



ARGUS

UAV REMOTE ID SPOOFING DEFENSE SYSTEM

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

TABLE OF CONTENT

Introduction

Methodology

Simulation

Conclusion

An abstract geometric design featuring two thin, dark grey lines that intersect on a light grey background. One line is oriented diagonally from the top-left towards the bottom-right, while the other is oriented from the top-right towards the bottom-left. The intersection point is located in the upper-left quadrant of the image.

INTRODUCTION

ARGUS

A many-eyed giant in Greek mythology. Known for his perpetual vigilance, he served the goddess Hera as a watchman. His most famous task was guarding Io, a priestess of Hera, whom Zeus had transformed into a heifer.



OVERVIEW

Research Question:

How can we detect and defend against Remote ID spoofing attacks in UAV swarms?

Approach:

1. Graph-theoretic analysis
2. Cryptographic verification

Novelty:

First systematic evaluation comparing cryptographic and graph-based detection methods for UAV spoofing attacks.

PROBLEM STATEMENT

UAV Remote ID Mandate

- **FAA 14 CFR Part 89:** All UAVs must broadcast Remote ID
- **Purpose:** Identify drones in airspace for safety and security
- **Vulnerability:** Broadcast protocol is unauthenticated

Security Threats

- **Phantom UAV Injection** - Broadcast fake drone identities
- **Position Falsification** - Report false GPS coordinates
- **Coordinated Attacks** - Multiple attackers working together

Impact: Disrupts air traffic control, misleads authorities, enables malicious operations

RESEARCH OBJECTIVES

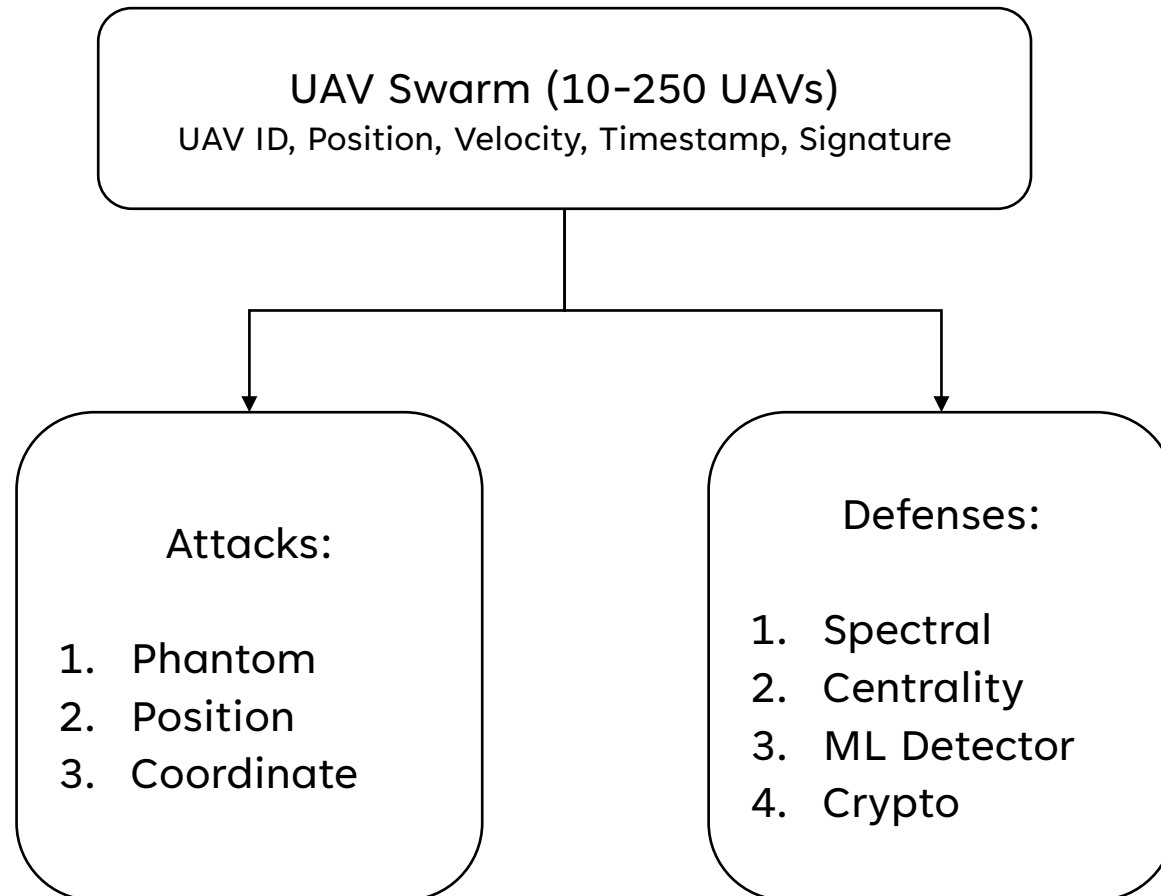
Primary Goals

- *Model* UAV swarms as dynamic graphs
- *Implement* realistic attack scenarios
- *Develop* detection methods using graph theory
- *Evaluate* cryptographic defenses (Ed25519)
- *Quantify* performance trade-offs

Success Criteria

- Detect attacks with high TPR, low FPR
- Real-time performance (<100ms per detection)
- Scalable to 50+ UAV swarms
- Publication-quality results and visualizations

SYSTEM ARCHITECTURE



An abstract graphic featuring two thin, dark grey lines that intersect on a light grey background. One line runs diagonally from the top-left towards the bottom-right, while the other runs from the top-right towards the bottom-left. The word "METHODOLOGY" is positioned to the right of the intersection point, in a bold, black, sans-serif font.

METHODOLOGY

GRAPH REPRESENTATION

UAVs as Nodes:

- Each UAV = vertex in graph
- Node attributes: Remote ID, position, velocity, timestamp, signature

Communication as Edges:

- Edge exists if distance \leq communication range
- Typical range: 100-200 meters
- Graph updates every timestep (1 Hz)

Graph Properties

- **Connectivity:** Single connected component (legitimate swarm)
- **Density:** Depends on UAV spacing
- **Dynamics:** Topology changes as UAVs move

ATTACK IMPLEMENTATION

1. Phantom UAV Injection

Method: Add fake nodes to the graph

- Inject 5-10 phantom UAVs at random positions
- Phantoms broadcast Remote ID messages
- No cryptographic keys (cannot sign)

Detection Challenge: Phantoms can mimic movement patterns

ATTACK IMPLEMENTATION

2. Position Falsification

Method: Legitimate UAVs report false GPS coordinates

- Select 10-20% of UAVs to compromise
- Add random offset (50-100m) to reported position
- True position used for movement (topology preserved)

Detection Challenge: Graph structure unchanged

ATTACK IMPLEMENTATION

3. Coordinated Attack

Method: Multiple phantoms in formation

- Create 8+ phantoms moving together
- Patterns: circle, line, or coordinated formation
- Synchronized velocities

Detection Challenge: Appears like legitimate sub-swarm

DETECTION METHOD

1. Spectral Analysis

Theory

- **Laplacian Matrix:** $L = D - A$ where D = degree matrix (diagonal), A = adjacency matrix
- **Eigenvalues:** $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$
 - λ_2 = algebraic connectivity
 - Distribution reveals structural properties

Detection Algorithm

- **Baseline:** Compute mean/std of eigenvalues from clean graphs
- **Monitor:** Track eigenvalue distribution over time
- **Detect:** Flag anomalies when Z-score > threshold (2.5-2.8)

DETECTION METHOD

2. Centrality Analysis

Centrality Metrics

- **Degree Centrality:** Number of connections: $C_D(v) = \frac{\deg(v)}{(n-1)}$
- **Betweenness Centrality:** Bridge position: $C_B(v) = \sum \left(\frac{\sigma_{st}(v)}{\sigma_{st}} \right)$
- **Closeness Centrality:** Average distance to others: $C_C(v) = \frac{1}{\sum d(v,u)}$

Detection Algorithm

Anomaly Score = $0.4 \times \text{degree}_z + 0.4 \times \text{betweenness}_z + 0.2 \times \text{closeness}_z$,

flag if score > 6.0

DETECTION METHOD

3. ML Detector

Architecture

- **Ensemble:** Isolation Forest
- **Features:** 4 graph metrics (degree, betweenness, clustering, closeness.)
- **Training:** Contamination parameter $\alpha = 0.15$ (expects 15% anomalies)

Detection Algorithm

- Extract 4-feature vector per node from baseline graphs
- Train Isolation Forest on normal (clean) swarm behavior
- Compute anomaly scores for test graph nodes
- Flag nodes with scores below threshold

DETECTION METHOD

4. Cryptography

Ed25519 Digital Signatures

- **Fast:** ~50 μ s signing, ~100 μ s verification
- **Secure:** 256-bit security level
- **Small:** 32-byte keys, 64-byte signatures
- **Deterministic:** No nonce reuse vulnerability

Implementation

Signing (legitimate UAVs):

```
signature = private_key.sign(message_bytes)  
remote_id.signature = signature # 64 bytes
```

Verification (receivers):

```
public_key.verify(message_bytes, signature) # Raises exception if invalid
```

An abstract graphic on a light gray background. Two thin, dark gray lines intersect. One line is oriented diagonally from the top-left towards the bottom-right. The other line is oriented diagonally from the top-right towards the bottom-left. The intersection point is located in the upper-left quadrant of the image. To the right of the intersection, the word "SIMULATION" is written in a bold, black, sans-serif, all-caps font.

SIMULATION

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
	Crypto	1.000	0.000	1.000	1.000	1.000	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
	Crypto	0.000	0.000	0.000	0.000	0.000	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
	Crypto	1.000	0.000	1.000	1.000	1.000	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%	+62.5%
	Crypto	100.0%	0.0%	100.0%	0.0%	0.0%
Position	Spectral	0.0%	0.0%	1.9%	3.2%	+3.2%
	Crypto	0.0%	0.0%	0.0%	0.0%	0.0%
Coordinated	Spectral	98.5%	0.0%	99.0%	6.7%	+6.7%
	Crypto	100.0%	0.0%	100.0%	0.0%	0.0%

		POSITIVE	NEGATIVE
ACTUAL VALUES	POSITIVE	TP	FN
	NEGATIVE	FP	TN

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

An abstract graphic featuring two thin, dark grey lines that intersect on a light grey background. One line is oriented diagonally from the top-left towards the bottom-right, while the other is more horizontal, sloping slightly downwards from left to right. The intersection point is located in the upper-left quadrant of the image. To the right of this intersection, the word "CONCLUSION" is written in a bold, black, sans-serif typeface.

CONCLUSION

LIMITATIONS

Simplified UAV Model:

- Constant velocity (no acceleration)
- No wind/turbulence effects
- Straight-line movement

Attack Models:

- Basic phantom injection
- No sophisticated evasion tactics
- Static attack parameters

Network Model:

- Perfect communication within range
- No packet loss or latency
- Simplified Remote ID format

CHALLENGES

Position Falsification:

- All tested methods achieve 0% TPR ($F1=0.0$)
- Topology-preserving attacks are fundamentally undetectable
- Requires alternative approaches (multi-lateration, Byzantine consensus)

Graph Heuristics (ML, Centrality):

- Unacceptable FPR (87-100%) for production
- Configuration-sensitive and mobility-degraded
- Probabilistic detection subject to evasion

Cryptographic Authentication:

- Requires PKI infrastructure
- Cannot detect compromised UAVs with valid keys signing false data
- 30× computational overhead vs spectral (but acceptable)

FUTURE WORK

Realistic Flight Dynamics:

- Add acceleration/deceleration
- Implement waypoint navigation
- Model formation flying
- Environmental effects (wind, turbulence)

Communication-Level Features:

- Message timing analysis
- Protocol compliance checking
- Latency/jitter patterns
- Signature validation timing

FUTURE WORK

Position Verification:

- Multi-lateration using signal strength and time-of-arrival
- Physics-based validation (impossible velocities/accelerations)
- Byzantine consensus protocols (fault-tolerant position verification)

Mobility-Aware Thresholds:

- Adaptive threshold functions: $\tau(v_{swarm}) = \tau_0 + k \cdot \bar{v}$
- Improve heuristic detector robustness in dynamic scenarios
- Reduce mobility-induced false positives

CONCLUSION

Cryptographic authentication is **non-negotiable** for security-critical systems

Spectral methods **effective but insufficient** alone

Graph heuristics have **fundamental limitations**



THANK YOU

Sang Xing

xings@my.erau.edu

github.com/Sang-Buster/Argus