

# Cryptographic Authentication for UAV Swarm Security: A Comparative Analysis of Detection Methods Against Spoofing Attacks

Sang Xing\*, Laxima Niure Kandel†

Department of Electrical Engineering and Computer Science  
Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA  
Email: \*xings@my.erau.edu, †niurekal@erau.edu

**Abstract**—Unmanned Aerial Vehicles (UAVs) increasingly rely on Remote ID broadcast systems for airspace safety, yet these systems are vulnerable to spoofing attacks including phantom UAV injection, position falsification, and coordinated multi-UAV attacks. This paper presents a comprehensive evaluation of cryptographic versus graph-based detection methods for identifying malicious UAVs in swarm networks. We implement and evaluate four detection approaches: (1) Ed25519 digital signature verification, (2) spectral graph analysis, (3) centrality-based anomaly detection, and (4) machine learning with graph embeddings. Our experimental results demonstrate that cryptographic authentication achieves perfect detection (100% True Positive Rate, 0% False Positive Rate) for phantom and coordinated attacks across all mobility scenarios, while graph-based heuristic methods exhibit inconsistent performance with false positive rates ranging from 0% to 100%. For position falsification attacks, all methods show fundamental limitations rooted in the attack’s topology-preserving nature. We conclude that only cryptographic authentication provides provable security guarantees suitable for production UAV systems, with acceptable computational overhead (~58ms per detection cycle for 30 UAVs). Our findings establish that cryptographic methods transform UAV security from a probabilistic problem (heuristics with variable detection rates) to a deterministic problem with mathematical security guarantees. The complete implementation is available as open-source software at <https://github.com/Sang-Buster/Argus>.

**Index Terms**—UAV security, Remote ID, cryptographic authentication, Ed25519, graph anomaly detection, spoofing attacks, swarm networks

## I. INTRODUCTION

### A. Motivation

The proliferation of Unmanned Aerial Vehicles (UAVs) in commercial and recreational airspace has necessitated the development of Remote ID broadcast protocols, which enable ground observers and other aircraft to identify and track nearby UAVs [1]. However, these broadcast-based identification systems are inherently vulnerable to spoofing attacks, where malicious actors inject false messages or manipulate transmitted data to deceive monitoring systems. Such attacks pose significant risks to airspace safety, privacy, and security [2], [3].

Remote ID systems face three primary attack vectors. First, phantom UAV injection occurs when attackers broadcast fabricated messages appearing to originate from non-existent UAVs, potentially overwhelming airspace manage-

ment systems or masking unauthorized operations. Second, position falsification involves compromised legitimate UAVs reporting false GPS coordinates, enabling unauthorized flight in restricted zones while appearing compliant. Third, coordinated multi-UAV attacks employ multiple compromised UAVs executing synchronized spoofing to achieve objectives beyond single-UAV capabilities. These threats collectively undermine the fundamental trust assumptions of broadcast-based identification systems.

### B. Related Work

Prior research on UAV security has explored various detection approaches:

**Graph-based methods** leverage network topology analysis for anomaly detection. Spectral graph theory [4], [5] has been applied to detect structural anomalies through eigenvalue analysis of the graph Laplacian. Centrality-based approaches [6] identify suspicious nodes based on abnormal betweenness, degree, or closeness centrality metrics.

**Machine learning approaches** employ supervised and unsupervised learning on graph features. Node2Vec [7] generates graph embeddings through random walks, enabling anomaly detection via classification or clustering algorithms.

**Cryptographic methods** provide authentication through digital signatures. Ed25519 [8], [9], based on elliptic curve cryptography, offers fast signing and verification suitable for resource-constrained embedded systems. Hardware acceleration techniques [10] further improve performance for embedded deployments.

However, **no comprehensive comparison** exists evaluating cryptographic versus graph-based approaches specifically for UAV Remote ID security across multiple attack scenarios and mobility conditions.

### C. Contributions

This paper makes the following contributions:

- 1) **First systematic evaluation** comparing cryptographic and graph-based detection methods for UAV spoofing attacks
- 2) **Novel hybrid approach** combining cryptographic authentication with spectral analysis

- 3) **Comprehensive mobility analysis** examining detector performance in stationary versus mobile swarm scenarios
- 4) **Quantitative overhead analysis** establishing the computational cost-benefit trade-off of cryptographic security
- 5) **Open-source framework** (Argus) enabling reproducible UAV security research

#### D. Paper Organization

Section II describes our threat model and system architecture. Section III details the four detection methods. Section IV presents experimental methodology. Section V reports comprehensive results across three attack types and two mobility scenarios. Section VI analyzes the findings and discusses implications. Section X concludes with recommendations for production deployment.

## II. SYSTEM MODEL AND THREAT MODEL

### A. UAV Swarm Network Model

We model a UAV swarm [11] as a dynamic communication graph  $G(t) = (V(t), E(t))$  where vertices  $V(t) = \{v_1, v_2, \dots, v_n\}$  represent  $n$  UAVs at time  $t$ , and edges  $E(t) \subseteq V(t) \times V(t)$  represent bidirectional communication links. Edge existence is determined by Euclidean distance such that  $(v_i, v_j) \in E(t)$  if and only if  $\|p_i(t) - p_j(t)\| \leq r_{comm}$ , where  $\|\cdot\|$  denotes the Euclidean norm,  $p_i(t) \in \mathbb{R}^3$  denotes the position of UAV  $i$  at time  $t$ , and  $r_{comm}$  represents the communication range.

**Remote ID Messages:** Each UAV broadcasts periodic messages containing:

```
RemoteID_Message = {
  uav_id: String,
  position: (x, y, z) in R^3,
  velocity: (vx, vy, vz) in R^3,
  timestamp: Unix_Time,
  signature: Bytes[64] (optional)
}
```

### B. Threat Model

We consider three attack scenarios:

#### Attack 1: Phantom UAV Injection ( $A_1$ )

An adversary injects  $k$  fabricated messages appearing to originate from non-existent UAVs. Formally, at time  $t_{attack}$ :

$$V(t_{attack}) \leftarrow V(t_{attack}) \cup \{phantom_1, \dots, phantom_k\} \quad (1)$$

where phantom UAVs lack valid cryptographic credentials.

#### Attack 2: Position Falsification ( $A_2$ )

An adversary compromises  $m$  legitimate UAVs, causing them to report false positions while maintaining valid cryptographic signatures:

$$\forall i \in Compromised : p'_i(t) = p_i(t) + \delta_i \quad (2)$$

where  $\delta_i$  is a position offset vector and  $\|\delta_i\| \leq M_{max}$  (maximum falsification magnitude).

#### Attack 3: Coordinated Attack ( $A_3$ )

Multiple phantom UAVs are injected in a coordinated geometric pattern (e.g., circular formation) to achieve specific spatial coverage or deception objectives.

**Adversary Capabilities:** We assume an adversary capable of broadcasting arbitrary RF signals in the Remote ID frequency band, capturing and analyzing legitimate UAV messages, compromising legitimate UAV flight controllers to enable position falsification, and coordinating multiple attack nodes for synchronized operations.

**Adversary Limitations:** However, the adversary cannot extract private keys from secure hardware elements such as TPM or SGX, cannot solve the discrete logarithm problem underlying Ed25519's security, and remains subject to physical constraints including RF propagation characteristics and message timing requirements.

### C. System Architecture

Our experimental framework (Argus) consists of five main components: UAV Swarm Simulation (physics, communications, mobility), Attack Injection Module (phantom, position, coordinated), Communication Graph construction  $G(t) = (V, E)$ , Crypto Detection (Ed25519, PKI) and Graph-Based Detection (spectral, centrality, ML), and Evaluation & Metrics (TPR, FPR, F1, detection time, reproducibility).

## III. DETECTION METHODS

We implement and evaluate four detection approaches representing the state-of-the-art in UAV security.

### A. Cryptographic Authentication

**Principle:** Verify message authenticity through Ed25519 digital signatures based on elliptic curve cryptography over Curve25519.

---

#### Algorithm 1 Cryptographic Detection

---

**Require:** Graph  $G(t)$ , Public Key Database PKI

**Ensure:** Set of anomalous UAV IDs

Initialize:  $Anomalous \leftarrow \emptyset$

**for** each node  $v \in V(G)$  **do**

$msg \leftarrow \text{latest\_message}(v)$

**if**  $msg.signature = \text{NULL}$  **then**

$Anomalous \leftarrow Anomalous \cup \{v\}$  {No signature}

**else if**  $v \notin \text{PKI}$  **then**

$Anomalous \leftarrow Anomalous \cup \{v\}$  {Unknown UAV}

**else if**  $\neg \text{Verify\_Ed25519}(msg, PKI[v])$  **then**

$Anomalous \leftarrow Anomalous \cup \{v\}$  {Invalid signature}

**end if**

**end for**

**return**  $Anomalous$

---

**Key Generation:** Each legitimate UAV generates an Ed25519 keypair  $(sk_i, pk_i)$  where the private key  $sk_i \in \{0, 1\}^{256}$  is stored securely on-board and the public key  $pk_i$  is registered in the system's PKI during initialization.

**Message Signing:** UAV  $i$  signs each Remote ID message  $m$  (containing position, velocity, timestamp) as:

$$\sigma_i = \text{Ed25519.Sign}(sk_i, m) \quad (3)$$

where  $\sigma_i$  is the resulting digital signature.

**Signature Verification:** Detector verifies signature validity:

$$\text{Ed25519.Verify}(pk_i, m, \sigma_i) \stackrel{?}{=} \text{True} \quad (4)$$

**Security Guarantee:** Based on the computational hardness of the discrete logarithm problem on Curve25519. Forging a valid signature requires  $O(2^{128})$  operations, computationally infeasible even for nation-state adversaries.

**Limitations:** Cannot detect position falsification by compromised UAVs with valid keys (they can sign false position data).

### B. Spectral Graph Analysis

**Principle:** Detect topological anomalies through eigenvalue analysis of the graph Laplacian matrix.

**Graph Laplacian:** For graph  $G = (V, E)$ , the Laplacian matrix is:

$$L = D - A \quad (5)$$

where  $D$  is the degree matrix and  $A$  is the adjacency matrix.

---

#### Algorithm 2 Spectral Detection

---

**Require:** Graph  $G(t)$ , Baseline statistics  $(\mu_\lambda, \sigma_\lambda)$

**Ensure:** Set of anomalous UAV IDs

Compute Laplacian  $L = D - A$

Compute eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

Compute eigenvectors  $u_1, u_2, \dots, u_n$

// Algebraic connectivity check

$z_{AC} \leftarrow |\lambda_2 - \mu_{\lambda_2}| / \sigma_{\lambda_2}$  {z-score, where  $\mu_{\lambda_2}, \sigma_{\lambda_2}$  are baseline mean, std}

// Per-node anomaly scoring

**for** each node  $v \in V$  **do**

$score_{degree} \leftarrow$  z-score of  $deg(v)$

$score_{eigenvector} \leftarrow$  residual in subspace projection

$score_{position} \leftarrow$  topology-position consistency

$combined\_score \leftarrow 0.25 \cdot score_{degree} + 0.25 \cdot z_{AC} + 0.20 \cdot score_{eigenvector} + 0.30 \cdot score_{position}$

**if**  $combined\_score > threshold$  **then**

        flag  $v$  as anomalous

**end if**

**end for**

---

**Key Features:** The algebraic connectivity  $\lambda_2$  (Fiedler value) indicates overall graph connectivity, eigenvector centrality identifies structurally important nodes, and the spectral gap  $\lambda_n - \lambda_2$  (where  $\lambda_n$  is the largest eigenvalue) characterizes graph robustness against perturbations.

**Threshold:** Adaptive threshold  $\tau = 2.5$  standard deviations for spatial domain (1000×1000m).

### C. Centrality-Based Detection

**Principle:** Identify anomalous nodes through abnormal centrality metrics.

**Centrality Metrics:**

1) **Degree Centrality:**  $C_D(v) = \frac{deg(v)}{n-1}$   
where  $deg(v)$  is the degree (number of edges) of node  $v$

2) **Betweenness Centrality:**  $C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$   
where  $\sigma_{st}$  is the number of shortest paths from  $s$  to  $t$ , and  $\sigma_{st}(v)$  is the number of those paths passing through  $v$

3) **Closeness Centrality:**  $C_C(v) = \frac{n-1}{\sum_{u \neq v} d(v, u)}$   
where  $d(v, u)$  is the shortest path length between nodes  $v$  and  $u$

**Detection:** Flag nodes with z-scores exceeding threshold  $\tau = 6.0$  in any centrality metric.

### D. Machine Learning Detection

**Principle:** Learn normal graph structure patterns and identify deviations using Isolation Forest [12], an unsupervised anomaly detection algorithm. Graph embeddings from Node2Vec [7] combined with anomaly detection methods [13] have shown promise in network security applications [14].

**Feature Extraction:** For each node  $v$ , compute feature vector:

$$\mathbf{f}(v) = [C_D(v), C_B(v), C_{clustering}(v), C_C(v)] \quad (6)$$

where  $C_{clustering}(v)$  is the clustering coefficient:

$$C_{clustering}(v) = \frac{2|\{(u, w) : u, w \in N(v), (u, w) \in E\}|}{deg(v)(deg(v) - 1)}$$

and  $N(v)$  denotes the neighborhood (set of adjacent nodes) of  $v$ .

**Training:** Isolation Forest with contamination parameter  $\alpha = 0.15$  (expects 15% anomalies).

---

#### Algorithm 3 ML Detection

---

**Require:** Baseline graphs  $\{G_1, \dots, G_m\}$ , Test graph  $G(t)$

**Ensure:** Anomalous nodes

**Training Phase:**

Extract features  $F_{baseline} \leftarrow \{f(v) : v \in G_i, i = 1..m\}$

Train  $IF \leftarrow \text{IsolationForest}(F_{baseline}, contamination = 0.15)$

**Detection Phase:**

Extract features  $F_{test} \leftarrow \{f(v) : v \in G(t)\}$

Compute  $anomaly\_scores \leftarrow IF.decision\_function(F_{test})$

Flag nodes with scores  $< threshold$

---

## IV. EXPERIMENTAL METHODOLOGY

### A. Simulation Environment

**Implementation:** Python 3.10 with NetworkX (graph operations), NumPy (numerical computation), scikit-learn (machine learning), and pycryptodome (cryptographic primitives).

**Swarm Parameters:** We evaluate swarms containing  $n \in \{20, 25, 30\}$  UAVs within a spatial domain  $\Omega = [0, 1000] \times [0, 1000] \times [0, 200]$  meters. Each UAV maintains a communication range of  $r_{comm} = 200$  meters and broadcasts Remote ID messages at 1 Hz. Each simulation runs for 50 timesteps to capture attack injection, detection, and recovery phases.

**Mobility Model:** We consider two mobility scenarios to assess detector robustness. In stationary mode, UAVs remain at fixed positions throughout the simulation. In mobile mode, UAVs move toward randomly selected destinations at velocities of  $v = 10 \pm 2$  m/s, with new destinations assigned upon arrival to simulate realistic patrol or surveillance patterns.

**Reproducibility:** All experiments use fixed random seed (seed=42) and standardized configuration parameters to ensure reproducibility.

### B. Attack Scenarios

**Scenario 1: Phantom UAV Attack** involves injecting  $k = 3$  phantom UAVs at  $t = 10$  seconds for a duration of 20 seconds. Phantom positions are randomly selected within communication range of legitimate UAVs to maximize network integration and detection difficulty.

**Scenario 2: Position Falsification** simulates the compromise of  $m = 4$  legitimate UAVs (approximately 13% of the swarm), each reporting false positions offset by  $\|\delta\| = 100$  meters in random directions. This magnitude represents significant spatial deception while maintaining communication topology.

**Scenario 3: Coordinated Attack** tests organized threat patterns by injecting  $k = 5$  phantom UAVs arranged in a circular formation with radius 50 meters, simulating sophisticated adversaries capable of geometric coordination.

### C. Evaluation Metrics

For each detector  $D$  and attack scenario  $A$ , we compute:

**True Positive Rate (Recall):**

$$TPR = \frac{TP}{TP + FN} \quad (7)$$

**False Positive Rate:**

$$FPR = \frac{FP}{FP + TN} \quad (8)$$

**Precision:**

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

**F1 Score** (harmonic mean):

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (10)$$

**Detection Time:** Average time per detection cycle (milliseconds)

where  $TP$  = true positives (malicious UAVs correctly identified),  $FP$  = false positives (legitimate UAVs incorrectly flagged),  $TN$  = true negatives,  $FN$  = false negatives.

### D. Experimental Procedure

Each experiment follows a standardized eight-phase protocol. During initialization, we create a swarm with  $n$  UAVs positioned randomly within the spatial domain. The baseline collection phase runs for 30 timesteps without attacks, recording graphs  $\{G_1, \dots, G_{30}\}$  to characterize normal operation. We then train each detector on these baseline graphs to establish anomaly detection thresholds. Attack injection occurs at  $t = 10$  seconds according to the specified scenario. Throughout the detection phase, we run each detector at every timestep and record classification results. At  $t = 30$  seconds, we remove the attack to test detector recovery behavior. The post-attack phase continues for 20 additional timesteps to observe how quickly detectors return to normal operation. Finally, metrics computation aggregates detection results across all timesteps to calculate TPR, FPR, and F1 scores.

**Statistical Rigor:** Each configuration is tested with controlled random seeds to ensure reproducibility. Detection time is measured using high-precision timers.

## V. EXPERIMENTAL RESULTS

### A. Stationary Swarm Results

Table I presents detection performance for stationary swarms (30 UAVs,  $r_{comm} = 200$ m).

**Key Observations:** The results validate our primary hypothesis that cryptographic detection achieves perfect performance (TPR=1.0, FPR=0.0, F1=1.0) for phantom and coordinated attacks. Notably, spectral detection also achieves perfect performance for these attack types, demonstrating that topology-based methods can be effective when attacks create structural anomalies. Position falsification proves undetectable by topology-preserving methods (spectral, crypto) as expected, since falsifying position without changing communication links creates no graph structural changes. Centrality and ML detectors exhibit unacceptable false positive rates (87–100%), rendering them impractical for production deployment. Regarding computational overhead, cryptographic verification requires approximately  $30\times$  more time than spectral analysis (58ms vs 2ms) but remains acceptable for real-time operation at 1 Hz broadcast rates.

### B. Mobility Impact Analysis

Table II compares detector performance in mobile swarms where UAVs move at 10 m/s.

**Key Observations:** Cryptographic detection exhibits complete mobility invariance, maintaining perfect performance (100% TPR, 0% FPR) regardless of UAV movement patterns. This confirms that signature verification operates independently of network dynamics. In contrast, spectral detection degrades significantly under mobility, with false positive rates increasing from 0% to 62.5% for phantom attacks. This degradation occurs because movement-induced topology changes create false anomaly signals that spectral methods cannot distinguish from genuine attacks. The sensitivity of graph-based methods to dynamics manifests as shifts in eigenvalue

TABLE I  
DETECTION PERFORMANCE IN STATIONARY SWARMS

Attack Type	Detector	TPR	FPR	Precision	Recall	F1	Time (ms)
Phantom	Spectral	1.000	0.000	1.000	1.000	1.000	1.97
	Centrality	0.667	1.000	0.062	0.667	0.114	1.24
	<b>Crypto</b>	<b>1.000</b>	<b>0.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	57.90
	ML	0.333	0.933	0.034	0.333	0.062	4.88
Position	Spectral	0.000	0.000	0.000	0.000	0.000	1.75
	Centrality	1.000	1.000	0.100	1.000	0.182	1.12
	<b>Crypto</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	57.27
	ML	0.667	0.963	0.071	0.667	0.129	4.63
Coordinated	Spectral	1.000	0.000	1.000	1.000	1.000	1.90
	Centrality	1.000	1.000	0.091	1.000	0.167	1.51
	<b>Crypto</b>	<b>1.000</b>	<b>0.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	57.21
	ML	1.000	0.867	0.103	1.000	0.188	5.11

TABLE II  
STATIONARY VS MOBILE SWARM PERFORMANCE

Attack	Detector	Stationary (TPR/FPR)	Mobile (TPR/FPR)
Phantom	Spectral	100.0%/0.0%	99.2%/62.5%
	<b>Crypto</b>	<b>100.0%/0.0%</b>	<b>100.0%/0.0%</b>
Position	Spectral	0.0%/0.0%	1.9%/3.2%
	<b>Crypto</b>	<b>0.0%/0.0%</b>	<b>0.0%/0.0%</b>
Coordinated	Spectral	98.5%/0.0%	99.0%/6.7%
	<b>Crypto</b>	<b>100.0%/0.0%</b>	<b>100.0%/0.0%</b>

distributions, necessitating mobility-aware threshold adaptation for practical deployment.

### C. Detection Time Analysis

Table III presents the computational overhead comparison across all detection methods. Cryptographic verification incurs approximately  $30\times$  computational overhead compared to spectral analysis, primarily due to Ed25519 signature verification ( $\sim 0.1\text{ms}$  per UAV), graph traversal ( $\sim 10\text{ms}$  for 30 UAVs), and message serialization ( $\sim 5\text{ms}$ ).

TABLE III  
COMPUTATIONAL OVERHEAD COMPARISON

Detector	Mean (ms)	Std Dev (ms)	Overhead
Spectral	1.87	0.12	$1.0\times$
Centrality	1.29	0.17	$0.7\times$
<b>Crypto</b>	<b>57.46</b>	<b>2.31</b>	<b><math>30.7\times</math></b>
ML	4.87	0.51	$2.6\times$

For a 30-UAV swarm at 1 Hz broadcast rate, total overhead is 58ms per detection cycle, well within real-time constraints ( $< 1000\text{ms}$  budget).

**Trade-off:** Pay 56ms additional latency for 100% security guarantee (TPR=1.0, FPR=0.0) versus probabilistic heuristics.

## VI. ANALYSIS AND DISCUSSION

### A. Why Cryptographic Methods Succeed

**Mathematical Foundation:** Ed25519 security [8], [9] relies on the computational hardness assumption that solving the

discrete logarithm problem on elliptic curves requires super-polynomial time. Specifically:

**Theorem 1 (Informal):** Given a public key  $pk = [sk]G$  where  $G$  is the generator point on Curve25519 and  $sk$  is the private key, computing  $sk$  from  $pk$  requires expected time  $O(2^{128})$  using best-known algorithms (Pollard's rho).

**Consequence:** Phantom UAVs cannot forge valid signatures without the private key, enabling perfect detection.

**Advantage over Heuristics:** While graph-based methods detect anomalies probabilistically (requiring threshold tuning and accepting trade-offs between TPR and FPR), cryptographic methods provide **deterministic guarantees**: a valid signature proves authenticity with probability  $1 - \epsilon$  where  $\epsilon < 2^{-128}$  (negligible).

### B. Why Graph-Based Methods Fail or Degrade

**Configuration Sensitivity:** Spectral and centrality detectors require threshold calibration specific to spatial domain size (which affects graph density), communication range (which affects node degree distribution), and swarm size (which affects eigenvalue magnitudes). Our experiments demonstrate this sensitivity clearly. Increasing spatial domain from  $500\times 500\text{m}$  to  $1000\times 1000\text{m}$  required adjusting the spectral threshold from 4.0 to 2.5 (lower for reduced density) and the centrality threshold from 3.0 to 6.0 (higher to reduce FPR).

**Mobility Vulnerability:** Movement creates dynamic topology changes that shift baseline eigenvalue distributions, create transient centrality anomalies as UAVs enter and leave neighborhoods, and generate false positive signals when benign topology changes are mistaken for attacks.

**Evasion Potential:** Adaptive attackers [15] can study baseline statistics and craft attacks that blend with normal variations. For instance, injecting phantoms in low-density regions minimizes eigenvalue impact, matching legitimate UAV degree distributions evades centrality detection, and exploiting threshold gaps allows operation just below detection thresholds.

**Fundamental Limitation:** Graph methods detect topology anomalies. Attacks preserving topology (position falsification with  $\|\delta\| < r_{comm}$ ) are inherently undetectable.

### C. Position Falsification: A Special Case

**Observation:** All detectors achieve 0% TPR on position falsification attacks (Table I).

**Root Cause:** When compromised UAVs report false positions  $p'_i = p_i + \delta$  with  $\|\delta\| < r_{comm}$ , the communication graph remains unchanged:

$$G(p') = G(p) \implies L(p') = L(p) \quad (11)$$

Since spectral eigenvalues depend only on graph structure (not node labels/positions), position falsification is **topologically invisible**.

**Cryptographic Limitation:** Valid signatures only prove message authenticity (sender identity and data integrity), NOT data truthfulness. A compromised UAV can sign false position data with its legitimate private key.

**Mitigation Strategies** (future work): Position falsification detection requires orthogonal approaches beyond graph topology analysis. Multi-lateration [16], [17] could verify reported positions against distance measurements from neighboring UAVs, exploiting physical propagation constraints. Physics-based validation could detect impossible velocities or accelerations inconsistent with UAV dynamics. Byzantine consensus protocols [18], [19] could require agreement from  $\lfloor n/3 \rfloor + 1$  UAVs before accepting position claims, providing fault tolerance against compromised nodes.

### D. ML Detector Performance Analysis

**Observation:** ML detector exhibits high FPR (87–96%) despite perfect recall (TPR=1.0) for phantom/coordinated attacks.

**Root Cause Analysis:** The high false positive rate stems from three primary factors. First, contamination mismatch occurs because the initial parameter  $\alpha = 0.05$  expected 5% anomalies while actual attack density reached approximately 9% (3 phantoms among 33 total nodes), causing the Isolation Forest to flag additional legitimate UAVs. Second, the 4D feature vector (degree, betweenness, clustering, closeness) captures limited topological information compared to spectral eigenvectors which encode full graph structure. Third, the Isolation Forest trained on only 30 baseline graphs may overfit to training data patterns and fail to generalize to topology variations during attack scenarios.

Increasing contamination to  $\alpha = 0.15$  reduced FPR from 96.7% to 93.3%, a modest improvement suggesting fundamental approach limitations rather than mere parameter tuning issues. We conclude that the current ML approach (Node2Vec + Isolation Forest) remains unsuitable for production deployment. Future work should explore Graph Neural Networks (GNN) for richer feature learning, temporal models (LSTM, GRU) for trajectory analysis, and ensemble methods combining multiple detector outputs.

## VII. NOVELTY AND CONTRIBUTIONS

### A. Novel Contributions

This work makes three novel contributions to UAV security research:

**1. First Comprehensive Cryptographic Comparison:** Prior work focuses exclusively on either cryptographic or graph-based approaches without systematic comparison. We provide the first comprehensive evaluation demonstrating quantitative overhead analysis ( $30\times$  computational cost), mobility impact assessment (crypto invariant, heuristics degrade), and reproducibility analysis (crypto consistent, heuristics configuration-sensitive).

**2. Threshold Scaling Analysis:** We identify and characterize the spatial scaling problem for graph-based detectors where optimal thresholds depend on spatial domain size, communication range, and swarm size according to  $\tau_{optimal}(|\Omega|) = \tau_0 \cdot f(|\Omega|, r_{comm}, n)$ , where  $\tau_0$  is the base threshold and  $f$  is a scaling function. Our experiments reveal that small domains ( $500\times 500\text{m}$ ) require spectral threshold 4.0 and centrality threshold 3.0, while large domains ( $1000\times 1000\text{m}$ ) require spectral threshold 2.5 and centrality threshold 6.0.

**3. Mobility Impact Quantification:** We quantify mobility-induced performance degradation, showing that spectral FPR increases from 0% to 62.5% for phantom attacks in mobile scenarios while cryptographic FPR remains at 0%. This demonstrates cryptographic methods' fundamental advantage of mobility-invariant security.

### B. System Architecture Novelty

The Argus framework introduces several architectural innovations. It provides unified attack simulation supporting phantom, position falsification, and coordinated attacks within a single framework. Multi-modal mobility enables configurable stationary and mobile scenarios with physics-based movement models. Standardized evaluation ensures reproducible testing through controlled random seeds and standardized parameters. Real-time visualization offers live animated display of attack progression and detection results, facilitating intuitive understanding of detector behavior.

### C. Practical Implications

For the UAV industry, our results demonstrate that Ed25519 signatures provide suitable real-time performance for Remote ID applications with 58ms latency, establish that heuristic-only approaches prove insufficient for security-critical applications, and provide open-source reference implementations supporting standardization efforts. For the security research community, we quantify the security-performance trade-off ( $30\times$  overhead for provable security), identify position falsification as an unsolved challenge requiring multi-modal verification approaches, and establish empirical baselines for future UAV security research.

## VIII. DISCUSSION

### A. Cryptographic Security vs Heuristic Detection

Our results establish a fundamental distinction:

**Heuristic Methods (Spectral, Centrality, ML)** provide probabilistic guarantees, detecting anomalies with probability  $P(\text{detect}|\text{attack})$  that varies with configuration parameters. Their performance depends heavily on spatial scale, network

density, and mobility patterns. Adaptive attackers can potentially evade detection by crafting attacks below configured thresholds. These methods offer fast detection (1–5ms) and require no key distribution infrastructure, making deployment straightforward but security guarantees weak.

**Cryptographic Methods (Ed25519)** provide deterministic guarantees with detection probability  $1 - 2^{-128}$ , where the error term is cryptographically negligible. Performance remains invariant to spatial scale and mobility patterns, eliminating configuration sensitivity. Evasion requires solving the discrete logarithm problem, which is computationally infeasible for any adversary. The computational cost is higher (58ms detection time, representing  $30\times$  overhead), and deployment requires public key distribution infrastructure. However, these costs purchase provable security rather than probabilistic detection.

**Recommendation:** For security-critical UAV applications (urban air mobility, critical infrastructure inspection, emergency response), cryptographic authentication is **mandatory**. Heuristic methods may serve as supplementary monitoring tools but should never be sole security mechanisms.

### B. Hybrid Approach Feasibility

A hybrid detector combining cryptographic authentication and spectral analysis could potentially address both phantom UAVs and position falsification:

**Phase 1:** Cryptographic filtering eliminates all phantom UAVs (100% TPR, 0% FPR)

**Phase 2:** On crypto-verified UAVs, apply spectral analysis to detect position-based topology inconsistencies

However, our results show spectral analysis achieves 0% TPR on position attacks even after crypto filtering, indicating that position falsification with  $\|\delta\| < r_{comm}$  creates no detectable topology changes.

**Conclusion:** Hybrid approach does not improve position attack detection. Alternative approaches required including multi-lateration [17], physics-based validation, and Byzantine consensus [18].

### C. Limitations

Several limitations constrain the generality of our findings. All tested methods fail to detect position falsification attacks when falsified positions preserve communication topology, representing a fundamental open problem requiring alternative sensor modalities. Cryptographic methods cannot detect insider threats where legitimate UAVs possessing valid keys behave maliciously, as signatures only verify identity and integrity, not behavioral correctness. The  $30\times$  computational overhead of cryptographic verification compared to spectral analysis may limit scalability to very large swarms exceeding 1000 UAVs, though hardware acceleration could mitigate this constraint. Deployment requires secure key distribution infrastructure (PKI), adding operational complexity compared to infrastructure-free heuristic methods. Finally, our experiments focus on swarms of 20–30 UAVs; validation with larger-scale deployments remains necessary to confirm scalability predictions.

### D. Threat Landscape Evolution

**Current Threats:** Our cryptographic approach successfully addresses phantom UAV injection and coordinated attacks with 100% detection rates, providing complete protection against unauthorized identity spoofing.

**Emerging Threats:** Several threat vectors remain unaddressed by current methods. GPS spoofing [20], [21] affects all UAVs simultaneously by manipulating satellite signals rather than individual messages. RF jamming creates denial-of-service conditions that prevent legitimate communication. AI-powered adaptive attacks [15] could learn detector thresholds through repeated probing and craft evasive attacks. Quantum computing threatens the long-term viability of Ed25519 through Shor’s algorithm [22], necessitating post-quantum cryptographic alternatives [23], [24] for future deployments.

**Future-Proofing:** Ed25519 [8] remains secure against classical computers but may be vulnerable to quantum attacks via Shor’s algorithm [22]. Post-quantum alternatives such as CRYSTALS-Dilithium [23] and Falcon [24] should be considered for long-term deployments to ensure quantum-resistant authentication.

## IX. FUTURE WORK

### A. Short-Term Extensions

Several near-term extensions could enhance detector capabilities. Implementing a position verification layer using multi-lateration [16], [17] with signal strength and time-of-arrival measurements from neighboring UAVs could address the position falsification blind spot. Developing mobility-aware threshold adjustment functions such as  $\tau(v_{swarm}) = \tau_0 + k \cdot \bar{v}$  would improve heuristic detector robustness in dynamic scenarios. Replacing the Isolation Forest with Graph Neural Networks [25], [26] such as Graph Convolutional Networks (GCN) or Graph Attention Networks (GAT) could improve ML feature learning and reduce false positive rates. Testing with swarms of 100–1000 UAVs [11] would validate scalability predictions and identify performance bottlenecks.

### B. Long-Term Research Directions

Long-term research should address quantum threats and deployment scalability. Evaluating lattice-based signatures such as CRYSTALS-Dilithium [23] and Falcon [24] would provide quantum-resistant authentication for long-term system deployments. Implementing Byzantine Fault Tolerance consensus protocols [18], [19] could enable distributed position verification resistant to compromised nodes. Developing federated learning approaches would enable privacy-preserving collaborative anomaly detection across multiple swarm operators without sharing raw operational data. Hardware acceleration [10] using GPU or FPGA implementations could optimize Ed25519 verification to achieve sub-millisecond latency, enabling support for very large swarms.

## X. CONCLUSION

This paper presents the first comprehensive comparison of cryptographic and graph-based detection methods for UAV spoofing attacks. Through extensive experimentation across three attack types (phantom UAV injection, position falsification, coordinated attacks) and two mobility scenarios (stationary and mobile swarms), we establish the following conclusions:

Cryptographic authentication using Ed25519 digital signatures achieves **perfect detection** (TPR=1.0, FPR=0.0, F1=1.0) for phantom and coordinated attacks across all tested scenarios, while graph-based heuristic methods exhibit inconsistent performance with false positive rates up to 100%.

Our evaluation reveals four critical findings. First, only cryptographic methods provide security guarantees based on computational hardness assumptions, while heuristic methods offer probabilistic detection subject to evasion. Second, mobility invariance distinguishes cryptographic approaches (zero performance degradation) from graph-based methods (FPR increases up to 62.5%). Third, position falsification remains an open problem for all tested methods, requiring multi-modal sensor fusion or consensus-based verification. Fourth, the computational overhead proves acceptable, with 58ms detection latency for 30 UAVs at 1 Hz broadcast rate meeting real-time requirements.

We recommend deploying Ed25519 cryptographic authentication as the primary security mechanism for all production UAV systems. Spectral analysis may supplement this as a monitoring and anomaly alerting tool but should not serve security-critical functions. Centrality and ML detectors in their current form should be avoided due to unacceptable false positive rates.

For security-critical UAV applications, cryptographic authentication is not optional but **mandatory**. Graph-based methods may complement but cannot replace cryptographic security.

Our open-source Argus framework enables reproducible research in this domain and provides a reference implementation for Remote ID security evaluation.

## REFERENCES

- [1] ASTM International, "Standard specification for remote id and tracking," ASTM International, Tech. Rep. ASTM F3411-19, 2019. [Online]. Available: <https://www.astm.org/f3411-19.html>
- [2] S. Javaid, S. Zeadally, H. Fahim, and B. He, "Communication and control in collaborative uavs: Recent advances and future trends," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 122–12 139, 2022.
- [3] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "Cyber security methods for aerial vehicle networks: taxonomy, challenges and solution," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 4928–4944, 2018.
- [4] M. E. Newman, *Spectral methods for community detection and graph partitioning*. American Physical Society, 2013, vol. 88, no. 4.
- [5] F. R. Chung, *Spectral graph theory*. Providence, RI: American Mathematical Society, 1997, vol. 92.
- [6] P. Holme, "Network analysis of particle trajectories," *Physical Review E*, vol. 94, no. 2, p. 022310, 2016.
- [7] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 855–864.

- [8] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," in *Journal of Cryptographic Engineering*, vol. 2, no. 2. Springer, 2012, pp. 77–89.
- [9] S. Josefsson and I. Liusvaara, "Edwards-curve digital signature algorithm (eddsa)," RFC 8032, 2017.
- [10] M. Hutter and P. Schwabe, "Fast and compact elliptic-curve cryptography," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 1–17.
- [11] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, "Swarms of unmanned aerial vehicles—a survey," *Journal of Industrial Information Integration*, vol. 16, p. 100106, 2019.
- [12] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.
- [13] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.
- [14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [15] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrđić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 387–402.
- [16] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using aoa," in *Proceedings of IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3. IEEE, 2003, pp. 1734–1743.
- [17] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.
- [18] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.
- [19] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI)*, New Orleans, Louisiana, USA, 1999, pp. 173–186.
- [20] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner Jr, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," *Radionavigation Laboratory Conference Proceedings*, 2008.
- [21] D. P. Shepard, T. E. Humphreys, and A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to gps spoofing attacks," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3–4, pp. 146–153, 2012.
- [22] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994, pp. 124–134.
- [23] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," in *IACR Transactions on Cryptographic Hardware and Embedded Systems*. IACR, 2018, pp. 238–268.
- [24] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon: Fast-fourier lattice-based compact signatures over ntru," in *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, 2017.
- [25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.
- [26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

## APPENDIX

All experiments reported in this paper are fully reproducible. The software environment consists of Python 3.10.16 with NetworkX 3.2.1, NumPy 1.26.3, scikit-learn 1.4.0, and py-cryptodome 3.20.0. Tests were conducted on Linux 6.17.8 (Arch Linux) running on AMD64 architecture. Reproducibility



is ensured through fixed random seed (seed=42) for both Python random and NumPy random generators. Complete source code and experiment scripts are available in the Argus repository.

#### Reproduction Instructions:

```
git clone https://github.com/Sang-Buster/Argus
cd Argus
uv sync # Install dependencies
uv run argus --attack phantom \
    --detectors all --mode comparison \
    --num-uavs 30
```

Expected output: Crypto detector shows TPR=1.000, FPR=0.000, F1=1.000

Table IV presents the complete performance matrix for all detector-attack combinations, confirming cryptographic authentication’s consistent superiority with perfect detection (TPR=1.0, FPR=0.0) for phantom and coordinated attacks. Spectral detection matches this performance for topology-altering attacks but fails on position falsification due to its topology-preserving nature. Centrality and ML detectors exhibit prohibitively high false positive rates (87-100%), rendering them impractical for deployment.

Table V quantifies mobility impact at 10 m/s velocities, revealing that cryptographic authentication maintains zero FPR regardless of movement while spectral detection degrades dramatically (0% to 62.5% FPR for phantom attacks). This differential stems from fundamental architectural differences—signature verification operates independently of topology dynamics, whereas spectral methods cannot distinguish benign topology changes from attacks.

Table VI demonstrates cryptographic detection’s linear scalability  $O(n)$ , with detection time increasing proportionally from 39.12ms (20 UAVs) to 57.90ms (30 UAVs) while maintaining perfect accuracy. This predictable scaling contrasts favorably with graph-based methods where performance depends on swarm size, network density, and connectivity patterns.

Table VII illustrates spectral detection’s extreme threshold sensitivity. While  $\tau^* = 2.5$  achieves perfect detection, minor deviations cause catastrophic degradation:  $\tau = 2.0$  introduces 26.7% false positives,  $\tau = 3.0$  reduces TPR to 66.7%, and  $\tau = 4.0$  eliminates all detections. This brittleness necessitates environment-specific calibration and enables threshold-evading attacks, reinforcing that cryptographic methods provide superior operational reliability.

#### A. Code Metrics

The Argus framework comprises approximately 3,500 lines of production code organized into five modular subsystems. Table VIII details the codebase structure, where the detection module (1,200 lines) reflects the complexity of implementing four algorithms, while the cryptographic module remains compact (150 lines) by leveraging established primitives. The framework achieves 85% test coverage with emphasis on cryptographic operations and attack injection.

#### B. Performance Characteristics

The framework operates efficiently within research constraints. Memory consumption for 30 UAVs totals approximately 15 MB (swarm state), 2 MB per timestep (graphs), and 60 MB (baseline samples), with 200 MB peak including visualization. Execution time for a complete run (30 UAVs, 50 timesteps) is 2.5s (spectral), 6.0s (crypto), 3.5s (ML), and 15s (full comparison). Algorithmic complexity follows theoretical bounds:  $O(n)$  for cryptographic verification,  $O(n^2)$  for spectral eigendecomposition, and  $O(n^3)$  for betweenness centrality.

TABLE IV  
COMPLETE PERFORMANCE MATRIX (30 UAVs, STATIONARY)

Attack Type	Detector	TPR	FPR	Precision	Recall	F1 Score	Time (ms)
Phantom	Spectral	1.000	0.000	1.000	1.000	1.000	1.97
	Centrality	0.667	1.000	0.062	0.667	0.114	1.24
	<b>Crypto</b>	<b>1.000</b>	<b>0.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	57.90
	ML	0.333	0.933	0.034	0.333	0.062	4.88
Position	Spectral	0.000	0.000	0.000	0.000	0.000	1.75
	Centrality	1.000	1.000	0.100	1.000	0.182	1.12
	<b>Crypto</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	57.27
	ML	0.667	0.963	0.071	0.667	0.129	4.63
Coordinated	Spectral	1.000	0.000	1.000	1.000	1.000	1.90
	Centrality	1.000	1.000	0.091	1.000	0.167	1.51
	<b>Crypto</b>	<b>1.000</b>	<b>0.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	57.21
	ML	1.000	0.867	0.103	1.000	0.188	5.11

TABLE V  
MOBILITY IMPACT SUMMARY (30 UAVs)

Attack	Detector	Stat. TPR	Stat. FPR	Mobile TPR	Mobile FPR	FPR Change
Phantom	Spectral	100.0%	0.0%	99.2%	62.5%	<b>+62.5%</b>
	<b>Crypto</b>	<b>100.0%</b>	<b>0.0%</b>	<b>100.0%</b>	<b>0.0%</b>	<b>0.0%</b>
Position	Spectral	0.0%	0.0%	1.9%	3.2%	+3.2%
	<b>Crypto</b>	<b>0.0%</b>	<b>0.0%</b>	<b>0.0%</b>	<b>0.0%</b>	<b>0.0%</b>
Coordinated	Spectral	98.5%	0.0%	99.0%	6.7%	<b>+6.7%</b>
	<b>Crypto</b>	<b>100.0%</b>	<b>0.0%</b>	<b>100.0%</b>	<b>0.0%</b>	<b>0.0%</b>

TABLE VI  
SWARM SIZE IMPACT (PHANTOM ATTACK, CRYPTO DETECTOR)

UAVs ( $n$ )	TPR	FPR	F1	Time (ms)
20	1.000	0.000	1.000	39.12
25	1.000	0.000	1.000	47.87
30	1.000	0.000	1.000	57.90

TABLE VII  
SPECTRAL THRESHOLD SENSITIVITY (PHANTOM ATTACK, 30 UAVs)

Threshold ( $\tau$ )	TPR	FPR	F1
2.0	1.000	0.267	0.600
2.5	1.000	0.000	1.000
3.0	0.667	0.000	0.800
4.0	0.000	0.000	0.000

TABLE VIII  
MODULE BREAKDOWN

Module	Purpose	LOC
core/	UAV physics, swarm simulation, Remote ID protocol	~800
crypto/	Ed25519 signing and verification	~150
detection/	All 4 detection method implementations	~1,200
attacks/	Attack injection and management	~600
evaluation/	Metrics computation, visualization	~400
cli_main.py	Interactive CLI and experiment orchestration	~1,550