commonly adopted in IoT and UAV-edge environments. The selected baselines are intended to provide comparison from both security-overhead and lightweight deployment perspectives. In particular, the PKI-based scheme represents conventional asymmetric-key coordination architectures, while the lightweight

baselines represent low-overhead authenticated communication approaches emphasizing computational efficiency on edge-class devices.

The key update interval Δ is evaluated within the range of 1–5 s. A shorter interval restricts the adversarial observation window for any derived session key, thereby enhancing security containment. However, it implicitly demands tighter baseline synchronization stability. Based on our empirical trials on the Raspberry Pi 4B platform, a 1–5 s window maintains a resilient trade-off that does not cause artificial packet rejections under moderate network jitter.

The IDS model is implemented as a lightweight deep neural network with multiple fully connected layers and ReLU activation functions. The model is first trained using the full parameter set, and then optimized using the structured pruning strategy optimized for traffic-feature-aware IDS deployment. Different pruning ratios are evaluated to analyze the trade-off between model size and detection performance. The experimental configuration is summarized in Table 4.

Table 4: Experimental and model configuration.

Category	Parameter/Dataset	Value/Description
Simulation NS-3	Swarm size	10, 20, 50, 100 UAVs
	Communication model	IEEE 802.11s mesh, 2.4 GHz band
	Propagation model	Log-distance path loss, mobility with random waypoint
Prototype Hardware	UAV platform emulation	Raspberry Pi 4B, Quad-core ARM Cortex-A72, 4 GB RAM
	Operating system	Raspbian OS, Linux kernel 6.x
Cryptography	TSCID cipher	Ascon-128 (AEAD), Ascon-Hash (PRF)
	Key update interval (Δ)	1–5 s
IDS	Dataset	CIC-IDS2017 (DoS, MITM, spoofing, port scan flows)
	Model architecture	DNN with 3 hidden layers, ReLU activation
	Optimization	Structured pruning (50%–60% parameter reduction)

7.2 Intrusion Detection Performance

This section evaluates the detection capability of the proposed IDS model under different attack scenarios. The experiments are conducted using the CIC-IDS2017 dataset, focusing on representative attack types relevant to UAV swarm environments, including DoS, spoofing, and MITM attacks.

The performance of the IDS is measured using standard classification metrics, including accuracy, precision, recall, F1-score, and false positive rate (FPR). Table 5 summarizes the detection results for each attack category.

Table 5: IDS detection performance under different attack types.

Attack Type	Accuracy	Precision	Recall	F1-Score	FPR
DoS	93.5%	92.8%	94.1%	93.4%	3.2%
Spoofing	94.2%	93.6%	94.8%	94.2%	2.9%
MITM	92.1%	91.5%	92.7%	92.1%	3.7%
Average	93.3%	92.6%	93.9%	93.2%	3.3%

As listed in Table 5, the proposed method achieves consistently high detection performance across all attack types. In particular, spoofing attacks achieve the highest accuracy of 94.2%, while DoS and MITM attacks are also detected with accuracy above 92%. The average detection accuracy reaches 93.3%, demonstrating the robustness of the model in distinguishing between normal and malicious traffic patterns. The false positive rate remains below 4% for all attack types, indicating that the IDS is able to effectively minimize incorrect classification of legitimate traffic. This is particularly important in UAV swarm environments, where excessive false alarms may disrupt mission execution and increase system overhead. The strong detection performance can be attributed to the feature representation defined in Section 6.2, which captures packet-level, temporal, and statistical characteristics of network traffic. In particular, temporal features contribute to identifying replay and flooding behaviors, while statistical features help detect coordinated or persistent attack patterns. Furthermore, the IDS operates on authenticated packets provided by the TSCID framework, ensuring that only valid and structurally correct data are analyzed. This design reduces noise in the input data and improves classification reliability compared with standalone IDS approaches.

7.3 Impact of Traffic-Aware Pruning

This section evaluates the effectiveness of the proposed pruning strategy in reducing model complexity while maintaining detection performance. The pruning mechanism is applied to the trained IDS model with different pruning ratios, and its impact on accuracy, model size, and inference latency is analyzed. Table 6 summarizes the performance of the IDS under different pruning ratios. The baseline model corresponds to the unpruned network.

Table 6: Impact of traffic-aware pruning on IDS performance.

Pruning Ratio (%)	Accuracy	F1-Score	Model Size	Latency
0%(Baseline)	93.3%	93.2%	12.4 MB	18.6 ms
20%	93.1%	93.0%	10.1 MB	16.2 ms
40%	92.8%	92.7%	7.6 MB	13.9 ms
60%	92.3%	92.1%	5.2 MB	10.8 ms

As shown in Table 6, the proposed pruning strategy significantly reduces model size and inference latency while maintaining high detection performance. When the pruning ratio reaches 60%, the model size is reduced by approximately 58%, and inference latency decreases from 18.6 to 10.8 ms. Despite this substantial reduction, the accuracy only decreases by 1.0%, demonstrating the robustness of the proposed approach.

The results indicate that the traffic-aware pruning strategy effectively removes redundant neurons while preserving critical features for intrusion detection. In particular, the incorporation of feature-aware importance scores and group-wise constraints ensures that temporal and statistical features, which are essential for detecting replay and coordinated attacks, are retained during the pruning process. Furthermore, the performance degradation remains gradual as the pruning ratio increases, suggesting that the IDS model contains significant redundancy that can be exploited for optimization. Compared with conventional pruning methods, the proposed approach achieves a better trade-off between accuracy and efficiency by leveraging traffic characteristics specific to UAV communication.

Fig. 3 further illustrates the relationship between pruning ratio and model performance. It can be observed that accuracy remains stable up to a pruning ratio of 40%, after which a slight degradation occurs. This suggests that moderate pruning can be applied without compromising detection reliability, making the

model suitable for deployment on resource-constrained UAV platforms. The results show that the detection accuracy remains stable under moderate pruning, while inference latency is significantly reduced. When the pruning ratio reaches 60%, the model achieves substantial efficiency improvement with only marginal degradation in accuracy. The experimental results confirm that the proposed pruning method effectively reduces computational overhead while preserving detection capability. This optimization plays a crucial role in enabling IDS deployment within the TSCID framework.

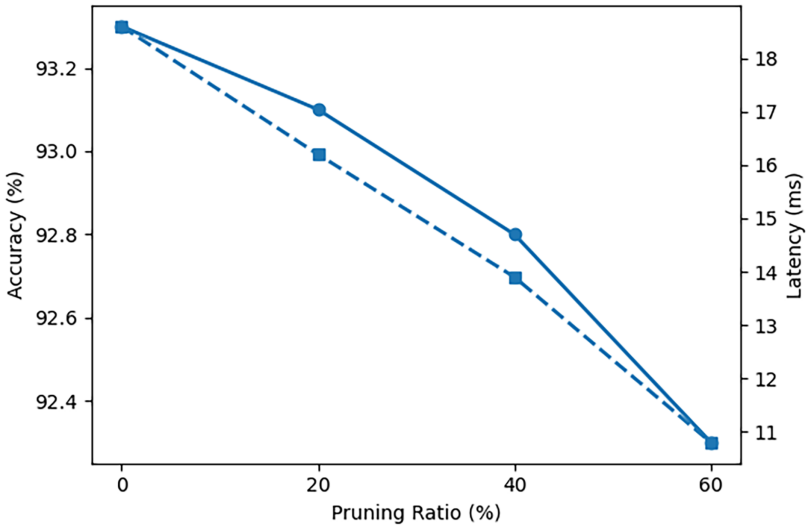

Figure 3: Impact of pruning on IDS performance and inference latency.

7.4 Computational Efficiency and TSCID Performance

This section evaluates the computational efficiency of the proposed framework by jointly analyzing the TSCID communication protocol and the optimized IDS model. The objective is to demonstrate that the integrated system achieves real-time performance with low computational overhead, making it suitable for resource-constrained UAV platforms.

We first analyze the performance of the TSCID protocol in comparison with a conventional PKI-based scheme. The evaluation focuses on communication latency and energy consumption per packet, which are critical metrics for UAV swarm deployments. As shown in Fig. 4, the proposed TSCID protocol consistently achieves lower communication latency across different swarm sizes. For example, when the swarm size is 10 UAVs, the average latency is reduced from 35 ms (PKI) to 22 ms (TSCID), corresponding to a 37% improvement. At a larger scale of 100 UAVs, latency decreases from 130 to 84 ms, achieving a 35% reduction. These improvements are mainly attributed to the lightweight time-indexed key derivation mechanism, which eliminates the need for computationally expensive public-key operations.

The energy efficiency comparison is presented in Fig. 5. The proposed TSCID protocol reduces per-packet energy consumption by approximately 25%–30% across all swarm sizes. For instance, at a swarm size of 50 UAVs, energy consumption is reduced from 2.8 J (PKI) to 2.0 J (TSCID). This improvement results from avoiding frequent key exchange procedures and relying on hash-based lightweight operations.

At the computational level, the TSCID protocol relies on lightweight symmetric cryptographic primitives, including hash functions and authenticated encryption AEAD. For each packet transmission, a constant number of operations is performed, resulting in an overall computational complexity of $O(1)$ per

message. In contrast, PKI-based approaches typically involve computationally expensive operations such as modular exponentiation, leading to higher latency and energy consumption.

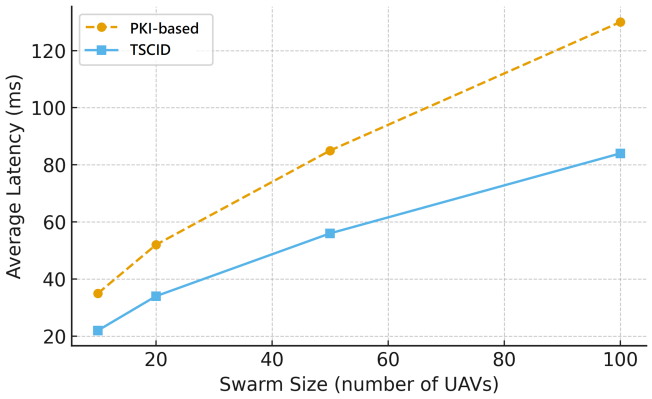

Figure 4: Average latency comparison between TSCID and a PKI-based scheme under different swarm sizes.

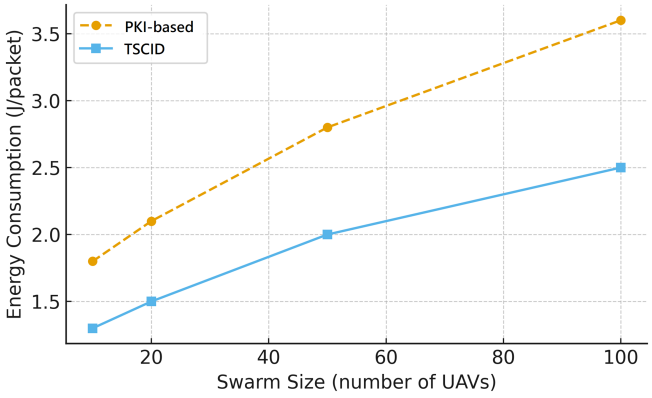

Figure 5: Energy consumption per UAV packet for TSCID and a PKI-based scheme.

For the IDS component, we evaluate the impact of the proposed traffic-aware structured pruning on computational efficiency. As shown in Table 7, the pruned model reduces parameter count by approximately 58%, resulting in a reduction in memory usage from 12.4 to 5.2 MB and a decrease in inference latency from 18.6 to 10.8 ms. These results demonstrate that pruning effectively reduces computational cost while maintaining detection performance.

Table 7: IDS efficiency before and after pruning.

Model Version	Parameters	Memory	Latency
Original IDS	100%	12.4 MB	18.6 ms
Pruned IDS	42%	5.2 MB	10.8 ms

Fig. 6 illustrates the effective IDS workload as a function of the authenticated packet ratio under different pruning configurations. It can be observed that the workload decreases proportionally with the reduction of authenticated traffic, indicating that the TSCID-based packet verification effectively filters out invalid inputs before IDS processing. Moreover, higher pruning ratios further reduce the workload, demonstrating the

complementary effect between traffic-aware pruning and protocol-level filtering. The slight variations across measurements reflect runtime fluctuations on resource-constrained hardware platforms, while the overall trend remains consistent.

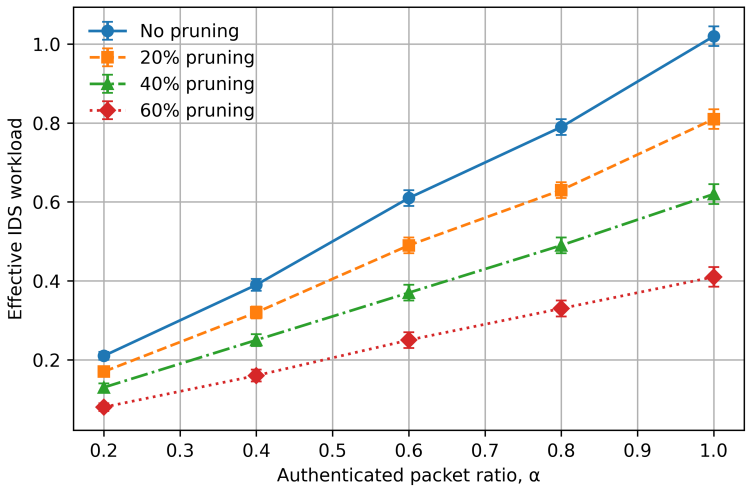

Figure 6: Effective IDS workload after TSCID-based packet verification.

Fig. 7 presents the normalized end-to-end computational cost under varying invalid packet ratios. As the proportion of invalid traffic increases, the overall system cost decreases due to early rejection of unauthenticated packets by the TSCID framework. In addition, higher pruning ratios consistently lead to lower computational cost across all scenarios. This demonstrates that the combination of protocol-level filtering and model-level optimization enables superior efficiency, particularly under adversarial traffic conditions.

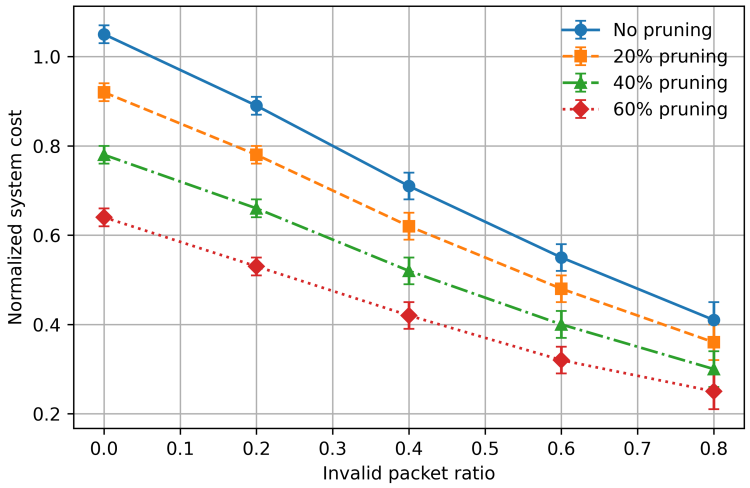

Figure 7: The normalized end-to-end computational cost under different invalid packet ratios.

The proposed framework achieves a favorable balance between security and efficiency. By combining lightweight TSCID communication with a pruned IDS model, the system enables real-time secure communication and anomaly detection while maintaining low latency and energy consumption. These results validate the analytical observations in Section 6.6 and confirm the practical deployability of the proposed TSCID-IDS framework.

7.5 System-Level Evaluation

To evaluate the effectiveness of the proposed framework at the system level, we compare three configurations. The first one is IDS-only, where intrusion detection is applied directly to incoming traffic without cryptographic filtering. The second is TSCID-only, where secure communication is enforced without behavioral analysis, and the third is the proposed TSCID + IDS framework, which integrates cryptographic verification with anomaly detection. Table 8 summarizes the performance comparison in terms of detection accuracy, false positive rate (FPR), and inference latency.

Table 8: System-level performance comparison.

Configuration	Accuracy	FPR	Latency
IDS only	91.8%	5.6%	18.6 ms
TSCID only	–	–	12.3 ms
TSCID + IDS	93.3%	3.3%	13.7 ms

As shown in Table 8, the IDS-only configuration achieves reasonable detection performance but suffers from a higher false positive rate, as it processes all incoming traffic, including malformed or adversarial packets. In contrast, the TSCID-only configuration provides secure communication but lacks the capability to detect behavioral anomalies.

The proposed integrated framework achieves the best overall performance, improving detection accuracy to 93.3% while reducing the false positive rate to 3.3%. This improvement is primarily attributed to the filtering effect of the TSCID protocol, which ensures that only authenticated and structurally valid packets are analyzed by the IDS. In terms of efficiency, the additional latency introduced by IDS processing remains minimal. The observed improvement mainly reflects the reduction of malformed and structurally inconsistent packets before feature extraction, rather than a modification of the IDS classifier itself. Compared with the TSCID-only configuration, the latency increases from 12.3 to 13.7 ms, which is acceptable for real-time UAV applications. This demonstrates that the integration of IDS does not significantly compromise system responsiveness.

Furthermore, the results confirm the effectiveness of the proposed dual-layer security architecture. The TSCID protocol mitigates cryptographic-level threats such as replay and spoofing at the communication layer, while the IDS captures higher-level behavioral anomalies, including traffic flooding and coordinated attacks. This complementary design enables comprehensive protection against a wide range of attack vectors. The system-level evaluation demonstrates that the proposed TSCID-IDS framework achieves a favorable balance between security, detection capability, and computational efficiency. The integration of lightweight cryptographic mechanisms with a pruned IDS model enables practical deployment in resource-constrained UAV swarm environments, fulfilling the design objectives of the proposed approach.

7.6 Discussion

While the experimental results validate the computational efficiency and detection latency of the proposed TSCID framework on the Raspberry Pi 4B platform, several practical deployment insights, architectural trade-offs, and methodology limitations merit detailed discussion.

(1) Security Containment vs. Synchronization Overheads

The empirical verification of the key update interval Δ within the 1–5 s range demonstrates a resilient balance between cryptographic strength and systemic stability. A shorter synchronization interval

fundamentally restricts the adversarial observation window for any intercepted ciphertext, thereby achieving strong forward-security containment. However, this architectural benefit implicitly presumes consistent baseline synchronization among UAV nodes. Under simulated moderate network jitter, the 1–5 s window prevents artificial packet rejections. Nevertheless, in extreme GPS-denied environments or tactical scenarios prone to severe packet collisions, the degradation of time synchronization would inevitably expand the validation variance Δ, potentially resulting in false authentication drops. Consequently, adapting Δ dynamically based on real-time network quality of service (QoS) remains a critical operational necessity.

(2) Cross-Layer Complementarity and Boundary Conditions

The primary structural advantage of TSCID is its cross-layer integration, which links deterministic cryptographic verification at the network layer with behavioral anomaly detection via the pruned DNN at the application layer. It is crucial to delineate the formal boundaries of this defense-in-depth model. As articulated in our threat model, the framework is optimized to mitigate over-the-air protocol violations, network-level spoofing, and resource-exhaustion DoS injection. However, if a UAV node undergoes a catastrophic physical capture where its cryptographic material is statically extracted via hardware-level side-channel profiling, the network layer's cryptographic identity is compromised. In such boundary conditions, the security perimeter shifts entirely to the DNN-based IDS, which must isolate the compromised node based purely on anomalous behavioral trajectories or routing patterns, rather than cryptographic validation.

(3) Dataset Constraints and Evaluation Validity:

The evaluation of the IDS component utilizes the benchmark CIC-IDS2017 dataset, which provides a comprehensive, standardized suite of multi-vector attack profiles essential for evaluating baseline classification capabilities. However, we explicitly acknowledge that standard enterprise or static network datasets do not fully reflect the specialized telemetry protocols, high-mobility routing dynamics, and distinct signal-to-noise ratios characteristic of dedicated UAV swarm networks. While the feature mapping and normalization pipeline described in Section 7.2 mitigates part of this domain discrepancy, validating the pruned model against real-world UAV physical testbeds under operational flight dynamics constitutes an indispensable phase for practical airworthiness verification.

(4) Swarm Heterogeneity and Scalability:

The current evaluation assumes a homogenous UAV edge environment to establish a clean comparative baseline against conventional PKI architectures. In complex industrial or search-and-rescue deployments, swarms are frequently heterogeneous, comprising platforms with highly asymmetric computational capabilities, varying battery capacities, and disparate onboard sensors. Introducing resource asymmetry impacts the spatial distribution of the collaborative IDS workload and the scheduling frequency of key updates. Scalability analysis indicates that while communication overhead scales linearly O(N) with swarm size, resource-constrained helper nodes may require adaptive offloading mechanisms to sustain continuous deep-learning inference without accelerating battery depletion.

8 Conclusion

This paper presents TSCID, a lightweight cross-layer security framework designed for resource-constrained UAV swarm networks. The proposed framework integrates time-indexed secure communication and intrusion detection modeling within a unified architecture to support lightweight authenticated communication, replay protection, and real-time behavioral anomaly detection for UAV swarm environments. By combining lightweight AEAD-based packet protection, dynamic session-key management, and resource-aware IDS optimization, the framework aims to achieve a practical balance between communication security, computational efficiency, and deployment feasibility on edge-class UAV platforms.

To support lightweight secure communication, the proposed framework employs time-indexed session-key derivation and authenticated packet verification with low computational overhead. A security analysis under standard symmetric-key cryptographic assumptions is provided to evaluate confidentiality, integrity, authentication, replay resistance, and session-key isolation properties under the defined threat model. In addition, a lightweight DNN-based IDS optimized through structured pruning is integrated into the framework to enable efficient anomaly detection suitable for resource-constrained UAV-edge environments.

Experimental results on Raspberry Pi platforms demonstrate that the proposed framework reduces communication latency and energy consumption compared with conventional PKI-based approaches while maintaining effective intrusion detection performance. The results indicate that the proposed deployment-oriented architecture is feasible for lightweight and moderate-scale UAV swarm communication environments requiring both secure transmission and real-time behavioral monitoring.

While the current architecture establishes a solid baseline, deploying TSCID within heterogeneous UAV swarm environments introduces distinct practical challenges. Variations in computational capabilities, communication latencies, synchronization stabilities, and onboard energy constraints among dynamic UAV nodes can significantly complicate lightweight session management, real-time authenticated packet-verification timing, and distributed IDS workload balancing. Consequently, such operational asymmetries may degrade synchronization consistency and call for flexible resource scheduling across the swarm. To mitigate these constraints, our future research will focus on developing adaptive, resource-aware coordination mechanisms, lightweight distributed trust management protocols, and comprehensive heterogeneous UAV swarm optimization strategies to ensure scalable and long-term security resilience.

Acknowledgement: The authors would like to thank all colleagues and students who contributed to this study. We are grateful to D. D. Yao and C. F. Tsai for their collaboration during preliminary investigations. During the preparation of this manuscript, the authors utilized ChatGPT for linguistic editing, including grammar correction, spelling checks, punctuation, style improvements. The authors have carefully reviewed and revised the output and accept full responsibility for all content.

Funding Statement: This work was supported by the National Science and Technology Council, Taiwan, under Project NSTC 113-2221-E-029-012-MY2.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Li-Woei Chen, Kun-Lin Tsai, Fang-Yie Leu, and Chao-Tung Yang; algorithm design: Li-Woei Chen, Kun-Lin Tsai, Fang-Yie Leu; software, experiment and simulation, Wei-Zong Liang; draft manuscript preparation: Li-Woei Chen, Chao-Tung Yang and Kun-Lin Tsai. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The CIC-IDS2017 dataset used in this study is publicly available from the Canadian Institute for Cybersecurity at the University of New Brunswick (UNB): https://www.unb.ca/cic/datasets/ids-2017.html.

Ethics Approval: Not applicable.

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

References

1. Khaga SYK, Avireneni RT, Koneru SH, Yelkoti NKKR. Towards proactive cloud security: a survey on ML and deep learning-based intrusion detection systems. J Contemp Edu Theo Artif Intell. 2025;4(1):100116. doi:10.47991/2996-4954/JCETAI-116.
2. Alshinwan M, Batyha RM, Alayed W, Alqahtany SS, Abuowaida S, Mashagba HA, et al. Enhancing intrusion detection systems using hybrid AI-based approaches. Comput Mater Continua. 2026;87(2):1–10. doi:10.32604/cmc.2026.072806.

3. Almofarreh M, Alshahrani A, Alharbi NH, Ahmed AO, Alshahrani H, Alzahrani A, et al. Boosting cybersecurity: a zero-day attack detection approach using equilibrium optimiser with deep learning model. Comput Model Eng Sci. 2025;145(2):2631–56. doi:10.32604/cmes.2025.070545.
4. Sharma J, Mehra PS. Secure communication in IOT-based UAV networks: a systematic survey. Internet Things. 2023;23(10):100883. doi:10.1016/j.iot.2023.100883.
5. Sarkar S, Shafaei S, Jones TS, Totaro MW. Secure communication in drone networks: a comprehensive survey of lightweight encryption and key management techniques. Drones. 2025;9(8):583. doi:10.3390/drones9080583.
6. Pu C, Wall A, Choo KR, Ahmed I, Lim S. A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of drones environment. IEEE Internet Things J. 2022;9(12):9918–33. doi:10.1109/JIOT.2022.3163367.
7. Khan MA, Ullah I, Kumar N, Oubbati OS, Qureshi IM, Noor F, et al. An efficient and secure certificate-based access control and key agreement scheme for flying ad-hoc networks. IEEE Trans Veh Technol. 2021;70(5):4839–51. doi:10.1109/TVT.2021.3055895.
8. Hassija V, Chamola V, Agrawal A, Goyal A, Luong NC, Niyato D, et al. Fast, reliable, and secure drone communication: a comprehensive survey. IEEE Commun Surv Tutor. 2021;23(4):2802–32. doi:10.1109/COMST.2021.3097916.
9. Yu Z, Wang Z, Yu J, Liu D, Herbert Song H, Li Z. Cybersecurity of unmanned aerial vehicles: a survey. IEEE Aerosp Electron Syst Mag. 2024;39(9):182–215. doi:10.1109/MAES.2023.3318226.
10. Cecchinato N, Toma A, Drioli C, Oliva G, Sechi G, Foresti GL. Secure real-time multimedia data transmission from low-cost UAVs with a lightweight AES encryption. IEEE Commun Mag. 2023;61(5):160–5. doi:10.1109/MCOM.001.2200611.
11. Alexan W, Aly L, Korayem Y, Gabr M, El-Damak D, Fathy A, et al. Secure communication of military reconnaissance images over UAV-assisted relay networks. IEEE Access. 2024;12:78589–610. doi:10.1109/ACCESS.2024.3407838.
12. Hidawi K, Carminati B, Ferrari E. AeroTrust-5D: a five-dimensional trust management system for optimizing AAV cooperation and path planning in urban areas. IEEE Trans Veh Technol. 2025;74(12):18369–84. doi:10.1109/TVT.2025.3583143.
13. Ahmad Awan K, Uddin M, Althobaiti MM, Alaidaros H, Alsalman D, Farouk A. TUB-IoT: quantum trust-based big data UAV architecture for security-centric Internet of Things. IEEE Open J Commun Soc. 2026;7:1235–48. doi:10.1109/OJCOMS.2026.3658464.
14. Dobraunig C, Eichlseder M, Mendel F, Schläffer M. Ascon v1.2: lightweight authenticated encryption and hashing. J Cryptol. 2021;34(3):33. doi:10.1007/s00145-021-09398-9.
15. Baliński S, Śniatała P, Sobieraj M, Grocholewska-Czuryło A, Xie J, Ren S. Analysis of selected cryptographic algorithms for data transmission in airborne networks. In: Proceedings of the 32nd International Conference on Mixed Design of Integrated Circuits and System (MIXDES); 2025 Jun 26–27; Szczecin, Poland. p. 206–10. doi:10.23919/mixdes66264.2025.11092059.
16. Vigneswaran RK, Vinayakumar R, Soman KP, Poornachandran P. Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. In: Proceedings of the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT); 2018 Jul 10–12. Bangalore, India. p. 1–6. doi:10.1109/icccnt.2018.8494096.
17. Choudhary G, Sharma V, You I, Yim K, Chen IR, Cho JH. Intrusion detection systems for networked unmanned aerial vehicles: a survey. In: Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC); 2018 Jun 25–29. Limassol, Cyprus. p. 560–5. doi:10.1109/iwcmc.2018.8450305.
18. AL-Syouf RA, Bani-Hani RM, AL-Jarrah OY. Machine learning approaches to intrusion detection in unmanned aerial vehicles (UAVs). Neural Comput Appl. 2024;36(29):18009–41. doi:10.1007/s00521-024-10306-y.
19. Hassler SC, Ahmad Mughal U, Ismail M. Cyber-physical intrusion detection system for unmanned aerial vehicles. IEEE Trans Intell Transp Syst. 2024;25(6):6106–17. doi:10.1109/TITS.2023.3339728.

20. Ihekoronye VU, Ajakwe SO, Lee JM, Kim DS. DroneGuard: an explainable and efficient machine learning framework for intrusion detection in drone networks. IEEE Internet Things J. 2025;12(7):7708–22. doi:10.1109/JIOT.2024.3519633.
21. Mohammed AB, Fourati LC. Investigation on datasets toward intelligent intrusion detection systems for Intra and inter-UAVs communication systems. Comput Secur. 2025;150(1):104215. doi:10.1016/j.cose.2024.104215.
22. Ahmad Mughal U, Elshazly A, Atat R, Ismail M. Generalizable topology-aware GNN-based intrusion detection system for UAV swarms. IEEE Internet Things J. 2026;13(1):1569–80. doi:10.1109/JIOT.2025.3630488.
23. Han S, Pool J, Tran J, Dally WJ. Learning both weights and connections for efficient neural network. In: Proceedings of the 29th International Conference on Neural Information Processing Systems; 2015 Dec 7–12; Montreal, QC, Canada. New York, NY, USA: The Association for Computing Machinery (ACM); 2015. p. 1135–43.
24. Luo JH, Wu J, Lin W. ThiNet: a filter level pruning method for deep neural network compression. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. p. 5068–76. doi:10.1109/iccv.2017.541.
25. Kumar A, Shaikh AM, Li Y, Bilal H, Yin B. Pruning filters with L1-norm and capped L1-norm for CNN compression. Appl Intell. 2021;51(2):1152–60. doi:10.1007/s10489-020-01894-y.
26. Fang G, Ma X, Song M, Bi Mi M, Wang X. DepGraph: towards any structural pruning. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. p. 16091–101. doi:10.1109/cvpr52729.2023.01544.
27. Guo Z, Cao J, Ren X, Zhou Y, Cheng L, Yin P, et al. LDST-UAVS: a lightweight data secure transmission protocol for unmanned aerial vehicle swarms in emergency rescue scenarios. IEEE Trans Netw Serv Manag. 2026;23:2258–79. doi:10.1109/TNSM.2026.3656973.
28. Halli Sudhakara S, Haghnegahdar L. Security enhancement in AAV swarms: a case study using federated learning and SHAP analysis. IEEE Open J Intell Transp Syst. 2025;6(24):335–45. doi:10.1109/OJITS.2025.3550792.