Solving the Identity-Privacy Paradox

How to Identify Friend or Foe Without Self-Disclosure?

Ema Šujster

Delft University of Technology

TU Delft

Solving the Identity-Privacy Paradox

How to Identify Friend or Foe Without Self-Disclosure?

by

Ema Šujster

Ema
Šujster

Supervisors:	L. Markatou, G. Smaragdakis
External Supervisor:	K. Wrona
Project Duration:	November, 2025 - July, 2026
Faculty:	Faculty of Electrical Engineering, Mathematics and Computer Science, Delft

Cover:	A Swarm of Drones and Guided Munitions: The Ministry of Strategic Industry organised demonstration tests of Ukrainian solutions (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

I would like to thank my friends for their support and encouragement throughout this process. Their understanding, patience, and confidence in me have meant a great deal. They have listened to more conversations about this thesis than anyone should reasonably have had to, celebrated the small victories along the way, and helped me keep things in perspective whenever I lost sight of it. Their presence made this process much lighter than it otherwise would have been.

I want to thank Lilika and George, my TU Delft supervisors, for allowing me the freedom to explore my own ideas and encouraging independence. I appreciate the opportunity to pursue a topic that closely aligned with my interests and ambitions. The space to develop the project in my own way with your guidance has been invaluable for this research, my own development, and my future plans.

I also want to thank my NATO Communications and Information Agency (NCIA) supervisor, Konrad, for agreeing to undertake this project with me and for sharing insights from his work in the field. His expertise provided an important perspective throughout the research, and I have learnt a lot from our discussions. I particularly appreciated that my assumptions were constantly challenged and encouraged me to think more critically about the problem and the claims made in this work.

Ema Šujster
Delft, July 2026

Abstract

Reliable identification of cooperative UAVs in contested environments remains challenging because infrastructure connectivity cannot be assumed, while persistent identifiers expose friendly assets to passive tracking. Existing Identification Friend or Foe (IFF) systems rely on centralised certificate infrastructures and stable communication links, motivating a privacy-preserving alternative that operates autonomously at the tactical edge. This thesis presents the IFF Relation (R_{IFF}), a non-interactive zero-knowledge attestation protocol in which a prover demonstrates federation membership, possession of an enrolled hardware secret, and valid mission credentials without revealing any underlying attributes, using a single Groth16 proof. The protocol jointly constrains membership and hardware attestation within one zero-knowledge relation, preventing cross-asset credential composition, replay attacks through nonce binding, and attribute disclosure, while enabling independent re-verification by multiple observers. The relation is defined over collision-resistant hash functions and Merkle paths, making it migratable to post-quantum backends, although the current Groth16/BN128 instantiation is not post-quantum secure. A proof-of-concept implementation evaluated on x86-64 and ARM hardware under controlled network impairment and adversarial scenarios completes authentication in under $2.2\,\mathrm{s}$ on x86-64 but requires 25–$28\,\mathrm{s}$ on ARM, constraining feasible encounter distances to over 350 m at typical closure speeds. The results demonstrate that privacy-preserving control-state attestation is achievable on capable platforms, while relay attacks remain unmitigated, enabling an adversary without credentials to be classified as friendly by relaying a legitimate proof. Closing this vulnerability requires distance-bounding or channel-binding mechanisms beyond the current design.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AO	Approaching Object
CI	Confidence Interval
CRS	Common Reference String
CRL	Certificate Revocation List
CRP	Challenge Response Pair
DVFS	Dynamic Voltage and Frequency Scaling
DID	Decentralized Identifier
DoS	Denial of Service
ECC	Elliptic-Curve Cryptography
FANET	Flying Ad Hoc Network
FMN	Federated Mission Networking
GCS	Ground Control Station
GMP	GNU Multiple Precision Arithmetic Library
HMAC	Hash-based Message Authentication Code
HSM	Hardware Security Module
ICS	Industrial Control Systems
IFF	Identification Friend or Foe
IoD	Internet of Drones
IoT	Internet of Things
IQR	Interquartile Range
ISR	Intelligence, Surveillance, and Reconnaissance
KEM	Key Encapsulation Mechanism
MAC	Message Authentication Code
MAVLink	Micro Air Vehicle Link (autopilot data-link protocol)
MEC	Multi-access Edge Computing
MDS	Maximum distance separable (Poseidon property)
MPC	Multi-Party Computation
NIST	National Institute of Standards and Technology
NIZK	Non-Interactive Zero-Knowledge
OD	Observing Drone
OCSP	Online Certificate Status Protocol
PKI	Public Key Infrastructure
PoC	Proof of Concept
PPT	Probabilistic Polynomial-Time
PQC	Post-Quantum Cryptography
PUF	Physical Unclonable Function
QAP	Quadratic Arithmetic Program
R1CS	Rank-1 Constraint System
RF	Radio Frequency
RID	Remote Identification
ROR	Random Oracle Model
SMT	Sparse Merkle Tree
SNARK	Succinct Non-Interactive Argument of Knowledge
STARK	Scalable Transparent Argument of Knowledge
TPM	Trusted Platform Module

Abbreviation	Definition
UAV	Unmanned Aerial Vehicle
WSL2	Windows Subsystem for Linux 2
ZK	Zero-Knowledge
zkSNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge
ZMQ	ZeroMQ (asynchronous messaging library)

Mathematical Notation

Symbol	Definition
R_{IFF}	IFF attestation relation proven in zero knowledge
ϕ	Public input vector $(\phi_{\mathsf{reg}}, \phi_{\mathsf{nonce}}, \phi_{\mathsf{epoch}}, \phi_{\mathsf{hw}}, \phi_{\mathsf{pol}})$
ϕ_{reg}	Federation registry Merkle root (public input)
ϕ_{nonce}	Session nonce generated by the OD per encounter
ϕ_{epoch}	Active policy epoch identifier (public input)
ϕ_{hw}	Session-bound hardware commitment $H(H(k_{\mathsf{hw}}) \,\|\|\, \phi_{\mathsf{nonce}})$
ϕ_{pol}	Policy Merkle root (public input)
w	Private witness vector $(w_{\mathsf{reg}}, w_{\mathsf{hw}}, w_{\mathsf{pol}})$
π	Groth16 NIZK proof (137 bytes over BN128)
pk	Proving key (CRS component used by the prover)
vk	Verification key (CRS component used by the verifier)
$H(\cdot)$	Collision-resistant hash function (Poseidon over BN128 scalar field)
k_{hw}	Live hardware-derived key, produced in volatile memory at attestation time
$h_{\mathsf{hw}}^{\mathsf{enrol}}$	Enrolled hardware reference value $H(k_{\mathsf{hw}}^{\mathsf{enrol}})$, stored in the OD registry
ℓ	Registry leaf $H(H(ID_{\mathsf{priv}}) \,\|\|\, h_{\mathsf{hw}}^{\mathsf{enrol}})$ committed at enrolment
λ	Security parameter
$\mu(\lambda)$	Negligible function of the security parameter
D_{crit}	Critical separation distance: minimum starting separation for one complete authentication cycle before encounter
T_{cycle}	Theoretical end-to-end cycle time used in the D_{crit} derivation
v	Platform closure speed (m/s)
N_{attempts}	Number of complete authentication cycles available before encounter, $\lfloor (D/v)/T_{\mathsf{cycle}} \rfloor$
d	Injected one-way RF link delay (ms)
l	Per-message packet loss probability
n	Sample size (number of cycles per experimental condition)
σ	Standard deviation of a timing measurement
$\mathcal{E}$	PPT knowledge extractor (used in Theorem 3.1)
$\|\|$	Concatenation of bit strings
L_{e2e}	Corresponding measured end-to-end latency (empirical)
L_{base}	Measured end-to-end latency at zero injected delay ($d = 0$)

1

Introduction

Modern airspace has become a contested domain. Low-cost unmanned aerial vehicles (UAVs) are increasingly used in hybrid operations, and the authentication frameworks that support current air defence architectures have not kept pace. Traditional Identification Friend or Foe (IFF) systems rely on centralised infrastructure and stable communication links that contested environments cannot guarantee. When a UAV operating at speed encounters an unidentified airframe, it must decide locally, autonomously, and without disclosing sensitive identity or mission information whether to treat it as a coalition asset or a threat. No deployed system currently enables this combination of properties.

This thesis addresses that gap by proposing a privacy-preserving authentication framework based on non-interactive zero-knowledge proofs that enables cooperative UAVs to verify each other's authorisation and hardware binding locally, without revealing their identities or mission contexts to the verifier or passive channel observers.

1.1. Operational Context and Motivation

The operational need for local UAV authentication has become concrete in recent years. Along NATO's eastern flank, low-cost aerial incursions have highlighted a fundamental cost asymmetry: a single inexpensive UAV can force the deployment of costly interceptors, making large-scale aerial harassment economically sustainable for adversaries [65]. In the Baltic region, UAVs and other aerial objects have repeatedly breached sovereign airspace, forcing airport closures and disrupting civilian infrastructure [76]. In response, multiple nations have begun coordinating heterogeneous sensor and interceptor networks spanning from the Baltic to the Black Sea [26]. Because these networks are composed of hardware and software from different national vendors, assets from different nations must be able to verify one another locally and instantly, without relying on a single central authority [26].

This challenge is reflected in recent European policy. In February 2026, the European Commission introduced an EU Action Plan on Drone and Counter-Drone Security, calling for improved identification frameworks, interoperable command-and-control capabilities, and coordinated response mechanisms across heterogeneous platforms [21, 22]. Regulatory identification improves accountability but does not address real-time trust establishment among heterogeneous operational actors in contested or degraded communication environments. That is the gap this thesis targets: enabling rapid, local discrimination between authorised and unverified UAVs without relying on central databases or disclosing sensitive mission metadata [35].

1.2. Problem Statement

Contemporary air defence systems lack a reliable mechanism to establish trust between cooperative UAVs under contested conditions, for two reasons.

The first is infrastructure dependence. Current authentication frameworks assume reliable connectivity to central infrastructure, online credential verification, and a known set of participants. In contested

environments, none of these assumptions can be guaranteed. Communication links may be delayed, degraded, or denied, and UAVs must be capable of establishing trust locally, without continuous connectivity to a ground control station.

The second, and more fundamental, problem concerns the semantics of identity-based authentication. Existing approaches bind trust to possession of a valid credential establishing federation membership. Such credentials can be extracted from captured platforms, replayed, or reused by an adversary. More critically, identity alone does not encode whether a UAV is currently authorised to operate within a specific mission context, geographic sector, or temporal window. A legitimate federation member may simultaneously be operating outside its assigned role or under adversarial control.

This thesis thus adopts a stronger notion of trust based on *control state*: a composite property that establishes simultaneously that a UAV is a registered and non-revoked federation member, possesses a hardware-derived secret consistent with its enrolled reference, and holds a valid authorisation for the active mission context. The core problem is to enable cooperative UAVs to verify the control state of other assets locally, without continuous connectivity, without disclosing identity or mission information, and within the computational constraints of resource-limited hardware.

This problem introduces an inherent tension between verifiability and confidentiality. In a multi-national coalition, the verifying party learns neither the identity nor mission context of the asset being verified, yet the verification outcome must be strong enough to support operational decision-making. Non-interactive zero-knowledge proofs offer a cryptographic foundation for resolving this tension, and this thesis examines whether they can do so practically under the constraints of contested environments and resource-constrained UAV platforms.

Within this work, a *friend* is a UAV that can provide cryptographic evidence of a valid control state. A *foe* is any UAV that cannot establish such a state. This is a provisional protocol-level classification, not an engagement decision, and all consequential actions remain subject to human authority as described in Section 1.4. The false-foe risk under degraded link conditions is addressed in Section 5.3.

1.2.1. System Requirements

The following requirements follow directly from the problem identified above and the operational constraints of Section 1.1.

Functional Requirements

FR-1: Local Trust Establishment The protocol must enable UAVs to establish trust autonomously and locally, without requiring connectivity to a Ground Control Station or centralised authority.

FR-2: Control State Verification The protocol must verify composite control state, federation membership, possession of an enrolment-bound hardware secret, and current policy authorisation, rather than static identity alone.

FR-3: Revocation Support The protocol must support dynamic revocation such that revoked assets cannot produce accepting proofs after removal from the federation registry.

FR-4: Multi-Observer Forwarding The protocol must produce non-interactive proofs that can be forwarded between observers and independently re-verified without recomputation by the original prover.

FR-5: Standardisation Compatibility The protocol must be deployable over existing UAV communication standards, including MAVLink.

Security Requirements

SR-1: Replay Resistance Each proof must be bound to a session-specific verifier-supplied nonce, preventing reuse across sessions at the issuing observer.

SR-2: Unlinkability The protocol must not reveal persistent identifiers or linkable attributes across sessions.

SR-3: Soundness Under Impersonation A computationally bounded adversary must not be able to produce an accepting proof without simultaneously satisfying all three trust layers, even in possession of partial credential material.

SR-4: Hardware Binding Authentication must be bound to a device-specific hardware secret that cannot be transferred between platforms.

SR-5: Context Binding Proof validity must be restricted to the intended policy epoch, sector, and role context.

SR-6: Post-Quantum Readiness The zero-knowledge *relation* must be defined using only collision-resistant hash functions and Merkle path verifications, with no pairing-based assumptions, so that it can be instantiated with a post-quantum proof backend without modification to the relation itself.

SR-7: Zero-Knowledge Privacy The proof must reveal nothing about the private witnesses beyond the fact that the relation holds.

Operational Requirements

OR-1: Verification Latency Proof verification must impose negligible latency relative to proof generation. Absolute proof generation latency must additionally satisfy OR-3 to be operationally useful.

OR-2: Communication Overhead The proof bundle must be compact enough for constrained UAV datalinks.

OR-3: Resource Compatibility The protocol must be executable on resource-constrained UAV platforms, with proof generation completing within the available encounter window for the target hardware class.

OR-4: Infrastructure Independence The protocol must remain operational when connectivity to ground infrastructure is intermittent or unavailable.

OR-5: Degraded Connectivity Tolerance The protocol must produce usable trust decisions under elevated delay and packet loss. Authentication timeouts under degraded conditions are an operational constraint rather than a protocol failure mode.

OR-6: Heterogeneous Interoperability The protocol must support trust establishment between assets from different national authorities operating under a shared federation root provisioned during pre-mission setup.

1.3. Contribution

In response to the problem and requirements above, this thesis makes four contributions.

A joint-circuit zero-knowledge attestation relation. R_{IFF} (the IFF Relation) encodes federation membership, possession of an enrolment-bound hardware secret, and policy authorisation as a single Groth16 proof. The joint circuit enforces a shared wire between the membership and hardware statements, preventing an adversary from composing a valid proof using credential material from two separately compromised assets, a threat specific to coalition environments where credentials are distributed across multiple national custodians.

A forwarding-capable multi-observer verification model. The non-interactive design allows any observer holding locally provisioned anchor values to re-verify a forwarded proof without contacting the original prover, distributing trust decisions across a swarm without runtime infrastructure connectivity.

A cross-platform empirical evaluation. The protocol is implemented and evaluated on x86-64 and embedded ARM hardware under controlled network impairment and adversarial scenarios, including a constructive test validating the asset binding theorem empirically.

An analytical deployability framework. The relationship $D_{\mathsf{crit}} = v \cdot T_{\mathsf{cycle}}$ translates measured proving latency into a minimum required standoff distance for a given closure speed, giving future optimisation work a concrete operational target.

1.4. Ethical Considerations

This work combines cryptographic protocol design with autonomous defence systems, raising ethical responsibilities beyond those in civilian security research.

The cryptographic primitives developed here, zero-knowledge proofs, hardware-bound attestation, and privacy-preserving credential verification, have direct civilian applications [56], but in a defence context

they may also obscure the accountability of autonomous systems operating beyond sanctioned boundaries. This dual-use tension [57] is not resolvable through technical design alone. The protocol includes accountability mechanisms in that the trusted authority can revoke any asset and the federation registry is auditable, but the governance question of who controls the trusted authority falls outside this work and requires policy-level resolution.

The protocol is designed for autonomous operation without continuous ground control station connectivity (FR-1, OR-4). The claim that responsibility remains with human operators should therefore be understood at the level of governance and rules of engagement established before deployment, not real-time oversight of individual encounters. The protocol produces only a binary trust classification and does not constitute an engagement or kinetic decision; kinetic decision processes are explicitly excluded from scope (Section 2.4).

The privacy guarantees that protect coalition assets from adversarial tracking also limit surveillance capacity. Internal accountability is preserved through revocation and auditable enrolment rather than identity disclosure during authentication. This work does not claim to resolve the tension between privacy and accountability fully [58], and the protocol is presented as a research prototype requiring formal verification, independent security evaluation, and regulatory approval before any operational deployment [8].

1.5. Thesis Outline

Chapter 2 surveys the state of the art in UAV authentication, zero-knowledge attestation, and post-quantum cryptography, establishes the cryptographic and operational foundations required for the protocol, states the research questions, defines the threat model, and describes the research methodology.

Chapter 3 defines the R_{IFF} relation, building on the threat model of Chapter 2, and specifies the multi-observer verification properties that the design must satisfy.

Chapter 4 describes the proof-of-concept implementation and empirical evaluation, covering baseline performance, network impairment sensitivity, adversarial security testing, and a cross-platform comparison of x86-64 and ARM results.

Chapter 5 interprets the results against the three sub-research questions, develops the deployability assessment, identifies implications for practice, and states the limitations of the work.

Chapter 6 summarises contributions and findings, outlines future work, and discusses broader impact.

The thesis closes with Chapter 7, which reflects on the research process, the implications of the work for the broader Computer Science field, and the use of Generative AI tools in accordance with TU Delft guidelines.

2

Research Framework

The operational context established in Chapter 1 makes clear that the establishment of autonomous UAV trust in contested environments remains an unsolved research problem. The convergence of low-cost aerial incursions, heterogeneous coalition interoperability demands, and the approach to post-quantum transition collectively defines a design space that existing solutions do not fully cover.

This chapter constructs the research foundation from which a solution is derived. Section 2.1 surveys the state of the art across drone authentication, hardware-rooted identity, zero-knowledge attestation, and post-quantum cryptography integration. Although this literature demonstrates progress in each of these areas individually, the review reveals that no existing approach simultaneously meets the functional, security, and operational requirements identified in Section 1.2.1. Specifically, no current solution combines decentralised operation, post-quantum resistance, privacy-preserving authorisation, and feasibility on resource-constrained UAV platforms in infrastructure-denied environments.

Section 2.2 introduces the foundational concepts required to follow the technical contributions of this work, covering UAV platform constraints, the threat characteristics of contested airspace, and the cryptographic primitives enabling the proposed solution.

With this foundation established, Section 2.3 states the primary research question and its three sub-questions, which together define the scope and measurable objectives of the thesis. The threat model in Section 2.4 then formally defines the adversary, the system context, and the attack surface against which the protocol must hold. Finally, Section 2.5 outlines the research methodology and explains how the experimental design, implementation, and evaluation strategy are aligned to answer the stated questions within the constraints of the defined threat model (Section 2.4).

2.1. Related Work

The increasing frequency of low-cost drone incursions and hybrid aerial threats has revealed significant shortcomings in current real-time identification and authentication systems. While solutions like Remote ID and centralised registration help with tracking, they do not address the need for local decentralised verification in complex, multi-national environments. This section looks at existing research on drone authentication, including hardware-based methods, zero-knowledge proofs, and post-quantum cryptography. The goal is to understand what works, what does not, and where there are still gaps, especially for lightweight, post-quantum, privacy-focused systems that can operate independently in contested areas.

2.1.1. Identification and Authentication in the Internet of Drones

Early work on the Internet of Drones (IoD) treats identification and authentication largely as a direct extension of conventional wireless and IoT security. Drones are enrolled as networked ”things” and rely on software- and/or hardware-based cryptography to prove their identity to ground infrastructure and peers. Michailidis and Vouyioukas [46] provide a comprehensive survey of IoD authentication mechanisms, organising them into hash-based schemes, Public Key Infrastructure (PKI) with RSA/ECC

(elliptic-curve cryptography), and more recent machine-learning and blockchain approaches. They highlight the inherent limitations of UAVs in terms of energy, computational power, and memory, which make the design of efficient and lightweight authentication solutions challenging. The absence of unified security standards and the heterogeneous nature of IoD networks further complicate the application of conventional cryptographic algorithms in this domain.

These resource constraints are compounded by a broad attack surface. Drones are seen as complex systems with various sensors, making them similar to Internet of Things (IoT) devices, and the increasing number of drones amplifies the risk of hacking and collisions [54]. They are vulnerable to a wide range of attacks, including eavesdropping, man-in-the-middle, spoofing, tampering, Denial-of-Service (DoS), impersonation, replay, and forgery attacks, all of which target sensitive data or cryptographic keys [46]. System-level surveys emphasise that authentication is the first line of defence for protecting telemetry, command links, and airspace usage. Failure to authenticate UAVs correctly can lead to safety risks such as hijacking and mid-air collisions in congested airspace [81]. Roodsari et al. [55] underscore the lack of standardisation and unification in drone communication, which exacerbates security and privacy challenges.

This authentication challenge is particularly acute in air traffic management and Intelligence, Surveillance, and Reconnaissance (ISR) environments, where any entity injecting control or surveillance data into the system must be correctly identified and authenticated. Agarwal et al. [1] argue that ground control stations, air traffic controllers, and UAVs all need mutually authenticated channels before control commands or tracking data can be trusted. They reviewed ECC, hyperelliptic-curve, and blockchain-assisted schemes as candidate mechanisms for securing bandwidth-constrained aviation links. Ramos et al. [54] highlight authentication and non-repudiation as critical properties for air-ground and UAV-UAV control links, with PKI, multi-factor authentication, and message authentication codes (MACs) as baseline requirements. They also noted that authorisation and auditing are crucial capabilities for ISR missions. The goal of secure communication in anti-drone systems is to protect the integrity and confidentiality of data between sensors, control units, and neutralisation mechanisms [19].

To organise the range of proposed solutions, Michailidis and Vouyioukas [46] classify authentication schemes into several categories:

- *Mutual authentication*, where two network elements confirm identities and exchange data securely,
- *Authentication of drones*, which can use acoustic signals, flight trajectories, gyroscopes, or specialised PUF chips,
- *Authentication of external users*, through passwords, smart cards, and biometrics,
- *Authentication of operators*, using behavioural biometrics.

Beyond these established categories, emerging technologies such as mobile edge computing (MEC), machine learning (ML), and blockchain are also being explored to address authentication challenges. MEC can reduce computational costs and support drone mobility, ML can predict threats by learning network entity behaviour, and blockchain can offer tamper-resistant and decentralised transactions. These advancements aim to achieve robust authentication solutions despite the unique challenges posed by the IoD environment.

The literature in this area is extensive and technically diverse, but shares a common architectural assumption: a trusted ground station or centralised authority mediates identity verification. This assumption may not hold in contested multi-national environments where ground connectivity is intermittent, and the verifying party may belong to a different national authority than the prover.

2.1.2. Privacy-Preserving Identification and Authentication Mechanisms

At the protocol level, most IoD schemes implement mutual authentication and session key establishment between UAVs and a ground control or server. Bera et al.'s ACPBS-IoT protocol [12] is a representative example. Designed for battlefield surveillance with drone-assisted IoT, it uses ECC-based mutual authentication and temporary identifiers to achieve device anonymity and untraceability. The protocol is analysed in the real-or-random (ROR) model and verified with AVISPA (automated validation of Internet security protocols and applications) to resist replay, man-in-the-middle, impersonation,

and ephemeral-secret leakage attacks. A key design goal is damage containment after node capture, ensuring that the compromise of one device's credentials does not affect others.

Similarly, Zhang et al. [82] propose LAPEC, an ECC-based lightweight authentication protocol for UAVs. LAPEC introduces a pre-registration phase in which UAVs and controllers enrol with a trusted authority and a refined session key update mechanism to ensure backward secrecy. Like ACPBS-IoT, LAPEC uses short-lived identifiers and fresh randomness to prevent passive observers from linking protocol runs to specific UAVs.

Beyond these ECC-based designs, several lightweight IoD authentication schemes aim to hide permanent identifiers while allowing controllers or infrastructure nodes to recognise authorised drones. Several hash-centric protocols derive pseudonyms or one-time authenticators from secret seeds, Chebyshev chaotic maps, and keyed hash functions. In this way, each session is tagged with a different label, unlinkable without knowledge of the underlying seed. In these schemes, the ground station maintains the mapping from long-term identities to hash-derived tokens, while external eavesdroppers observe only short, random-looking values.

A related approach uses short-lived certificates rather than hash-derived tokens. ECC-based IoD frameworks such as SENTINEL (described in [46]) use short binary certificates and ECDSA signatures for authentication. Certificates encode stable UAV identities, but are transmitted only to the trusted infrastructure, minimising static metadata exposure over the air. Other PKI-based IoD schemes support dynamic enrolment, revocation, and handover, but focus on hiding long-term names from third parties rather than the infrastructure itself.

A third strategy decouples "who you are" from "where you are" by tying identification to mobility patterns rather than fixed identifiers. UAVouch (described in [46]), for example, combines classical certificates with continuous monitoring of a UAV's position and trajectory. The controller flags deviations from expected flight paths as potential impersonation or Sybil behaviour [50].

Orthogonal to all of the above, another line of work authenticates drones using behavioural or physical fingerprints without cryptographic session setup. Kalman-filter-based schemes (surveyed in [46]) derive flight behaviour profiles from telemetry data, while federated learning over RF features can classify drones based on physical-layer emissions. These mechanisms support privacy-preserving identification, but are currently primarily used for intrusion detection rather than formal key-establishing handshakes.

These efforts are set against the backdrop of emerging "Remote ID" regulations, which require civil drones to broadcast plain-text identifiers, locations, and, indirectly, operator information every second [58]. Sciancalepore's analysis of RID shows that such broadcasts enable long-term trajectory reconstruction and serious privacy risks. Proposed mitigations include encrypting operator metadata, using rapidly rotating broadcast identifiers authenticated only to authorised receivers, and applying differential-privacy-style obfuscation to location beacons. However, the stringent constraints of RID beacons and the limited resources of small UAVS make these solutions challenging to implement.

The structural limitation shared across these strategies is that unlinkability is provided against external eavesdroppers but not against the verifying infrastructure itself. The ground station retains the mapping from long-term identities to session tokens and must thus remain online and trusted. In settings where the verifier belongs to a different national authority or where ground connectivity cannot be assumed, this architecture is insufficient.

2.1.3. Hardware-Rooted Identification and Authentication

A second strand of IoD work anchors UAV identity in dedicated hardware roots of trust such as trusted platform modules (TPMs), hardware security modules (HSMs), and physical unclonable functions (PUFs). Michailidis and Vouyioukas [46] argue that the volatility of UAV deployments and their exposure to physical capture make non-volatile key storage a liability. TPMs and HSMs can isolate long-term private keys and offload cryptographic operations from the main flight controller, but they still rely on conventional PKI, where identities are bound to static certificates. Privacy in these designs is therefore largely determined by the surrounding protocol rather than by the hardware root itself.

PUF-based schemes take a different approach, exploiting silicon manufacturing variations to derive

device-unique responses on demand and eliminating the need to store long-term secrets entirely. Sen et al. [60] propose a PUF-enabled authentication framework for Flying Ad Hoc Networks (FANETs). During registration, the ground station queries the UAV's PUF with a set of challenges and stores the resulting responses, while the UAV retains only a single initial challenge and a temporary identifier; no keys are stored on board. Subsequent handshakes combine fresh nonces, disposable pseudonyms, and PUF responses to achieve mutual authentication and session key derivation. The protocol provides anonymity, forward secrecy, and resistance to node capture and cloning, as verified via Mao-Boyd logic and AVISPA analysis.

Building on the same principle, earlier PUF-based schemes such as PARTH (described in [46]) target multi-layer UAV networks where a trusted ground station authenticates a small set of leader drones, which in turn authenticate resource-constrained mini-drones in their cluster. The verifier challenges the UAV's PUF using pre-restored challenge-response pairs (CRPs), and the UAV's responses are mixed with nonces to produce session keys. Since CRPs never leave the verifier and UAVs store no long-term secrets, physical adversaries cannot easily extract identifiers or signing keys from a captured drone. A related three-layer protocol for UAV-assisted sensor networks, surveyed in [46], also uses PUFs to derive session keys between sensors, UAVs, and base stations, ensuring that compromise in one layer does not expose others.

Gope et al. (described in [46]) extend this hardware-rooted approach to MEC IoD models. Their double-PUF configuration embeds one PUF in the UAV's memory subsystem and another in its control logic. MEC nodes authenticate UAVs based on PUF responses without the UAV storing explicit secret keys. This design ensures that invasive attacks recovering memory contents cannot clone the UAV's identity, and per-session keys derived from fresh challenges prevent linking previous flights.

Other designs combine PUFs with auxiliary technologies to address specific deployment constraints. A SRAM PUF–based RFID scheme, described in [46], uses weak PUF responses as intrinsic "fingerprints" for low-cost RFID tags mounted on drones. Ground interrogators authenticate tags by verifying that their noisy PUF responses match the expected codewords under a fuzzy extractor. CoMSeC++ (described in [46]) employs arbiter PUFs and lightweight hash functions for batch authentication in UAV-assisted wireless sensor networks. UAVs never store shared secret keys, and identifiers are derived from current challenges and responses rather than static IDs.

PUFs provide a hardware root of identity that resists cloning and key extraction, offering two key privacy benefits for IoD:

1. Capturing and reverse-engineering a single UAV does not reveal long-term secrets or keys for other drones, limiting the impact of physical compromise [46, 60].
2. Temporary identifiers derived from PUF responses can be rotated frequently without complex key distribution [46].

However, PUFs also introduce challenges:

- Their outputs are noisy and require error correction or fuzzy extractors, increasing the implementation complexity.
- Poorly designed error correction can leak information about the underlying response distribution [64].
- Exposing too many CRPs risks ML attacks that undermine unclonability and anonymity [64].

Thus, PUF-based schemes primarily target robust authentication and resistance to physical capture, with privacy benefits emerging as a byproduct rather than a fully formalised anonymity model.

The most relevant work beyond the UAV-specific literature is Petzi et al., who propose a PUF-based authentication framework for IoT that explicitly targets a strong physical adversary model, including invasive attacks and temporary device compromise [53]. They provide a decision methodology for when a PUF is required and what type of PUF is appropriate given the device's hardware and threat model. Their scheme combines a perfectly hiding commitment over the PUF response with zero knowledge and conceals both the response and the underlying challenge using an ephemeral Diffie-Hellman exchange, allowing safe challenge reuse while preventing an attacker with temporary physical access from learning either value. Performance measurements on a constrained device suggest that ZK-augmented

PUF protocols are feasible even for low-power nodes. Their interactive, ECC-based zero-knowledge construction does not, however, support non-interactive forwarding or multi-observer verification, distinguishing it from the non-interactive design pursued in this work.

In summary, hardware-rooted schemes substantially improve resistance to physical compromise relative to software-only approaches. The gap that remains is composition: hardware provenance and policy authorisation are treated as separate concerns across the surveyed literature, and no scheme combines hardware binding with zero-knowledge proof in a single constrained relation supporting multi-observer forwarding.

2.1.4. Distributed Trust and Scalable Collaborative Verification

Group-oriented authentication schemes address scalability in swarm settings, where individual 5G/6G core-network authentication of each UAV quickly becomes a bottleneck. Aydin et al. [5] introduce a group authentication framework where ”guard drones” authenticate newly arriving UAVs using polynomial-based group secrets and elliptic-curve operations. This avoids repeated round-trips to 5G core functions and reduces latency, scaling to swarm unification without per-node interaction with the core network. From an identification perspective, this framework treats membership as the primary identity claim, delegating fine-grained role and policy enforcement to higher-layer logic.

While Aydin et al. focus on membership-level scalability, other works explore decentralised identity constructs for more specific IoD deployments such as delivery systems. Parameswarath and Sikdar [52] propose using W3C Decentralised Identifiers (DIDs) and Verifiable Credentials (VCs) for mutual authentication and session-key establishment. Credential presentation replaces the disclosure of persistent identifiers, while revocation is supported via cryptographic accumulators. This credential-centric model shifts UAV identification from hardware- or network-bound identifiers to privacy-preserving attribute proofs. However, these approaches remain embedded in classical public-key and accumulator frameworks and are designed for cooperative commercial environments.

Overall, the existing literature demonstrates substantial progress in individual aspects of UAV authentication, yet no surveyed approach satisfies the full set of requirements identified in Section 1.2.1. The following subsection therefore reviews emerging work on zero-knowledge attestation, which most closely aligns with the objectives of this thesis.

2.1.5. Zero-Knowledge Attestation for UAV Authorisation

Zero-knowledge attestation enables UAVs to prove authorisation or policy compliance without revealing underlying identifiers or sensitive mission data. Yahuza et al. [81] draw a useful distinction between identification, which binds keys or tags to UAVs and operators, and authentication, which proves live possession of those keys. Zero-knowledge attestation addresses both simultaneously: a UAV can prove it holds valid credentials and is currently authorised, without disclosing what those credentials contain.

Naziri et al.’s ZAPS [49] is the closest prior work to the attestation approach pursued in this thesis. It uses zk-SNARKs to let a drone prove authorisation and geofenced flight path compliance without disclosing exact waypoints, combining ECDH key exchange with a succinct non-interactive zero-knowledge proof to achieve mutual authentication and flight path privacy. Proof verification is fast enough for per-mission checks, while proof generation can be amortised over path planning. ZAPS does not report standalone proof verification latency as a separate figure; indicative estimates used in comparative tables later in this thesis are based on standard Groth16 costs and should not be read as measurements reported by that work.

Extending the idea of zero-knowledge location and authorisation proofs to multi-stakeholder settings, Koulianos et al. [40] propose a public-blockchain framework where UAVs prove via zk-SNARKs that they hold valid certificates and operate within authorised areas. By encoding location and authorisation checks in the proving circuit, the system turns a binary authorisation question into a verification step over a succinct proof, decoupling authorisation from direct access to flight logs. On-chain verification is a platform feature of the blockchain rather than a protocol-level forwarding mechanism, and proofs are bound to a specific contract instance rather than being portable to an arbitrary verifier.

Similar ideas appear in the IoT literature. Wu et al.’s zk-PoL [78] uses a blockchain and zk-SNARKs

to let users prove presence in a location without revealing exact coordinates, supporting hierarchical privacy levels by varying the granularity of the proved region. Although zk-PoL targets location-based services rather than UAV operations, it reinforces the feasibility of succinct ZK proofs of geofencing constraints with constant-time verification suitable for delay-tolerant services.

Zero-knowledge techniques also appear in blockchain-centred IoD frameworks more broadly. SpyChain [3] uses pairing-based non-interactive zero-knowledge (NIZK) proofs within a lightweight cooperative blockchain to generate unlinkable ring-style signatures on transactions, verifiable as coming from an authorised group member but not linkable to a specific airframe. The blockchain records only pseudonymous transaction data and zero-knowledge proofs, with the mapping from on-chain pseudonyms to UAVs kept off-chain by the authority.

Approaching the problem from a connectivity rather than authorisation angle, Walshe et al. [77] propose a Merkle-tree-based NIZK protocol for IoT authentication in reduced-connectivity environments, allowing a device to authenticate in a single pre-computed message. This is particularly relevant for UAVs operating with intermittent links, though the scheme does not address hardware attestation or policy authorisation. Beyond the UAV domain, zk-SNARK-based anonymous-credential systems such as zk-creds [56] demonstrate that reusable, expressive zero-knowledge attribute proofs are already practical in the web-identity domain, suggesting a design pattern that could be adapted to UAV fleet credentials.

The gap analysis in Table 2.1 summarises the properties of the closest prior work against the requirements of Section 1.2.1. The ZK column marks any scheme using a formal zero-knowledge construction; LAPEC does not. The HW-bound column marks schemes that bind authentication to a device-specific hardware secret; only Petzi et al. do so using a PUF. The Forward. column marks schemes that design the proof for re-verification by an arbitrary observer holding only the verification key; all prior schemes require server-side, blockchain, or interactive verification rather than a portable verification key. The PQ-migr. column marks schemes whose relation uses only hash functions and Merkle paths such that a post-quantum backend could be substituted without modifying the relation; all prior schemes embed ECC or pairing operations in their core protocol, and Walshe et al. is marked Unclear because the paper does not address this.

Table 2.1: Gap analysis against closest prior work. ZK: zero-knowledge construction. HW-bound: hardware-bound secret possession. Forward.: protocol-level support for forwarding a proof to an arbitrary third-party observer holding only the verification key. PQ-migr.: relation-level compatibility with post-quantum backends without relation redesign, referring to the relation definition and not the proof system instantiation.

Scheme	ZK	HW-bound	Forward.	PQ-migr.
LAPEC [82]	No	No	No	No
Petzi et al. [53]	Yes	Yes (PUF)	No	No
ZAPS [49]	Yes	No	No	No
Koulianos et al. [40]	Yes	No	No[†]	No
Walshe et al. [77]	Yes	No	No	Unclear
This work	**Yes**	**Yes**	**Yes**[‡]	**Yes**

[†] On-chain verifiability is a platform property of Ethereum, not a design property of the proof system; proofs are bound to a specific contract instance rather than being portable to an arbitrary verifier holding only vk.
[‡] Protocol-level capability. The empirical evaluation demonstrates forwarding between co-located processes rather than across a distributed swarm topology.

No surveyed scheme simultaneously provides zero-knowledge attestation, hardware-bound secret possession, and protocol-level forwarding to an arbitrary third-party observer within a relation compatible with post-quantum backends. The designs reviewed here each address one or two of these properties in isolation. This gap motivates the joint-circuit design of Chapter 3, which enforces all three properties within a single constrained relation.

2.1.6. Post-Quantum Cryptographic Foundations for Authentication

Post-quantum cryptography for UAVs has matured from survey-level feasibility arguments to concrete protocol designs and embedded hardware benchmarks. Khan et al. [37] survey PQC candidates across lattice-based, code-based, hash-based, and multivariate families, establishing that quantum-resistant key encapsulation mechanisms (KEMs) and signature schemes are necessary for long-lived UAV deployments. Aissaoui et al. [2] confirm this empirically, benchmarking Kyber-768, HQC, and BIKE against RSA-2048 and ECDH on embedded UAV hardware. Kyber-768 completes key generation and encapsulation in milliseconds even under CPU load.

Building on these primitives, several protocol-level designs instantiate PQC directly in UAV authentication. LIGKYX [79] combines Kyber KEM with HMAC in a two-phase protocol between UAVs and ground stations, with ProVerif analysis confirming resistance to replay, impersonation, and known-session-key attacks. QRAC-UAV [11] adds an anonymity layer, using a Ring-LWE KEM, a fuzzy extractor over biometric templates, and a lattice-based anonymous-credential layer to achieve mutual authentication and untraceability, verified with Scyther and Tamarin. Roodsari et al. [55] take a more comprehensive approach, combining MLWE-based authenticated key exchange with dynamic credentials, timestamps, and context-aware checks for continuous integrity assurance, with security proven in the BPR model.

At the swarm layer, He et al. [34] integrate Kyber-based KEM, sparse Merkle trees, and lattice-based signatures for post-quantum group-key agreement, where an edge node builds an SMT over UAV identifiers and UAVs prove membership using Merkle paths. Ayebie et al. [6] push decentralisation further, combining ring-oscillator PUFs, lattice-based SIS primitives, and a compact lattice accumulator to allow any two drones to mutually authenticate using only local state, without relying on a ground control station.

Two further works address specific deployment contexts. Tawfeeq et al. [66] target military IoD, replacing ECC with Dilithium signatures in a three-party architecture of drones, soldiers, and a command centre, noting increased processing time and energy consumption relative to ECC-based schemes. Kim et al.'s pqRID [38] addresses Remote ID (RID) privacy, provisioning drones with Dilithium-signed pseudonym certificates and demonstrating on a Raspberry Pi 4B that generating a Dilithium-based authentication message takes approximately 9 ms and 0.8 J, confirming compatibility with RID broadcast constraints.

The emerging connection between post-quantum cryptography and zero-knowledge attestation is addressed by Boschini et al. [15], who construct efficient post-quantum SNARKs for RSIS and RLWE relations, demonstrating that lattice-based zero-knowledge proofs of possession are practical and can enable post-quantum group signatures with anonymity, traceability, and non-frameability.

The literature hence shows that post-quantum authentication for UAVs is feasible across a range of hardware classes and network configurations. The remaining gap relevant to this thesis is the integration of post-quantum security with hardware-bound zero-knowledge attestation within a single relation designed for non-interactive forwarding, a combination that has not been demonstrated in the surveyed literature.

2.1.7. Limitations of Existing Approaches

The surveyed literature demonstrates meaningful progress across authentication, privacy, hardware binding, and post-quantum security. Several persistent limitations remain, however, and collectively motivate the design choices of this thesis.

- **Infrastructure dependence.** Most schemes rely on trusted ground stations, blockchains, or centralised authorities to mediate verification, creating single points of failure that may not be tolerable in contested or infrastructure-denied environments [46, 49].
- **Separation of concerns.** Hardware binding, zero-knowledge privacy, and policy authorisation are addressed individually across different bodies of work but rarely composed within a single protocol, creating vulnerability at composition boundaries.
- **Privacy and accountability trade-offs.** Privacy-preserving mechanisms often conflict with regulatory requirements such as Remote ID, which mandate plain-text identifier broadcasts, or with operational needs such as auditing and revocation [58, 38].

- **PUF limitations.** PUFs are sensitive to environmental factors including voltage, temperature, and ageing, and excessive exposure of challenge-response pairs risks machine-learning attacks that undermine unclonability [60, 64].
- **Post-quantum integration.** While PQC schemes are increasingly practical, their integration with privacy-preserving zero-knowledge techniques is nascent, and performance trade-offs on embedded hardware remain understudied [15].
- **Setup and composability.** Security guarantees often depend on careful parameter selection and trusted setups that may not be feasible in multi-stakeholder environments. Zero-knowledge and PQC solutions are frequently bespoke and lack modularity for general IoD use [3, 15].
- **Limited real-world validation.** Most proposals are evaluated in simulation under controlled conditions, with limited validation under dynamic IoD network conditions and realistic hardware constraints [79].

These gaps collectively motivate a protocol that jointly addresses membership, hardware binding, and policy authorisation within a single zero-knowledge relation designed for non-interactive multi-observer verification without infrastructure dependency. The specific design choices made in response are described in Chapter 3.

2.2. Preliminaries

This section introduces the cryptographic foundations and operational context necessary to understand the protocol developed in this work. The presentation assumes familiarity with basic probability theory and discrete mathematics, but does not presuppose prior exposure to zero-knowledge proofs or hardware security primitives.

2.2.1. Cryptographic Foundations

The protocol relies on cryptographic hash functions, commitment schemes, Merkle trees, and zero-knowledge proofs as its core building blocks. This section introduces these primitives and situates them within the broader landscape of credential and authentication technologies, including digital signatures, PKI, and anonymous credentials, which were considered as design alternatives.

A *cryptographic hash function* is a deterministic, efficiently computable function $H : \{0,1\}^* \to \{0,1\}^n$ mapping inputs of arbitrary length to a fixed-length digest of n bits. Following Smart [62], three security properties are required: preimage resistance (given d, no probabilistic polynomial-time (PPT) adversary finds x with $H(x) = d$), second-preimage resistance (given x, no PPT adversary finds $x' \neq x$ with $H(x') = H(x)$), and collision resistance (no PPT adversary finds any pair (x, x') with $x \neq x'$ and $H(x) = H(x')$). Collision resistance implies second-preimage resistance but not preimage resistance. The choice of hash function also carries implications for circuit-based proof systems: SHA-256 requires significant constraint overhead when expressed as an R1CS, whereas Poseidon [32] is defined natively over prime fields and admits an efficient arithmetic circuit representation, substantially reducing circuit complexity for Merkle tree computations.

Closely related to hash functions, *commitment schemes* allow a party to bind itself to a value without revealing it. A committer publishes $c = \mathsf{Com}(v; r)$ for a randomly chosen blinding factor r, and later reveals (v, r) for verification. A secure scheme is *binding* (no PPT committer can open the same commitment to two different values) and *hiding* (c is indistinguishable from a commitment to any other value). Hash-based commitments $c = H(v \| r)$ achieve binding under collision resistance and hiding under ROM, given r is random and secret [62]. These binding and hiding properties underpin the minimal-disclosure guarantees sought in privacy-preserving authentication.

Hash functions and commitments together enable *Merkle trees* [45], binary hash trees in which each leaf holds $H(d_i)$ for a data element d_i, each internal node holds the hash of its two children, and the root authenticates the entire set under collision resistance. A membership proof for d_i consists of the sibling hashes along the path from leaf to root, requiring only $O(\log n)$ hash computations for both generation and verification. A *Sparse Merkle Tree* (SMT) extends this to a fixed key space $\{0,1\}^{256}$, assigning each possible key a unique leaf position with empty leaves receiving default hash values [31]. This enables efficient proofs of both membership and non-membership, and because leaf positions are

index-derived rather than insertion-order-dependent, the same set of entries always produces the same root hash, a property that is central to the revocation model of this work.

While hash-based primitives form the core of the protocol, it is useful to situate them against the alternatives. A *digital signature scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ satisfies existential unforgeability under chosen-message attack (EUF-CMA): no PPT adversary given signing oracle access can produce a valid signature on a new message [30]. Digital signatures underpin PKI, in which a Certificate Authority binds a public key to an identity via a signed certificate. PKI is the dominant real-world approach to identity management, but it is fundamentally at odds with unlinkability: a certificate is a persistent identifier in every interaction, allowing an observer to reconstruct session histories and behavioural patterns. Certificate revocation further requires CRL or OCSP queries, both of which assume infrastructure connectivity that cannot be guaranteed in contested environments.

Anonymous credential schemes [18] address the linkability problem by letting a holder prove possession of attributes without revealing a persistent identifier. The Camenisch-Lysyanskaya scheme issues credentials as signatures over committed attribute values, and BBS+ signatures [14] extend this to efficient multi-attribute selective disclosure with computationally unlinkable presentations. These schemes substantially improve on PKI for privacy, but still require the verifier to trust the issuing authority directly and typically assume online access for revocation. The approach adopted in this work differs: verification is performed entirely against locally held SMT roots and a zero-knowledge proof, removing the need for any per-session interaction with an issuing authority.

2.2.2. Zero-Knowledge Proof Systems

Zero-knowledge proof systems, first defined by Goldwasser, Micali, and Rackoff [29], constitute the cryptographic core of the protocol proposed in this work. Let $R \subseteq \mathcal{X} \times \mathcal{W}$ be an NP relation, where $x \in \mathcal{X}$ is a public statement and $w \in \mathcal{W}$ is a private witness. A proof system for R is a protocol between a prover $\mathcal{P}(x, w)$ and a verifier $\mathcal{V}(x)$, at the end of which $\mathcal{V}$ outputs accept or reject. Four properties are required.

Definition 1 (Completeness)**.** *A proof system is* complete *if for all $(x, w) \in R$, an honest prover convinces an honest verifier with overwhelming probability.*

Definition 2 (Soundness)**.** *A proof system is* sound *if for all $x \notin \mathcal{L}_R$ and all PPT cheating provers $\mathcal{P}^*$, $\mathcal{V}$ accepts with at most negligible probability.*

Definition 3 (Knowledge Soundness)**.** *A proof system is* knowledge sound *if there exists a PPT extractor $\mathcal{E}$ and a polynomial p such that, for any PPT prover $\mathcal{P}^*$ that produces an accepting proof with probability $\epsilon(\lambda)$, the extractor $\mathcal{E}$ outputs a valid witness w with probability at least $\epsilon(\lambda)/p(\lambda)$. In other words, the extractor succeeds in proportion to how often the prover does.*

Definition 4 (Zero-Knowledge)**.** *A proof system is* zero-knowledge *if there exists a PPT simulator $\mathcal{S}$ such that, given only x and without access to w, the output of $\mathcal{S}(x)$ is computationally indistinguishable from a real proof transcript:*

$$\left\{\mathsf{View}_{\mathcal{V}}\big(\mathcal{P}(x,w),\, \mathcal{V}(x)\big)\right\} \;\approx_c\; \left\{\mathcal{S}(x)\right\} \quad \textit{for all } (x,w) \in R.$$

Together, these properties mean that a proof π carries no information about w beyond the single bit that $(x, w) \in R$, the cryptographic formalisation of minimal disclosure. A prover can demonstrate authorisation, registry membership, and hardware binding without revealing the underlying attributes. Throughout this work, λ denotes the security parameter, implicit in expressions such as $ID_{\mathsf{priv}} \xleftarrow{\$} \{0,1\}^{\lambda}$, and adversaries are PPT algorithms. A function $\mu(\lambda)$ is *negligible* if for every polynomial $p(\cdot)$ there exists λ_0 such that $\mu(\lambda) < 1/p(\lambda)$ for all $\lambda > \lambda_0$; security claims throughout this work hold except with probability at most $\mu(\lambda)$, conservatively upper-bounded by $O(2^{-\lambda})$ under the collision resistance of H and the soundness of the instantiated proof system.

Interactive zero-knowledge protocols require synchronous communication, which is impractical in environments with intermittent connectivity. Blum, Feldman, and Micali [13] showed that interaction can be eliminated if the prover and verifier share a *Common Reference String* (CRS), generated during a one-time trusted setup. The prover computes $\pi \leftarrow \mathsf{Prove}(\mathsf{pk}, x, w)$ offline, and any party holding the verification key vk can independently check $\mathsf{Verify}(\mathsf{vk}, x, \pi)$. Because Verify is a deterministic function of its inputs alone, any two parties with identical inputs reach identical verdicts without coordination,

giving the proof public verifiability: it can be forwarded, stored, and re-verified by any number of parties without involving the original prover [33]. The CRS is generated from a trapdoor τ; recovery of τ would allow proof forgery. In practice, a multi-party computation ceremony destroys the trapdoor if at least one participant is honest [16].

A *zkSNARK* is a NIZK proof system that is additionally succinct, meaning proof size and verification time are sublinear in the computation size. To apply a zkSNARK, the relation R is expressed as an arithmetic circuit over a prime field, compiled into a Rank-1 Constraint System (R1CS), and further transformed into a Quadratic Arithmetic Program (QAP) [28]. This work uses the Groth16 construction [33] via the libsnark library [59]. Groth16 produces the smallest known pairing-based NIZK proofs: exactly three elliptic curve group elements, with verification requiring a constant number of bilinear pairing operations independent of circuit size. Its security relies on the q-Power Knowledge of Exponent assumption and the q-Strong Diffie-Hellman assumption in the generic group model [33]. The Knowledge of Exponent assumption is an extractability assumption considered non-falsifiable or heuristic by some cryptographers, as it posits the existence of an extractor without a reduction from a standard hardness problem; this is a known limitation of Groth16's security argument. Neither assumption holds against a quantum adversary, as Shor's algorithm solves the discrete logarithm problem in polynomial time [61]. The post-quantum implications are addressed in Section 2.2.4.

Overall, NIZK proofs in the CRS model provide the three properties this protocol requires: a prover can convince any verifier of a statement without revealing the witness, the proof is self-contained and can be forwarded to secondary verifiers without prover involvement, and the security of the Groth16 instantiation is well-characterised under standard assumptions, with the limitations of those assumptions explicitly acknowledged.

2.2.3. Consistency and Determinism in Distributed Verification

When a proof is verified by multiple independent parties, a natural question arises: under what conditions will all parties reach the same decision? In systems that rely on consensus protocols, such as Byzantine fault-tolerant state machine replication [42], agreement requires rounds of communication and must be engineered at runtime. In NIZK-based verification, consistency follows from a simpler and stronger property: determinism.

Because $\mathsf{Verify}(\mathsf{vk}, \phi, \pi)$ is a deterministic function with no randomness and no external state, any two parties invoking it on identical inputs will produce identical outputs. Agreement across a set of verifiers is thus not a coordination problem but a precondition problem: it requires only that all verifiers hold the same vk and the same public input vector ϕ. Disagreement is diagnostic of one of two conditions: a verifier holds a stale or corrupted anchor, or the proof has been modified in transit. Unlike probabilistic authentication schemes, where decisions depend on thresholds or aggregated evidence and consistency requires active coordination [41], deterministic NIZK verification achieves strong consistency as a structural consequence of the scheme, with no runtime communication between verifiers. This property is central to the multi-observer design evaluated in this work and is formalised as a required protocol property in Section 3.4.1.

2.2.4. Post-Quantum Compatibility

The protocol operates in a coalition context where authentication infrastructure may need to remain trustworthy over extended time horizons, making long-term cryptographic security a relevant design concern. As noted in Section 2.2.2, the Groth16 proof system relies on elliptic-curve pairings and is not secure against a quantum adversary. This section contextualises that limitation and explains how the design accounts for future migration.

Classical public-key cryptography derives its security from the hardness of integer factorisation and the discrete logarithm problem. Shor's algorithm [61], executable on a sufficiently large fault-tolerant quantum computer, solves both in polynomial time. By contrast, Grover's algorithm provides only a quadratic speedup for unstructured search, meaning hash functions and symmetric primitives are weakened but not broken; doubling digest lengths restores the original security margin. This asymmetry is directly relevant: the hash-based structures used throughout this work, including the SMT roots and the hardware commitment, retain meaningful quantum resilience, while the pairing-based proof system does not.

The urgency of transitioning to post-quantum cryptography is reflected in recent institutional guidance. NIST completed its standardisation process in August 2024, publishing FIPS 203 (ML-KEM) and FIPS 204 (ML-DSA) [48], and the European Commission has issued a coordinated recommendation for migration across critical infrastructure [23]. Post-quantum zkSNARK backends, including STARK-based [9] and lattice-based constructions, avoid pairing operations and are conjectured to resist quantum attacks, but currently carry substantially larger proof sizes and higher prover costs, and none has reached comparable tooling maturity. Using Groth16 in the current instantiation is therefore a deliberate pragmatic choice. The design accounts for this by defining the ZK relation, the SMT-based trust anchor model, and the verification protocol independently of the underlying proof system, allowing a post-quantum backend to be substituted once a sufficiently mature implementation becomes available.

2.2.5. Hardware-Based Trust

Cryptographic authentication protocols depend on secrets being bound to a specific entity in a trustworthy way. Software-only approaches enforce this binding through access control and key storage, both of which are vulnerable to software compromise or memory extraction. Hardware security mechanisms address this by anchoring trust in physical properties of the device. This section surveys the mechanisms relevant to this work and explains the tradeoffs that motivate the choice of PUFs in the proposed protocol.

Trusted Platform Modules (TPMs) and *FIDO2* authenticators [75, 24] represent the dominant deployed hardware trust mechanisms. Both provide tamper-resistant key storage and attestation capabilities, and both have been widely standardised and adopted. For this work, however, they share two disqualifying limitations. First, they store cryptographic keys in non-volatile memory, making key material potentially extractable by a motivated adversary with physical access; practical attacks have demonstrated this is not merely theoretical [47, 36]. Second, both rely on PKI-based certificate chains rooted in a manufacturer endorsement key, reintroducing the linkability and infrastructure-connectivity dependencies that this protocol is designed to avoid.

Physical Unclonable Functions (PUFs) offer a qualitatively different approach that eliminates the stored-secret problem [51, 27]. A PUF exploits uncontrollable sub-micron manufacturing variations to implement a device-specific mapping from challenges to responses: given a challenge $c \in \{0,1\}^k$, the PUF produces a response $r = \mathrm{PUF}(c) \in \{0,1\}^\ell$ determined entirely by the physical characteristics of that device instance. Four properties characterise a secure PUF: *uniqueness* (responses of distinct devices are statistically independent); *reproducibility* (the same device produces consistent responses modulo bounded noise); *unclonability* (it is physically infeasible to reproduce a device's challenge-response behaviour); and *unpredictability* (observed challenge-response pairs yield no useful information about unseen challenges). Because the device-specific secret is implicit in the physical structure and reproduced on demand, compromising the device's software state or non-volatile storage does not yield the key.

The practical limitation of PUFs is response instability caused by environmental factors such as temperature and supply voltage. This is addressed through helper data schemes [64], in which error-correcting codes applied at enrolment time allow a noisy runtime response to be corrected without leaking information about the underlying PUF response. PUFs are classified into weak variants, with a small fixed challenge space used to derive a stable device-specific key, and strong variants, with a large challenge space queryable directly for authentication [4]. This work uses a weak PUF, and k_{hw} is the key derived from it. At enrolment, a trusted authority records the corrected response as k_{hw}^{enrol}; at runtime, the device regenerates k_{hw} from the PUF and applies the stored helper data. The key exists only transiently in volatile memory during use and cannot be reconstructed from any stored value on the device.

The proof-of-concept implementation evaluated in Chapter 4 does not use a physical PUF; the hardware key is simulated, and the security properties described above therefore characterise the intended deployment model rather than the current prototype. This gap is discussed explicitly in Section 5.3.

2.2.6. UAV Systems, Operational Constraints, and Federation

An Unmanned Aerial Vehicle (UAV) is an aircraft operating without an onboard human pilot, either autonomously or under remote control. In security-critical deployments, UAVs are organised into swarms that coordinate to execute distributed tasks under shared mission parameters [5]. Swarm coordination

requires vehicles to authenticate one another dynamically as the swarm composition changes, over ad hoc wireless links subject to interception, jamming, and spoofing, and without reliable connectivity to ground infrastructure.

This operational context creates a specific set of authentication threats. Impersonation occurs when an adversary presents false credentials to gain swarm acceptance; in coalition environments, the verifying party may have no prior relationship with the claiming asset, making the authentication protocol the sole basis for trust. Replay attacks exploit the absence of per-session binding, allowing a captured exchange to be reused indefinitely. Sybil attacks involve a single adversary presenting multiple false identities to gain disproportionate influence over swarm decisions [50]. Context confusion arises when credentials valid in one operational context are presented in another. Finally, trajectory and identity linkage is a passive threat: certificate-based authentication includes a persistent identifier in every exchange, allowing an observer to reconstruct an asset's patrol route and operational pattern from traffic alone without breaking any cryptographic primitive.

The specific problem addressed in this thesis is encounter authentication: *when two UAVs from different national forces meet in shared airspace, each must verify that the other is a legitimate, non-revoked federation member operating under the correct mission role and sector authorisation, without disclosing sensitive identity or mission attributes, and without real-time infrastructure connectivity*. This differs from civilian Remote ID requirements, which mandate plain-text identity broadcasts; military coalition authentication must be selective, unlinkable across sessions, and infrastructure-independent.

The cryptographic foundation for this is a set of shared trust anchors distributed via the NATO Federated Mission Networking (FMN) model before deployment [63]. Rather than PKI-based certificate hierarchies requiring online revocation queries via CRLs or OCSP, this work uses authenticated digest-based anchors: a single root hash summarises the current state of a registry, and any participant holding that root can verify membership claims locally. Revocation propagates through root updates distributed via authenticated out-of-band channels, without any per-verification infrastructure query.

Finally, single-observer authentication introduces a single point of failure in contested environments. Multi-observer verification addresses this by distributing verification across independent swarm members, each verifying locally against their own copy of the anchor values and reaching the same decision by determinism of the verification function. These requirements motivate the non-interactive, publicly verifiable proof design described in Section 2.2.2 and are formalised as protocol properties in Section 3.4.1.

2.3. Research Questions

This research addresses the following research question and sub-research questions.

2.3.1. Research Question

To what extent can zero-knowledge attestation support identification-friend-or-foe (IFF) decisions among UAVs without relying on identity-based authentication in contested environments?

2.3.2. Sub-Research Questions

1. How can an asset prove that it is operating under an authorised and uncompromised control state[1] without disclosing persistent identifiers?
2. What computational, latency, and communication constraints limit the deployment of zero-knowledge attestation on resource-constrained UAV platforms in contested environments?[2]
3. What properties must a multi-observer verification protocol satisfy to improve trust decisions over single-observer authentication in UAV swarm encounters, and does the proposed design satisfy them?

[1] *Control state* is defined cryptographically as the combination of possession of an enrolment-bound hardware secret, active federation membership, and mission-level policy authorisation as jointly attested by the proof. It does not encompass real-time flight parameters, mission directives, or operational telemetry, which are outside the scope of the zero-knowledge relation.

[2] *Contested environments* refer to communication-degraded, infrastructure-denied, and adversarially observed wireless environments, including scenarios where GPS-based positioning is unreliable or denied. The specific operational conditions under which the protocol is evaluated are defined in Section 2.4.

2.4. Threat Model

This section establishes the adversarial context in which the proposed protocol must operate. It defines the system and its operational constraints, characterises the adversary in terms of capabilities and limitations, identifies the attacks explicitly considered within scope, and maps each attack to a corresponding MITRE ATT&CK for industrial control systems (ICS) technique. For each in-scope attack, the section concludes with the general cryptographic principle and specific protocol design choice that addresses it.

2.4.1. System Context

The system consists of multiple cooperative unmanned aerial vehicles (UAVs) operating in a swarm configuration during persistent border Intelligence, Surveillance, and Reconnaissance (ISR) missions. UAVs are required to autonomously establish local trust with nearby assets under intermittent connectivity and persistent wireless observation.

Cryptographic interaction is initiated *only after* an approaching object has been classified as *potentially friendly* using non-cryptographic sensing mechanisms such as radar, Remote ID (RID), or visual recognition [81]. The outcome of the cryptographic protocol is a *local trust classification* — authorised or unauthorised — and does not constitute an engagement, control, or kinetic decision.

The system does not assume continuous connectivity to a Ground Control Station (GCS) during encounters and must operate securely in infrastructure-degraded or denied environments. Pre-mission policy distribution is handled through Federated Mission Networking (FMN) [63], the NATO-standardised framework for sharing mission configurations, authentication policies, and trusted anchor values across coalition partners before deployment. Figure 2.1 illustrates this upstream provisioning mechanism. Once deployed, UAVs operate autonomously against locally held anchor values without further GCS involvement.

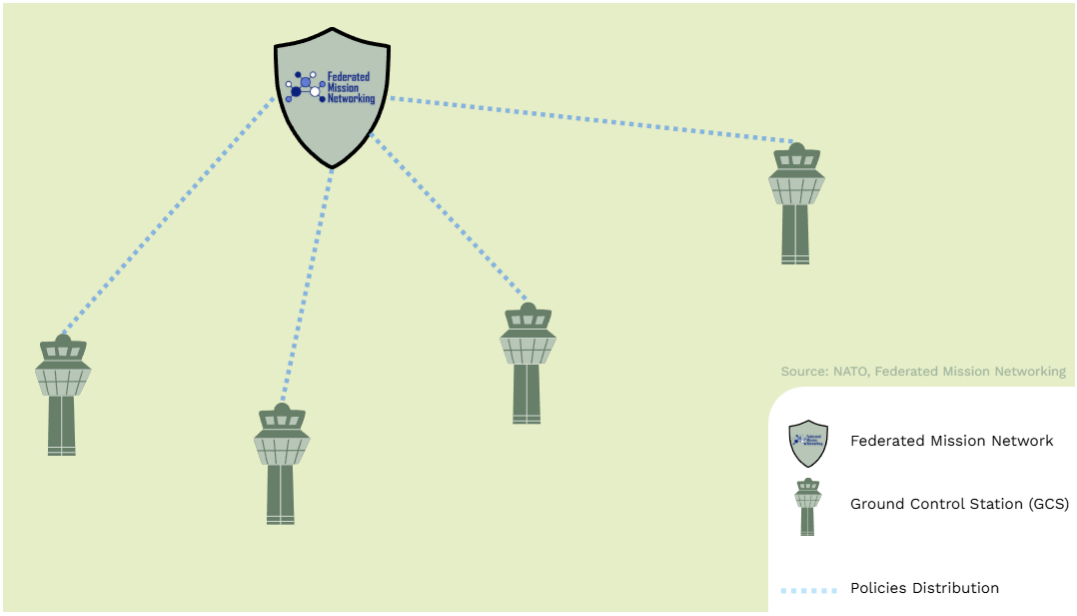

Figure 2.1: Federated Mission Networking (FMN) based policy distribution architecture [63]. Mission configurations, registry roots, policy roots, and verification keys are disseminated to ground control infrastructure before UAV deployment. Once in the field, UAVs verify against locally held copies of these values without real-time GCS connectivity.

Figures 2.2 through 2.6 illustrate two simultaneous UAV encounter scenarios unfolding at the boundary between home and potentially hostile territory. Green arrows indicate established distributed trust within the swarm. Red dashed arrows indicate forwarded proof bundles propagated to swarm peers during verification. Bidirectional orange arrows indicate active authentication exchanges. In both encounters, the swarm operates without real-time ground infrastructure support. The adversary observes both exchanges throughout but cannot extract identity, mission, or authorisation information from the

protocol traffic.

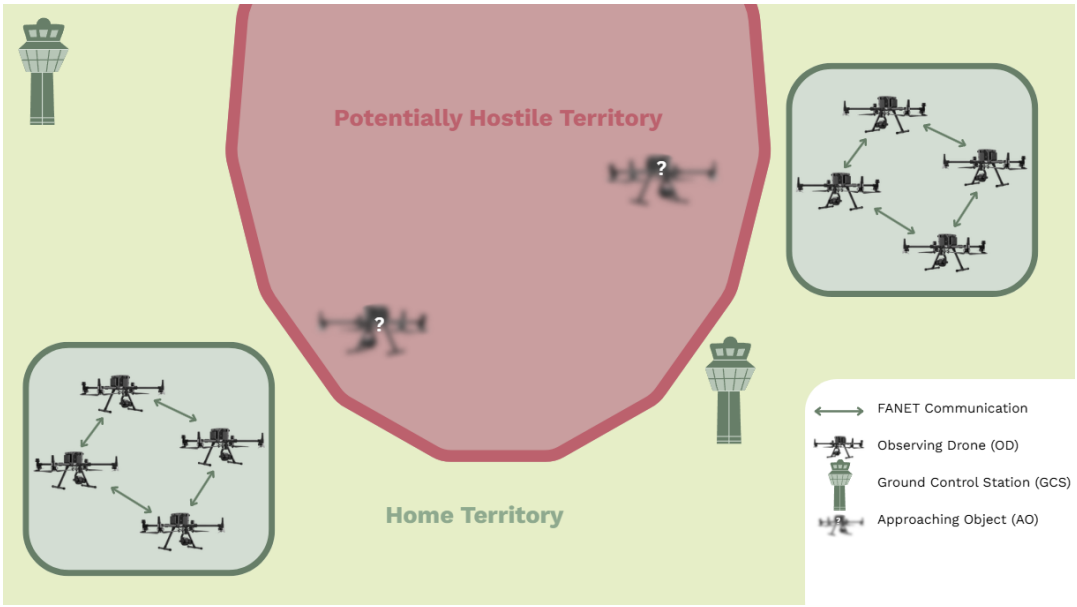

Figure 2.2: UAVs approaching the operational area before any classification or trust assessment has been performed. No authentication procedure has been initiated; the approaching objects are observed as potential entrants into the monitored space.

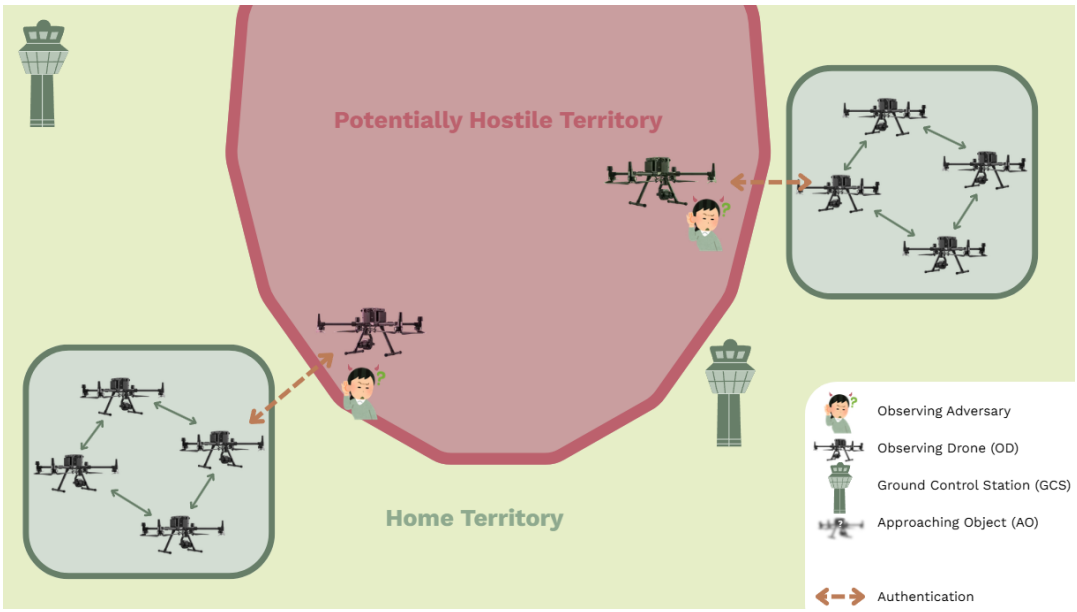

Figure 2.3: Two UAVs are detected near the operational boundary and classified as potentially friendly using non-cryptographic sensing. The Observing Drone independently initiates the NIZK authentication protocol for each encounter. The adversary passively observes the radio-frequency environment.

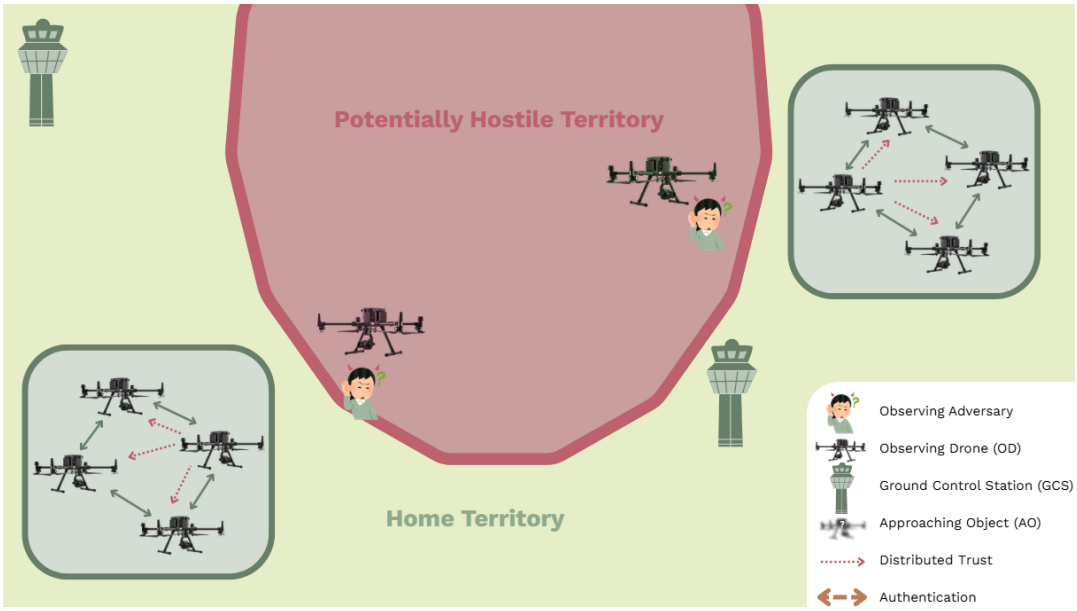

Figure 2.4: Each Observing Drone forwards the received proof bundle (π, ϕ) to neighbouring swarm members for independent secondary verification. This step distributes the trust decision across the swarm without requiring the prover to re-engage.

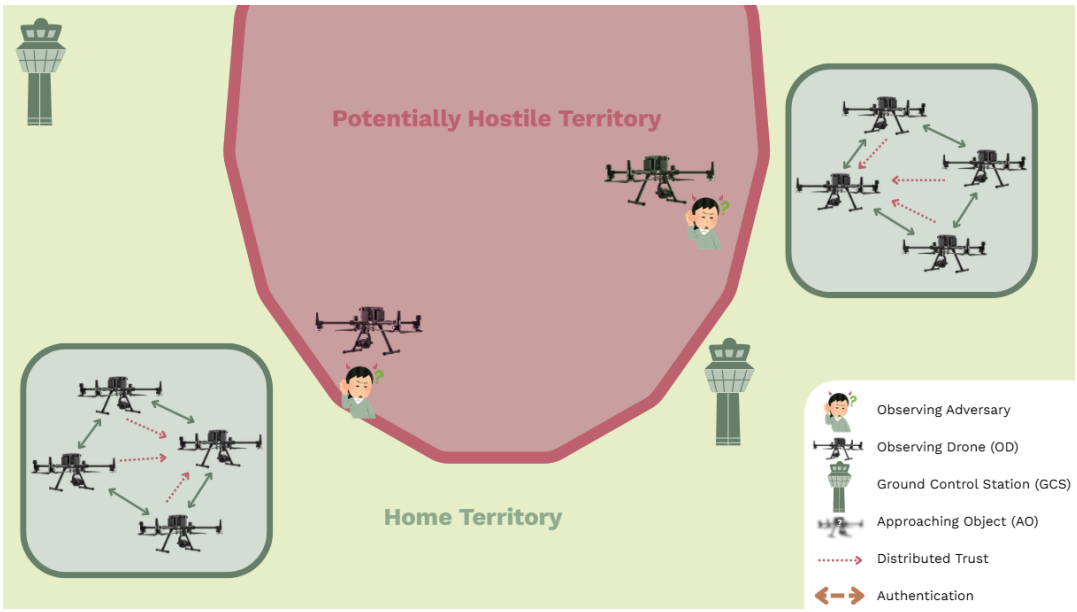

Figure 2.5: Swarm peers independently verify the forwarded proof bundles against their locally held anchor values and return their verdicts through the swarm communication network.

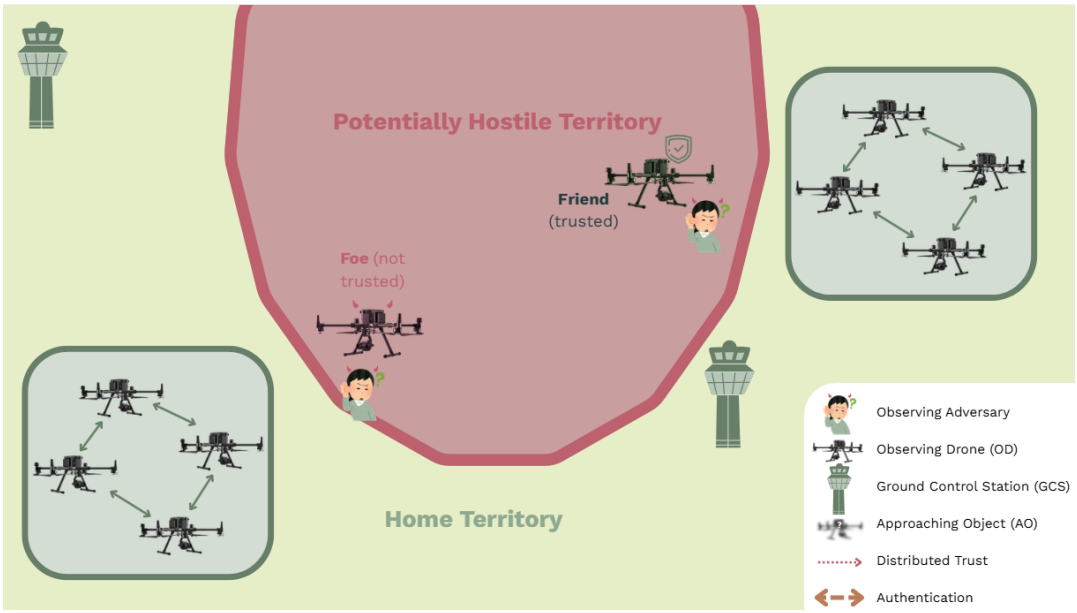

Figure 2.6: Trust decisions are finalised locally. The right-hand approaching object passes all verification checks and is classified as locally trusted. The left-hand approaching object fails verification and is classified as untrusted. The entire process operates autonomously without ground control station involvement.

2.4.2. Adversary Definition

The adversary is modelled as a computationally bounded classical adversary, additionally assumed capable of long-term passive collection of cryptographic material for future quantum cryptanalysis, possessing full control over the wireless protocol layer, including message injection, replay, and modification, but excluding sustained physical-layer denial-of-service or spectrum denial [54]. This exclusion reflects the focus of this work on trust establishment rather than availability and aligns with existing air defence doctrine, where separate electronic warfare countermeasures handle jamming [19]. The current prototype instantiates R_{IFF} with Groth16 over BN128 and thus does not provide post-quantum security. The quantum-harvest capability motivates SR-6 (relation-level post-quantum readiness).

The adversary model is grounded in the MITRE ATT&CK for ICS framework [67], which provides a structured taxonomy of adversary tactics and techniques applicable to cyber-physical systems operating under contested conditions. Where applicable, each attack considered in this work is mapped to a corresponding ICS technique.

2.4.3. Adversary Capabilities

The adversary is assumed to possess the following capabilities.

Passive Wireless Observation The adversary can continuously monitor radio-frequency (RF) channels, record protocol exchanges, and perform long-term traffic analysis in an attempt to infer operational patterns, trust relationships, or mission structure [80]. This capability corresponds to MITRE ATT&CK for ICS technique T0842 (Network Sniffing) [71].

Active Network Manipulation The adversary can inject, modify, delay, or replay wireless messages exchanged between UAVs, including authentication attempts and protocol control messages [81]. This capability corresponds to T0830 (Adversary-in-the-Middle) [70].

Physical Capture and Compromise The adversary may physically capture a cooperative UAV and extract long-term credentials stored in non-tamper-resistant memory, including cryptographic keys, certificates, or firmware images [46]. This capability corresponds to T0862 (Supply Chain Compromise) and T0849 (Masquerading) [74, 72]. The consequences of physical capture and the residual security guarantees are discussed in Section 3.2.

2.4.4. Adversary Limitations

The adversary is subject to the following constraints.

Cryptographic Assumptions Hold The adversary cannot break the cryptographic primitives on which the system relies, including the soundness and zero-knowledge properties of the employed proof system and the binding properties of hash-based commitments, under standard assumptions. The formalisation of these assumptions is given in Section 3.2.

Hardware-Bound Secrets Remain Secure Before Capture The adversary cannot reliably reproduce hardware-derived device secrets across hardware instances [51, 27]. If a device is physically captured and its hardware secret is extracted, the system relies on revocation through the federation registry as a recovery mechanism.

No Coercion of Uncompromised UAVs The adversary cannot coerce uncompromised cooperative UAVs into deviating from protocol execution or disclosing private witnesses during zero-knowledge proof generation.

2.4.5. Attacks Considered

The following attacks are explicitly considered within the scope of this work. Table 2.2 maps each attack to its corresponding MITRE ATT&CK for ICS technique and summarises the consequences of a successful attack.

Credential Replay Attacks The adversary attempts to reuse previously observed authentication messages to masquerade as a UAV, exploiting the absence of session binding in naive authentication schemes [77]. MITRE ATT&CK for ICS does not provide a dedicated technique for authentication message replay in UAV contexts; the closest structural analogues are T0856 (Spoof Reporting Message), which covers the injection of falsified messages to deceive a receiving system, and T0830 (Adversary-in-the-Middle), which covers the interception and reuse of legitimate traffic [73, 70]. Both are applicable here: the adversary intercepts a valid proof bundle (T0830) and resubmits it in a subsequent session (T0856) in an attempt to obtain a false friend classification.

Compromised Drone Impersonation A physically captured UAV, or an adversary in possession of extracted credentials, attempts to continue operating as a trusted asset within the swarm [46, 60]. This corresponds to T0849 (Masquerading) [72].

Trust and Trajectory Linking The adversary correlates authentication events across encounters to reconstruct patrol routes, swarm structure, or mission timelines from passively observed protocol traffic [58]. This corresponds to T0842 (Network Sniffing) [71].

Authority and Federation Inference The adversary attempts to identify the issuing authority, coalition partner, or command chain responsible for a UAV by analysing public protocol metadata, revealing sensitive operational or political information [58]. This corresponds to T0842 (Network Sniffing) [71].

Context Confusion Attacks The adversary attempts to reuse valid authorisation proofs outside their intended spatial, temporal, or policy context to induce false trust decisions [49]. This corresponds to T0830 (Adversary-in-the-Middle) and T0856 (Spoof Reporting Message) [70, 73].

Attack	MITRE ICS Technique	ID	Consequence if Successful
Credential replay	Spoof Reporting Message; Adversary-in-the-Middle	T0856; T0830	Adversary authenticates as a legitimate UAV using a replayed proof bundle; false friend classification
Compromised drone impersonation	Masquerading	T0849	Captured UAV continues operating as a trusted asset; mission integrity compromised
Trust & trajectory linking	Network Sniffing	T0842	Patrol routes and swarm structure reconstructed; operational security violated
Authority & federation inference	Network Sniffing	T0842	Issuing authority or coalition identity revealed; political and operational exposure
Context confusion	Adversary-in-the-Middle; Spoof Reporting Message	T0830; T0856	Valid proof accepted in the wrong context; false trust decision triggered

Table 2.2: Mapping of considered attacks to MITRE ATT&CK for ICS techniques and their consequences.

2.4.6. Attacks Out of Scope

The following threats are explicitly excluded from this work.

- Relay (wormhole) attacks. Because (π, ϕ) is a non-interactive bearer token, an adversary can relay a legitimate approaching drone's (AO) fresh proof to the observing drone (OD) without possessing any enrolled credentials. The nonce prevents cross-session reuse but does not bind the proof to a specific sender; mitigations such as distance bounding [17] or channel binding lie outside this design and are identified as an open problem in Section 5.3.
- Sustained jamming and denial-of-service (T0814, T0816). This work focuses on trust establishment under adversarial observation; jamming is addressed by separate electronic warfare countermeasures [68, 69].
- Kinetic attacks or weapons engagement decisions
- Authentication of non-cooperative hostile UAVs
- Real-time command-and-control integrity
- Side channel attacks
- Timing leakage
- Endpoint compromise
- Human-in-the-loop decision processes

2.4.7. Protocol Mitigations

For each attack considered, the protocol addresses the threat through both a general cryptographic principle and a specific design instantiation within R_{IFF}.

Credential Replay *General principle*: session binding via a verifier-supplied challenge prevents proof reuse across encounters. *Design instantiation*: the OD generates ϕ_{nonce} uniformly at random per session; the hardware commitment $\phi_{\mathsf{hw}} = H(H(k_{\mathsf{hw}}) \parallel \phi_{\mathsf{nonce}})$ is bound to this nonce inside the circuit, so a proof valid for one session fails verification against any other nonce. The OD additionally maintains a used-nonce cache to reject repeated values.

Compromised Drone Impersonation *General principle*: authentication based on control state rather than static identity prevents credential reuse after physical capture [53]. *Design instantiation*: Statement 1 requires a valid Merkle membership proof in the live registry; revocation removes the compromised asset's leaf, invalidating its proofs regardless of credential possession. Statement 2 additionally requires knowledge of the hardware-derived key k_{hw}, which is bound to the physical device and produced in volatile memory only.

Trust and Trajectory Linking *General principle*: zero-knowledge proofs reveal no persistent identifiers, making individual encounters unlinkable without auxiliary information [29, 33]. *Design instantiation*: ID_{priv}, k_{hw}, $Sector_{AO}$, and $Role_{AO}$ are private witnesses never transmitted; the public input vector ϕ carries only roots and nonces, which change per session. A passive observer cannot link two encounters to the same asset from protocol traffic alone.

Authority and Federation Inference *General principle*: a shared federation root across all participating nations prevents state-level attribution from public protocol metadata [63]. *Design instantiation*: $\phi_{\mathsf{reg}} = Registry^{*}_{root}$ is a joint root distributed to all observers under a single trusted authority via pre-mission provisioning consistent with the FMN model; it does not partition by nation or command chain. The zero-knowledge property ensures no attribute of the prover's identity or issuing authority is revealed. This property holds only under the architectural assumption of a shared root; per-nation anchors would restore state-level attribution from public inputs.

Context Confusion *General principle*: binding authorisation proofs to a specific policy epoch and session prevents cross-context reuse [49]. *Design instantiation*: Statement 3 enforces $\phi^{\mathsf{cred}}_{\mathsf{epoch}} \in \{\phi_{\mathsf{epoch}} - 1, \phi_{\mathsf{epoch}}\}$ inside the circuit; $\phi_{\mathsf{pol}} = Policy_{root}$ is controlled exclusively by the mission authority and distributed via authenticated out-of-band channels consistent with FMN provisioning [63]. A proof valid under one epoch cannot satisfy the relation under a different epoch without a recomputed credential path against the current $Policy_{root}$.

2.5. Methodology

This thesis follows a constructive research methodology. The primary contribution is a designed artefact, R_{IFF}, whose properties are argued formally and then validated empirically across three stages aligned to the three sub-research questions.

The first stage addresses SRQ1 through design and security analysis. R_{IFF} is specified as a zero-knowledge relation in Chapter 3 and its security properties are argued analytically against the threat model of Section 2.4. The five adversarial attack classes identified there are evaluated using empirical test vectors (T1–T5), which collectively cover the main protocol claims under controlled conditions, while a sixth constructive test (T6) validates the asset binding theorem.

The second stage addresses SRQ2 through empirical measurement. A proof-of-concept (PoC) implementation is evaluated across four operational flight scenarios under controlled network impairment on an x86-64 host and a Raspberry Pi 3B+. Timing and reliability metrics are collected over statistically sufficient sample sizes to support the claims in Chapter 5.

The third stage addresses SRQ3 through a combination of analytical argument and empirical observation. The four multi-observer properties defined in Section 3.4.1 are first argued from the determinism and public verifiability of Groth16, then confirmed empirically across all sweep cycles. Limitations of the evaluation setup, including co-located observers and a simulated RF channel, are stated explicitly in Section 5.3.

3

Solution Design

This chapter presents the proposed privacy-preserving UAV encounter authentication protocol and its implementation. The chapter first defines the formal authentication relation and protocol workflow, then describes the Groth16-based construction and prototype architecture.

3.1. Inspiration

The operational principles of electronic passport control inspire this NIZK-based proof system. In modern border systems [10], an electronic passport is verified along multiple trust layers. First, the issuer's authenticity is confirmed, ensuring that the issuing state is trusted. Second, the passport's integrity is verified, with the chip data linked to the physical document to prevent tampering [25]. Third, authorisation rules are enforced, including visa requirements, travel restrictions, and other local entry regulations [25]. Finally, session freshness is ensured through challenge–response mechanisms, preventing replay attacks from previous encounters [43].

Similarly, in federated mission operations [10], identity validity corresponds to the trusted federation issuing credentials, establishing that an asset is legitimately part of the authorised set. Hardware binding mirrors the chip-level proofs in e-passports, preventing cloning or transfer of credentials between platforms [25]. Policy authorisation corresponds to mission- or sector-specific rules, defining what operations an asset may perform in a given context [25]. Nonce and context binding enforce session freshness, analogous to challenge–response checks at border control, ensuring that each interaction is fresh and cannot be replayed [43].

These layers create a modular trust framework that separates global asset legitimacy, physical device assurance, and mission-scoped operational permissions. This design enables dynamic onboarding and asset revocation in contested environments, while maintaining transmittable, non-interactive zero-knowledge proofs suitable for distributed swarm operations. Unlike e-passports, however, this protocol has no physical presence requirement: the proof tuple is a self-contained bearer token that any intermediary can forward, a structural difference discussed further in Section 5.3.

3.2. Assumptions

The security of the solution construction is based on the following assumptions.

Cryptographic Assumptions

1. The hash function H is collision-resistant and efficiently computable in circuit.
2. The underlying NIZK proof system satisfies completeness, knowledge soundness, and zero knowledge.
3. The Common Reference String CRS is generated honestly, or via a secure multi-party computation such that no adversary obtains trapdoor information.

System Model Assumptions

1. The observing drone (OD) maintains authenticated local copies of $Registry^*_{root}$, $Policy^*_{root}$, and the verification key `vk`. The per-asset hardware enrolment registry is used only in the modular `all` mode and is not provisioned to the OD in the deployed joint mode.
2. These values are updated only via authenticated out-of-band channels, which the prover cannot influence.
3. The nonce ϕ_{nonce} is generated uniformly at random by the OD and is never reused.

Hardware Assumptions

1. The device contains a PUF that produces sufficiently high-entropy responses. A hardware-derived key k_{hw} is generated in volatile memory at attestation time and is never persisted. The error-correction mechanism used to derive k_{hw} is assumed not to leak information that materially reduces its secrecy as a witness.
2. During enrolment, the reference value $h^{\mathsf{enrol}}_{\mathsf{hw}} = H(k^{\mathsf{enrol}}_{\mathsf{hw}})$ is stored in the OD registry, while the raw key is discarded. Storing only the hash prevents direct recovery of the hardware secret from a registry compromise.
3. The AO computes the session-bound value $\phi_{\mathsf{hw}} = H(H(k_{\mathsf{hw}}) \parallel \phi_{\mathsf{nonce}})$ using its live hardware key and the OD-provided nonce, and includes it in the public input vector ϕ. This follows [53], where a hardware-bound commitment is checked against an enrolled reference.
4. The hardware binding is enforced within the circuit: $h^{\mathsf{enrol}}_{\mathsf{hw}}$ is a private witness value, and the circuit enforces $H(k_{\mathsf{hw}}) = h^{\mathsf{enrol}}_{\mathsf{hw}}$ internally. The OD neither retrieves $h^{\mathsf{enrol}}_{\mathsf{hw}}$ nor computes any external check on ϕ_{hw}; the binding is established entirely within the proof.
5. The OD publishes a rolling two-epoch policy window. The root ϕ_{pol} at epoch e contains credential leaves for epochs e and $e-1$; credentials from epoch $e-2$ and earlier are excluded and cannot be verified. The circuit enforces the validity condition $(\mathsf{epoch}_{\mathsf{cred}} - \phi_{\mathsf{epoch}})(\mathsf{epoch}_{\mathsf{cred}} - \phi_{\mathsf{epoch}} + 1) = 0$, but correctness of tree rotation is assumed from the OD and not verified inside the circuit.
6. Physical extraction of the hardware secret before revocation is out of scope.
7. Compromised devices are revoked via registry updates distributed to all observers.
8. A one-epoch grace period allows credentials from epoch $e-1$ to remain valid until epoch e expires. This mitigates propagation delays in contested environments; the acceptance window is bounded and deployment-defined.

Physical capture followed by extraction of the hardware secret breaks the binding guarantee. Recovery depends entirely on timely revocation via the registry.

Trust Boundary The zero-knowledge proof establishes that the prover knows witnesses that satisfy the relation with respect to the public inputs. It guarantees that private values such as identities, hardware responses, and authorisation attributes are not revealed to the verifier. The proof does not guarantee that the values are freshly generated by hardware at the time of attestation, nor does it protect against side channels outside the relation. Relay attacks are structurally possible within any non-interactive bearer-token design and are discussed in Section 5.3.

3.3. Non-Interactive Zero-Knowledge Relation

This section specifies the NP relation R_{IFF} that is proven in zero knowledge. The prover demonstrates that it simultaneously satisfies three conditions. It proves that it is a valid member of the federation, that it can reconstruct a hardware-bound secret consistent with enrolment, and that it is authorised under the currently active policy epoch. These conditions correspond to three distinct trust layers and are enforced jointly in a single proof.

3.3.1. Notation

Symbol	Meaning
ϕ	Public input vector $(\phi_{\mathsf{reg}}, \phi_{\mathsf{nonce}}, \phi_{\mathsf{epoch}}, \phi_{\mathsf{hw}}, \phi_{\mathsf{pol}})$
w	Private witness vector $(w_{\mathsf{reg}}, w_{\mathsf{hw}}, w_{\mathsf{pol}})$
π	NIZK proof
CRS	Common Reference String
R_{IFF}	NP relation proven by π
$H(\cdot)$	Collision-resistant hash function
k_{hw}	Live hardware-derived key
$h_{\mathsf{hw}}^{\mathsf{enrol}}$	Hash of the enrolled hardware reference key, $H(k_{\mathsf{hw}}^{\mathsf{enrol}})$, stored in the OD registry
ℓ	Registry leaf constructed at enrolment: $H(H(ID_{priv}) \parallel h_{\mathsf{hw}}^{\mathsf{enrol}})$
VerifyMembership	Merkle path verification against the federation registry root
VerifyAuthorisation	Merkle path verification against the policy tree root
$\parallel$	Concatenation

Table 3.1: Notation Used in the NIZK Specification.

3.3.2. Common Reference String

The protocol operates in the common reference string (CRS) model. A structured reference string $\mathsf{CRS} = (\mathsf{pk}, \mathsf{vk})$ is generated before deployment. The proving key pk is used by the prover and the verification key vk is used by the verifier. The CRS is not transmitted during attestation. The OD pins vk at deployment and rejects any session where it does not match the expected value.

The public inputs ϕ_{reg}, ϕ_{pol}, and ϕ_{hw} are not self-certifying and therefore require trusted anchors or internal consistency checks. The OD maintains local anchor values $Registry^*_{root}$ and $Policy^*_{root}$, provisioned at deployment, and verifies $\phi_{\mathsf{reg}} \stackrel{?}{=} Registry^*_{root}$ and $\phi_{\mathsf{pol}} \stackrel{?}{=} Policy^*_{root}$ before invoking the verifier. In contrast, the hardware binding is enforced entirely within the circuit: $h_{\mathsf{hw}}^{\mathsf{enrol}}$ is a private witness value, and the equality between k_{hw} and the enrolled reference is verified by the circuit itself. The OD neither retrieves $h_{\mathsf{hw}}^{\mathsf{enrol}}$ nor maintains a per-asset identity hypothesis. Instead, the proof establishes that the AO possesses the hardware secret corresponding to the enrolled reference without revealing the secret itself.

3.3.3. Public and Private Inputs

Public inputs The OD constructs the partial public input vector $\{\phi_{\mathsf{reg}}, \phi_{\mathsf{nonce}}, \phi_{\mathsf{epoch}}, \phi_{\mathsf{pol}}\}$ and transmits it to the AO. The AO appends ϕ_{hw}, computed from its live hardware key and the received nonce, to form the complete public input vector

$$\phi = \left\{\phi_{\mathsf{reg}},\ \phi_{\mathsf{nonce}},\ \phi_{\mathsf{epoch}},\ \phi_{\mathsf{hw}},\ \phi_{\mathsf{pol}}\right\}$$

which is used as input to both proof generation and verification.

Input	Description
$\phi_{\text{reg}} = Registry_{root}$	Root of the federation registry. Anchors the global set of valid assets and enables revocation by removing leaves from the tree. Supplied by the OD.
ϕ_{nonce}	Fresh random challenge generated per session by the OD. Binds the proof to a specific interaction, prevents replay, and acts as the hardware challenge to the AO's PUF.
ϕ_{epoch}	Active policy epoch identifier. Explicitly bound into the authorisation statement to prevent cross-epoch reuse. Supplied by the OD.
ϕ_{hw}	Session-bound hardware commitment $H(H(k_{\text{hw}}) \parallel \phi_{\text{nonce}})$, computed by the AO from its PUF-derived key in response to ϕ_{nonce} and returned to the OD as part of ϕ. In the joint circuit, this value is constrained internally to both k_{hw} and the enrolled reference; the OD does not verify it outside the proof.
$\phi_{\text{pol}} = Policy_{root}$	Merkle root of all authorised attribute tuples for the active policy epoch. Enables membership proofs without revealing the underlying attributes. Supplied by the OD.

Table 3.2: Public Inputs and Their Security Role

Private witnesses The AO holds the witness vector $w = \{w_{\text{reg}}, w_{\text{hw}}, w_{\text{pol}}\}$ which is never transmitted.

Witness	Description
$w_{\text{reg}} = (ID_{priv}, Path_{\text{reg}})$	Private identity and its Merkle authentication path to ϕ_{reg}. The registry leaf is $\ell = H(H(ID_{priv}) \parallel h_{\text{hw}}^{\text{enrol}})$, binding both the identity and the enrolled hardware reference to the same leaf, ensuring that S1 and S2 are proven for the same enrolled asset within the circuit.
$w_{\text{hw}} = (k_{\text{hw}})$	Hardware-derived key produced on-device at attestation time in response to ϕ_{nonce}. The circuit checks $H(H(k_{\text{hw}}) \parallel \phi_{\text{nonce}}) = \phi_{\text{hw}}$ and enforces $H(k_{\text{hw}}) = h_{\text{hw}}^{\text{enrol}}$, binding the live key to the enrolled reference committed in the leaf of S1. Never stored persistently.
$w_{\text{pol}} = (Sector_{AO}, Role_{AO}, Path_{\text{pol}})$	Attribute values and their Merkle path to ϕ_{pol}. The attribute tuple is bound to the active epoch inside the circuit to ensure correct authorisation semantics.

Table 3.3: Private Witnesses and Their Role in the Proof

3.3.4. The NIZK Relation

$$\begin{aligned} R_{\text{IFF}} = \{(\phi, w) \mid & \text{Joint\_Valid}(w_{\text{reg}}, w_{\text{hw}}, \phi_{\text{reg}}) \\ & \wedge \text{HW\_Valid}(w_{\text{hw}}, \phi_{\text{nonce}}, \phi_{\text{hw}}) \\ & \wedge \text{Policy\_Valid}(w_{\text{pol}}, \phi_{\text{epoch}}, \phi_{\text{pol}}) \ \} \end{aligned} \tag{3.1}$$

Each conjunct corresponds to a distinct trust layer, and all must hold simultaneously for a valid proof. The first conjunct, Joint_Valid, verifies registry membership against a leaf that encodes both the identity and the enrolled hardware reference, ensuring both are proven for the same enrolled asset. The second conjunct, HW_Valid, is not redundant with this: it enforces that the live hardware key k_{hw} produces the correct session commitment under the OD-supplied nonce, and that $H(k_{\text{hw}})$ equals the same $h_{\text{hw}}^{\text{enrol}}$ wire used in the leaf computation of Joint_Valid. The two conjuncts are thus interdependent: Joint_Valid

provides the registry-level binding and HW_Valid provides the session-level binding, with $h_{\mathsf{hw}}^{\mathsf{enrol}}$ as the shared internal wire that ties them together.

Statement 1: Membership and Hardware Binding The prover demonstrates that its identity and hardware reference are jointly enrolled in the federation registry:

$$\mathsf{VerifyMembership}\big(H(H(ID_{priv}) \parallel h_{\mathsf{hw}}^{\mathsf{enrol}}),\ Path_{\mathsf{reg}},\ \phi_{\mathsf{reg}}\big) = \mathsf{True},$$

where $h_{\mathsf{hw}}^{\mathsf{enrol}}$ is treated as an internal circuit wire whose value is enforced by the second constraint in Statement 2 to equal $H(k_{\mathsf{hw}})$, binding the enrolment-time reference to the live hardware key. The leaf $\ell = H(H(ID_{priv}) \parallel h_{\mathsf{hw}}^{\mathsf{enrol}})$ was constructed at enrolment and committed into the registry tree by the trusted authority. Revocation removes the leaf from the registry, simultaneously invalidating both the identity and hardware binding of the revoked asset.

Remark 1. *The leaf encodes $h_{\mathsf{hw}}^{\mathsf{enrol}} = H(k_{\mathsf{hw}}^{\mathsf{enrol}})$ rather than the raw key, so a compromise of the registry tree does not directly expose the underlying hardware secret.*

Statement 2: Enrolment-Bound Hardware Secret The prover demonstrates knowledge of a hardware-derived secret that is consistent with the enrolled reference and bound to the current session nonce. This statement realises a zero-knowledge challenge-response: the OD issues ϕ_{nonce} as a hardware challenge, and the AO responds by deriving k_{hw} from its PUF and producing a proof of knowledge of the correct response, following the approach of [53]. The circuit enforces two constraints jointly:

$$H(H(k_{\mathsf{hw}}) \parallel \phi_{\mathsf{nonce}}) \stackrel{?}{=} \phi_{\mathsf{hw}}$$

$$H(k_{\mathsf{hw}}) \stackrel{?}{=} h_{\mathsf{hw}}^{\mathsf{enrol}}$$

The first constraint binds the hardware key to the current session via the nonce. The second constraint enforces that k_{hw} is consistent with the enrolled reference committed in the registry leaf of Statement 1, closing the cross-asset substitution attack. In the joint relation, $h_{\mathsf{hw}}^{\mathsf{enrol}}$ is a private witness: the equality $H(k_{\mathsf{hw}}) = h_{\mathsf{hw}}^{\mathsf{enrol}}$ is enforced inside the circuit against the value committed in the registry leaf of Statement 1, so the OD performs no external check on ϕ_{hw}.

Remark 2. *Statement 2 proves possession of a hardware-derived secret consistent with the enrolled commitment: it does not attest hardware integrity, untampered firmware, secure boot, runtime state, or a trusted execution environment, nor that k_{hw} was freshly derived from the physical PUF at attestation time. We term this* enrolment-bound secret possession *and reserve* hardware integrity attestation *for the strictly stronger notion this protocol does not provide; freshness and integrity are platform-level properties outside the relation, following the separation of concerns in [53]. What Statement 2 adds is the binding of $h_{\mathsf{hw}}^{\mathsf{enrol}}$ as a wire shared with Statement 1's leaf computation, so that the same value is proven consistent with both the registry membership and the live hardware key within a single witness (Theorem 3.1).*

Statement 3: Policy Authorisation The prover demonstrates that it is authorised under the active policy epoch:

$$\mathsf{VerifyAuthorisation}\big(H(Sector_{AO} \parallel Role_{AO} \parallel \phi_{\mathsf{epoch}}^{\mathsf{cred}}),\ Path_{\mathsf{pol}},\ \phi_{\mathsf{pol}}\big) = \mathsf{True},$$

where the circuit additionally enforces $\phi_{\mathsf{epoch}}^{\mathsf{cred}} \in \{\phi_{\mathsf{epoch}} - 1,\ \phi_{\mathsf{epoch}}\}$. The epoch value $\phi_{\mathsf{epoch}}^{\mathsf{cred}}$ is the epoch encoded in the credential and is part of the private witness. The public input ϕ_{epoch} is the current epoch supplied by the OD. A one-epoch tolerance accommodates delayed credential propagation in contested environments while rejecting credentials older than one epoch. The security of this statement relies on $\phi_{\mathsf{pol}} = Policy_{root}$ being a trusted anchor controlled exclusively by the mission authority. An adversary cannot insert a forged attribute tuple into the tree without controlling the root, which is distributed to all observers via authenticated out-of-band channels before the mission. This statement proves that the asset operates within an authorised sector and role while keeping both values hidden from the verifier. The one-epoch tolerance is a deliberate design choice scoped to the current threat model (Section 2.4); a tighter freshness guarantee is left for future work.

Remark 3. *Although $\phi^{\mathsf{cred}}_{\mathsf{epoch}}$ is part of the private witness and the prover can choose its claimed value, a valid proof cannot be constructed by setting $\phi^{\mathsf{cred}}_{\mathsf{epoch}} = \phi_{\mathsf{epoch}}$ for a stale credential. Proof validity requires a Merkle path from $H(Sector_{AO} \parallel Role_{AO} \parallel \phi^{\mathsf{cred}}_{\mathsf{epoch}})$ to the current root $\phi_{\mathsf{pol}} = Policy_{root}$.*

An $e-2$ credential is not included in ϕ_{pol}, which contains only leaves for epochs e and $e-1$. Its leaf therefore cannot produce a valid path to the current $Policy_{root}$, regardless of the claimed epoch value. The phrase "committed under a different root" refers to a stale, rotated-out root under which the credential was originally inserted.

Security thus flows through ϕ_{pol} as a trusted anchor controlled by the mission authority, as stated in the System Model Assumptions of Section 3.2.

3.3.5. Protocol Workflow

Algorithm 1 initialises the federation binding between identity, hardware commitment, and registry membership. During enrolment, the Federated Mission Networking authority (FMN) provisions the cryptographic material required for subsequent proof generation and verification.

Algorithm 1 Enrolment

Require: Device d, FMN authority $\mathcal{F}$, federation registry $\mathcal{R}$
Ensure: Registry leaf ℓ committed to $\mathcal{R}$
1: $k^{\mathsf{enrol}}_{\mathsf{hw}} \leftarrow \mathsf{PUF}(d)$
2: $h^{\mathsf{enrol}}_{\mathsf{hw}} \leftarrow H(k^{\mathsf{enrol}}_{\mathsf{hw}})$
3: **erase** $k^{\mathsf{enrol}}_{\mathsf{hw}}$
4: $ID_{priv} \xleftarrow{\$} \{0,1\}^{\lambda}$
5: $\ell \leftarrow H(H(ID_{priv}) \parallel h^{\mathsf{enrol}}_{\mathsf{hw}})$
6: Insert ℓ into federation registry tree $\mathcal{R}$
7: $Registry_{root} \leftarrow \mathsf{UpdateRoot}(\mathcal{R})$
8: $Path_{\mathsf{reg}} \leftarrow \mathsf{GetMerklePath}(\mathcal{R}, \ell)$
9: $\mathcal{F}$ provisions to device d:
$(ID_{priv}, Path_{\mathsf{reg}}, \mathsf{pk})$
10: $\mathcal{F}$ provisions to observers:
$(Registry^{*}_{root}, Policy^{*}_{root}, \mathsf{vk})$

Algorithm 2 generates a session-bound NIZK proof attesting to valid federation membership, possession of an enrolment-bound hardware secret, and policy authorisation.

Algorithm 2 Proof Generation (AO side)

Require: Challenge tuple $(\phi_{\mathsf{reg}}, \phi_{\mathsf{nonce}}, \phi_{\mathsf{epoch}}, \phi_{\mathsf{pol}})$
Require: Private witness material $(ID_{priv}, Path_{\mathsf{reg}}, Sector_{AO}, Role_{AO}, Path_{\mathsf{pol}}, \mathsf{pk})$
Ensure: Proof tuple (ϕ, π)
1: $k_{\mathsf{hw}} \leftarrow \mathsf{PUF}(d)$
2: $\phi_{\mathsf{hw}} \leftarrow H(H(k_{\mathsf{hw}}) \parallel \phi_{\mathsf{nonce}})$
3: $\phi \leftarrow (\phi_{\mathsf{reg}},\ \phi_{\mathsf{nonce}},\ \phi_{\mathsf{epoch}},\ \phi_{\mathsf{hw}},\ \phi_{\mathsf{pol}})$
4: $w \leftarrow (ID_{priv},\ Path_{\mathsf{reg}},\ k_{\mathsf{hw}},\ Sector_{AO},\ Role_{AO},\ Path_{\mathsf{pol}})$
5: **if** $\mathsf{Satisfied}(R_{\mathsf{IFF}}, \phi, w) \neq \mathsf{True}$ **then**
6: **abort**
7: **end if**
8: $\pi \leftarrow \mathsf{Prove}(\mathsf{pk}, \phi, w)$
9: **erase** k_{hw}
10: **send** (ϕ, π) to OD

The satisfiability pre-check is a lightweight local validation step performed before Groth16 proof generation to avoid unnecessary prover execution on malformed witness assignments.

Algorithm 3 verifies the received proof tuple against locally trusted anchor state and freshness constraints. Lightweight deterministic checks are performed first to reject malformed or stale tuples without invoking pairing operations.

Algorithm 3 Verification (OD side)

Require: Received tuple (ϕ, π)
Require: Local anchors $(Registry^*_{root}, Policy^*_{root}, \mathsf{vk})$
Require: Nonce cache $\mathcal{N}$
Ensure: Trust classification $\in \{\mathsf{trusted}, \mathsf{unknown}\}$
if $\phi_{\mathsf{reg}} \neq Registry^*_{root}$ **then**
  return unknown
end if
if $\phi_{\mathsf{pol}} \neq Policy^*_{root}$ **then**
  return unknown
end if
if $\phi_{\mathsf{nonce}} \in \mathcal{N}$ **then**
  return unknown
end if
if $\phi_{\mathsf{epoch}} \notin \mathsf{ActiveEpochs}()$ **then**
  return unknown
end if
if $\mathsf{Verify}(\mathsf{vk}, \phi, \pi) \neq \mathsf{True}$ **then**
  return unknown
end if
$\mathcal{N} \leftarrow \mathcal{N} \cup \{\phi_{\mathsf{nonce}}\}$
Disseminate (ϕ, π) to neighbouring observers
return trusted

Nonce cache entries are retained for the active epoch window and discarded upon epoch expiration.

Figure 3.1 summarises the complete protocol workflow. During pre-deployment, the FMN distributes the proving key to enrolled devices and the verification key and anchor values to all observers. During an encounter, the OD issues a fresh nonce challenge to the AO. The AO derives a session commitment, constructs the witness vector, and generates a Groth16 proof over the relation R_{IFF}. The OD performs lightweight deterministic anchor checks before invoking cryptographic verification. Accepted tuples may subsequently be disseminated to neighbouring observers for independent re-verification.

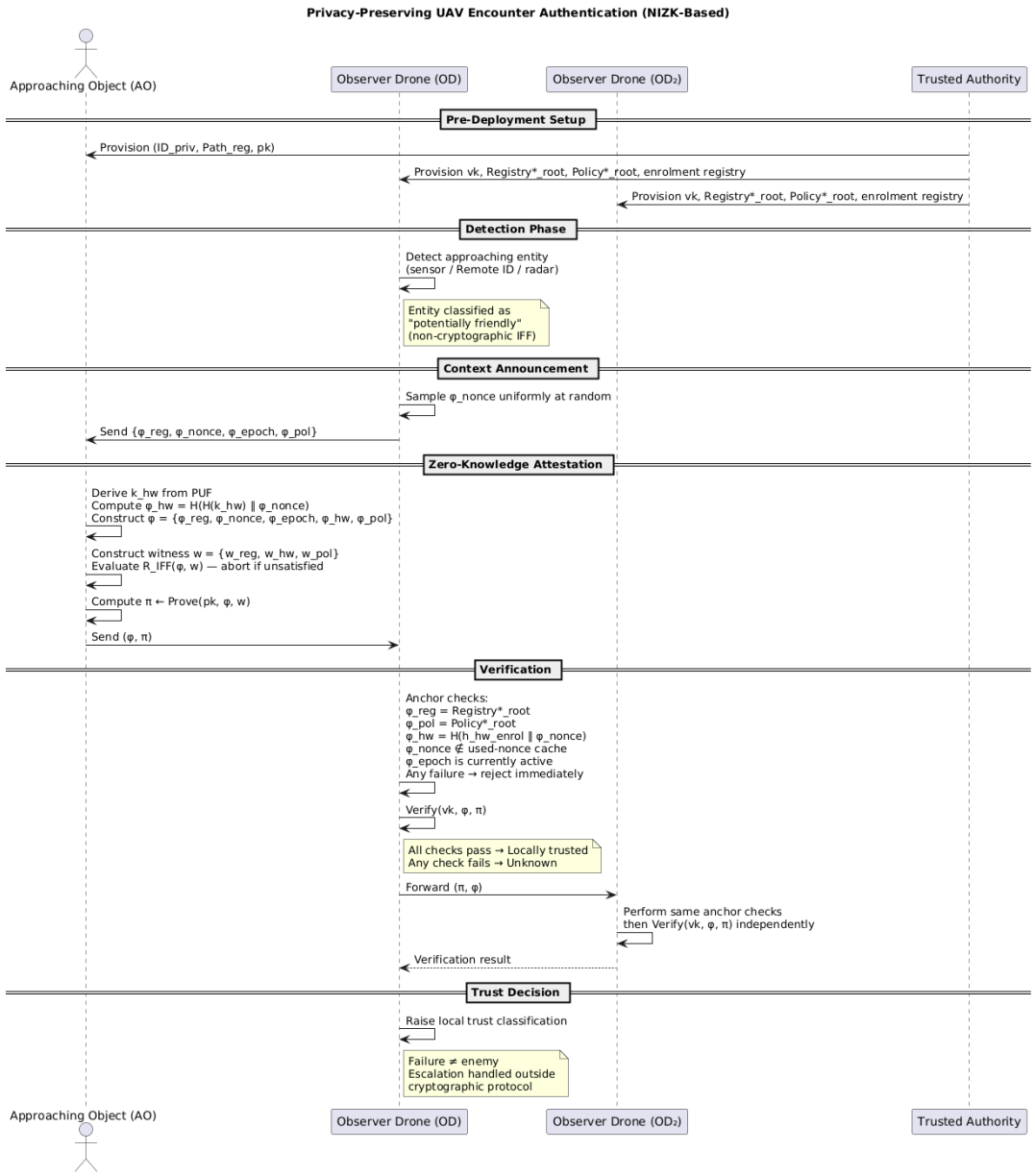

Figure 3.1: Protocol workflow for privacy-preserving UAV encounter authentication. The OD issues a nonce challenge, the AO generates a NIZK proof over the current session state, and observers independently verify the forwarded proof tuple against local anchor values.

3.4. Multi-Observer Verification and Proof Forwarding

Because the proof is non-interactive and bound to the public input vector ϕ, it can be forwarded between observers without requiring recomputation by the prover. An observer verifies π against its locally trusted anchors. The tuple (π, ϕ) may then be forwarded to other observers, each of which independently performs the same verification. Each secondary observer must hold authenticated local copies of $Registry^*_{root}$, $Policy^*_{root}$, and the nonce cache to perform the nonce and epoch anchor checks alongside proof verification. This enables distributed trust establishment in environments where connectivity is intermittent or adversarial without requiring any single observer to be globally trusted.

3.4.1. Required Properties for Multi-Observer Verification

For multi-observer verification to improve trust decisions over single-observer authentication in UAV swarm encounters, the protocol must satisfy four properties. These properties are stated formally below,

and Section 5.1.3 argues whether the proposed design satisfies each one analytically and empirically.

Consistency All observers holding the same trusted anchors and verification key vk must reach identical binary trust decisions when presented with the same proof tuple (π, ϕ). As established in Section 2.2.3, this follows from the determinism of $\mathsf{Verify}(\mathsf{vk}, \phi, \pi)$: any two parties invoking the verifier on identical inputs will produce identical outputs without coordination. Inconsistent verdicts across the swarm would undermine coordinated decision-making and could be exploited to selectively deceive subsets of observers.

Locality Each observer must be capable of performing verification independently using only locally held state — the anchor values $Registry^*_{root}$ and $Policy^*_{root}$ and the pinned verification key vk — without querying any external service or communicating with the original prover at verification time. This property is essential for operation in infrastructure-denied environments where connectivity to ground infrastructure cannot be assumed, and is a direct consequence of the public verifiability property of NIZK proofs established in Section 2.2.2 [33].

Forwarding Correctness A proof tuple (π, ϕ) forwarded from a primary observer to a secondary observer must verify correctly at the secondary observer without modification, provided both hold the same trusted anchors. The forwarded proof must carry the same trust guarantees as a directly verified proof, and the forwarding process must not introduce new trust assumptions. This property holds because π is a self-contained certificate of the statement $R_{\mathsf{IFF}}(\phi, w) = 1$, and its validity is independent of the communication path through which it was received [13].

Availability Under Degraded Connectivity The protocol must produce usable trust decisions even when communication conditions are degraded, specifically under elevated one-way delay and non-zero packet loss representative of contested wireless environments. A single observer completing verification and forwarding the result to secondary observers must provide a trust decision that remains valid for the full swarm, reducing the dependency on direct prover reachability. This property is motivated by the operational context of Section 2.2.6 and is evaluated empirically under simulated degraded conditions in Section 5.1.3.

3.5. Security Analysis

This section analyses the security properties of the proposed protocol with respect to R_{IFF} and the attacks identified in Section 2.4. Properties are organised into two categories: those inherited from the underlying proof system, which hold by instantiation, and those specific to the protocol design, which require an argument with respect to the protocol structure and assumptions of Section 3.2.

3.5.1. Inherited Properties

The following properties hold by instantiation of the Groth16 proof system [33] with R_{IFF}, under the q-Power Knowledge of Exponent assumption and the q-Strong Diffie-Hellman assumption in the generic group model. No additional argument is required beyond correctly specifying the relation, which is done in Section 3.3.

Completeness If the AO holds a valid witness w satisfying $R_{\mathsf{IFF}}(\phi, w) = 1$, that is, it is a registered and non-revoked federation member whose hardware key is consistent with the enrolled reference and whose policy credential is valid under the active epoch, then $\mathsf{Prove}(\mathsf{pk}, \phi, w)$ produces a proof π that $\mathsf{Verify}(\mathsf{vk}, \phi, \pi)$ accepts with overwhelming probability. This ensures that legitimate assets are not falsely rejected.

Knowledge Soundness Under Definition 3, no computationally bounded adversary can produce an accepting proof without simultaneously holding a private identity consistent with a leaf of $Registry_{root}$, a hardware key consistent with ϕ_{hw}, and a policy credential consistent with $Policy_{root}$. Partial credential possession is therefore insufficient to address the compromised drone impersonation attack (T0849) directly. This guarantee holds under the joint circuit construction described in Chapter 4. Statement 2

additionally relies on preimage resistance of H: observing $h_{\mathsf{hw}}^{\mathsf{enrol}}$ in the enrolment registry does not yield k_{hw}, and second-preimage resistance follows from the collision resistance assumed in Section 3.2.

Zero Knowledge By the zero-knowledge property of Groth16 [33], ID_{priv}, k_{hw}, $Sector_{AO}$, and $Role_{AO}$ remain computationally hidden from any observer holding only (ϕ, π, vk). Session unlinkability, within federation, follows from the freshness of ϕ_{nonce} and the resulting session-bound commitment ϕ_{hw}. A passive observer can identify the federation and epoch from the public inputs ϕ_{reg} and ϕ_{pol}, but cannot attribute a proof to a specific asset. The OD learns only that the prover is a valid, non-revoked member under the active epoch. The private attributes remain hidden, and consecutive proofs from the same asset cannot be linked through the proof itself. This unlinkability holds against any observer that does not hold the enrolment registry. A registry-holder could test a captured ϕ_{hw} against enrolled references, but in the deployed joint mode the OD is not provisioned with the enrolment registry (Section 3.2) and thus cannot perform this test. The capability is limited to the enrolling authority, which necessarily already knows asset identities; it is not available to the verifier. Any association between a proof and a physical airframe therefore arises from the non-cryptographic sensing channel described in Section 2.4.1, rather than from the proof.

3.5.2. Protocol-Specific Properties

The following properties are specific to the design of R_{IFF} and the surrounding protocol. They are stated as claims with supporting arguments; formal proofs via automated verification tools are left as future work.

Claim 3.5.1 (Nonce-Bound Replay Protection)**.** *A proof π generated under nonce $\phi_{\mathsf{nonce}}^{(1)}$ cannot be accepted by a verifier presenting nonce $\phi_{\mathsf{nonce}}^{(2)} \neq \phi_{\mathsf{nonce}}^{(1)}$, except with negligible probability at most $\mu(\lambda)$.*

Argument. In the joint circuit the OD performs no ϕ_{hw} anchor check; acceptance rests on the Groth16 verification and the used-nonce cache. Since Groth16 binds every public input and ϕ_{nonce} is one, π (produced under $\phi_{\mathsf{nonce}}^{(1)}$) satisfies the verification equation only for $\phi_{\mathsf{nonce}}^{(1)}$; presenting it under $\phi_{\mathsf{nonce}}^{(2)} \neq \phi_{\mathsf{nonce}}^{(1)}$ fails except with the knowledge-soundness error $\mu(\lambda)$. Presenting the matched pair $(\pi, \phi_{\mathsf{nonce}}^{(1)})$ instead passes Groth16 but is rejected by the freshness check, as $\phi_{\mathsf{nonce}}^{(1)}$ is not the issued challenge $\phi_{\mathsf{nonce}}^{(2)}$. Thus binding here follows from SNARK public-input binding, not collision resistance of H, under the assumptions of Section 3.2. □

Claim 3.5.2 (Context Binding)**.** *A proof π valid under policy epoch ϕ_{epoch} and roots $Registry_{root}$ and $Policy_{root}$ cannot satisfy R_{IFF} under a different epoch ϕ'_{epoch} or different roots without a recomputed witness, except with negligible probability at most $\mu(\lambda)$.*

Argument. The circuit enforces $\phi_{\mathsf{epoch}}^{\mathsf{cred}} \in \{\phi_{\mathsf{epoch}} - 1, \phi_{\mathsf{epoch}}\}$, binding the credential epoch to the verifier-supplied epoch identifier. The policy root $\phi_{\mathsf{pol}} = Policy_{root}$ is verified as a trusted anchor before proof evaluation; a forged or substituted root causes immediate rejection. An adversary wishing to reuse a valid proof under a different epoch would need to find a credential path that verifies against a different $Policy_{root}$, which requires either control of the mission authority or breaking the collision resistance of H. This directly addresses the context confusion attack (T0830, T0856). The one-epoch tolerance is a deliberate operational trade-off introducing a bounded acceptance window; this is acknowledged as a limitation in Section 5.3. □

Claim 3.5.3 (Asset-Level Anonymity Within a Shared Federation)**.** *Under a shared $Registry_{root}$ and $Policy_{root}$ distributed across all participating nations, no observer can determine the issuing authority, national affiliation, or command chain of the prover from the public inputs ϕ or the proof π, except with negligible probability at most $\mu(\lambda)$. In the deployed joint-circuit mode, the hardware secret is additionally protected at the relation level as a witness-only constraint, removing any plaintext exposure path beyond the zero-knowledge guarantee.*

Argument. By the zero-knowledge property, π reveals nothing about w beyond $R_{\mathsf{IFF}}(\phi, w) = 1$. The public inputs expose only $\phi_{\mathsf{reg}} = Registry_{root}$ and $\phi_{\mathsf{pol}} = Policy_{root}$, which are shared identically across all federation members regardless of national origin. No per-nation partitioning of the roots is present, so no state-level attribution is possible from ϕ alone. A global passive adversary observing multiple

proof tuples can identify that two proofs originate from the same federation and epoch, but cannot attribute either proof to a specific asset. In the operational context of this work, federation membership is not treated as a secret; instead, the existence of a coalition drone programme is assumed to be known to the adversary. The confidentiality claims concern asset-level attributes: identity, hardware binding, sector, and role, all of which remain protected by the zero-knowledge property. This claim holds only under the architectural assumption of a jointly managed shared root; per-nation trust anchors would restore state-level attribution and violate this claim. □

Theorem 3.1 (Asset Binding)**.** *Assume Groth16 knowledge soundness for R_{IFF} [33] and collision resistance of H. Assume further that k_{hw}^A and k_{hw}^B are independently sampled at enrolment with sufficient entropy such that $H(k_{\mathsf{hw}}^A) \neq H(k_{\mathsf{hw}}^B)$ except with negligible probability, as ensured by the PUF entropy assumption in Section 3.2. Then no PPT adversary, given enrolled credential material for two distinct assets $A \neq B$, specifically ID_{priv}^A and k_{hw}^B, can produce an accepting proof π for R_{IFF} under any statement ϕ, except with negligible probability.*

Proof. Assume an adversary $\mathcal{A}$ with ID_{priv}^A and k_{hw}^B for $A \neq B$ produces an accepting (ϕ, π). By Groth16 knowledge soundness (Definition 3), there exists an extractor $\mathcal{E}$ that returns a witness w such that $R_{\mathsf{IFF}}(\phi, w) = 1$ except with negligible probability.

Since R_{IFF} is a single R1CS enforcing both registry membership and hardware equality, the same value $h_{\mathsf{hw}}^{\mathsf{enrol}}$ must be used in both the leaf construction $\ell = H(H(ID_{priv}) \parallel h_{\mathsf{hw}}^{\mathsf{enrol}})$ and the constraint $H(k_{\mathsf{hw}}) = h_{\mathsf{hw}}^{\mathsf{enrol}}$. The witness must therefore correspond to a valid leaf in the registry root ϕ_{reg}.

We consider the possible origins of such a leaf.

Case 1: leaf of asset A. Matching $\ell^A = H(H(ID_{priv}^A) \parallel H(k_{\mathsf{hw}}^A))$ requires a value k'_{hw} such that $H(k'_{\mathsf{hw}}) = H(k_{\mathsf{hw}}^A)$. Since only k_{hw}^B is available and $k_{\mathsf{hw}}^A \neq k_{\mathsf{hw}}^B$, this implies a collision in H.

Case 2: leaf of asset B. Matching $\ell^B = H(H(ID_{priv}^B) \parallel H(k_{\mathsf{hw}}^B))$ requires ID_{priv}^B, which is not available to $\mathcal{A}$. To satisfy the equation without it, $\mathcal{A}$ would need to supply some $\mathsf{id} \neq ID_{priv}^B$ such that $H(H(\mathsf{id}) \parallel H(k_{\mathsf{hw}}^B)) = \ell^B$, which is a collision in H.

Case 3: any other leaf. Any non-enrolled or third-asset leaf has no valid Merkle path to ϕ_{reg}, so membership would require a colliding path in H.

Thus in all cases, acceptance implies a collision in H, contradicting collision resistance.

The result matches the empirical observation in Section 4.3.8 (T6): independently valid identity and hardware proofs can be recombined in the modular `all` mode, which lacks a shared-wire constraint, but are rejected by the joint circuit R_{IFF}. The mechanism itself is standard witness binding; the contribution is the observation that decomposing R_{IFF} into separately proven statements forfeits this binding in coalition settings where identity and hardware credentials originate from different custodians. □

4

Implementation

This chapter describes the proof-of-concept (PoC) implementation of R_{IFF} and the experimental evaluation conducted to validate the protocol against the research questions. The implementation comprises three layers: a C++ circuit library instantiating the four statements of Section 3.3 using the Groth16 proof system, a Python orchestration layer that drives the authentication protocol over simulated UAV telemetry, and a scenario-driven test setup that applies controlled network conditions across four operational flight scenarios. These layers demonstrate that the protocol produces correct trust decisions, meets latency requirements on the evaluation host, and degrades gracefully under network impairment. All timing measurements were obtained on a single x86-64 host (WSL2, Linux 6.6); ARM embedded hardware evaluation on a Raspberry Pi platform is reported in Section 4.4.[1]

4.1. System Architecture

The PoC adopts FlyNetSim's [7] application-layer messaging protocol, the ZeroMQ (ZMQ) topic prefix scheme (`@@@U_`, `@@@G_`), telemetry message structure, and control command vocabulary, as the communication substrate. The ArduPilot SITL flight-dynamics and ns-3 network-simulation layers of FlyNetSim are not used; UAV kinematics are derived from parameterised scenario profiles, and network impairment is emulated at the application layer, as described in Section 4.3.1.

The ZMQ telemetry bus is implemented as an XSUB/XPUB proxy: all publishers (synthetic telemetry generator, Authentication Bridge) connect to the XSUB frontend; all subscribers connect to the XPUB backend. This architecture allows multiple processes to publish on the same logical channel and is required for strict-network-path mode, in which the OD process must receive its own published challenge message back from the bus rather than via an in-process function call.

In the PoC, consistency of ϕ_{reg} and ϕ_{pol} with enrolled values is enforced by the circuit constraints rather than as a separate application-layer pre-check; Algorithm 3 describes the full pre-check sequence as it would operate in a deployed system.

Figure 4.1 shows the system component overview. The runtime roles map to the formal roles of Section 3.3 as follows.

[1] The full implementation, experimental harness, and raw data are available at `https://github.com/emasujster/RIFF`.

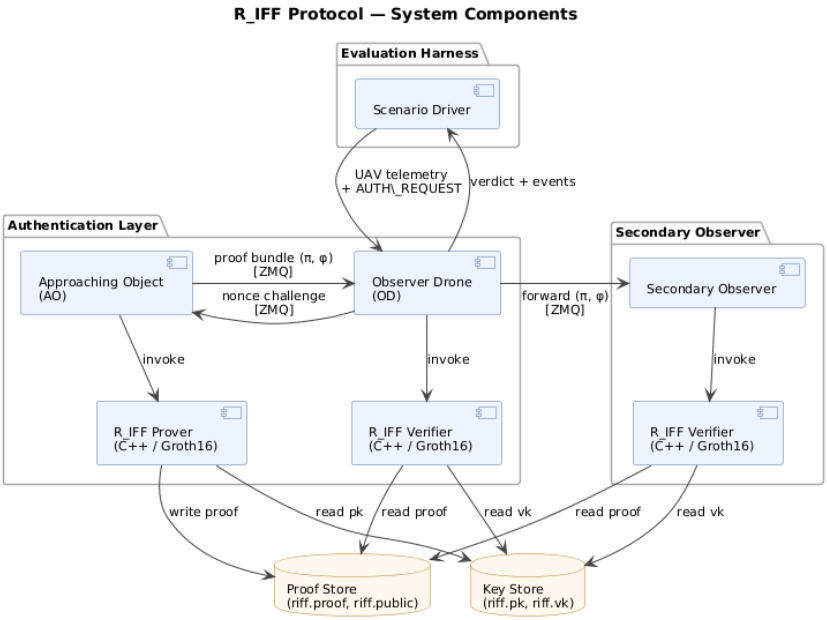

Figure 4.1: System component overview. The Scenario Driver publishes synthetic UAV telemetry over a ZMQ message bus. The prover and verifier are invoked as C++ subprocesses; keys and proof artefacts are exchanged via the shared file store.

- The *OD* role is played by the Authentication Bridge operating on behalf of the GCS. It generates nonces, assembles the public input challenge, performs anchor pre-checks, invokes the verifier binary, and publishes verdicts.
- The *AO* role is played by a second Authentication Bridge instance configured in prover mode. It receives the challenge, constructs the witness vector, invokes the prover binary, and returns the proof tuple.
- A *secondary observer* process independently re-verifies every forwarded proof tuple, demonstrating the multi-observer property of Section 3.4.

All three processes communicate exclusively via ZMQ through the shared telemetry bus, with no shared memory or direct function calls between them. The proving and verification keys (pk, vk) are generated once during pre-deployment setup via a dedicated `setup_main` binary and stored on disk. The Authentication Bridge pins the verification key fingerprint at startup and rejects any session where the presented vk does not match the pinned value, implementing the CRS anchoring described in Section 3.3.

Figure 4.2 shows the message sequence for a single authentication cycle, proceeding in four phases: challenge, proof generation ($\approx 963\mathrm{ms}$ ($\sigma \approx 14.2\mathrm{ms}$, $n = 80$)), verification (≈11 ms), and forwarding to the Secondary Observer.

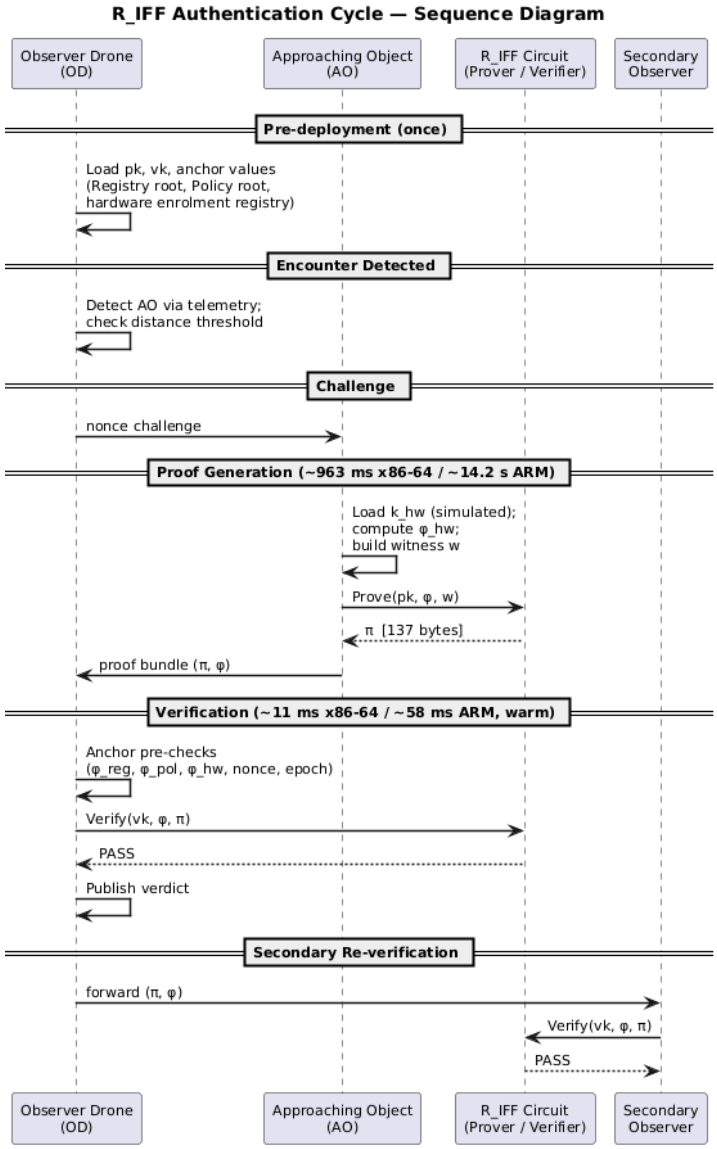

Figure 4.2: Authentication cycle sequence. The proof π is 137 bytes regardless of circuit complexity; verification does not require contact with the AO.

The proof tuple must be transmitted from the AO to the OD over MAVLink v2, which constrains the application payload to 255 bytes per frame [44]. The minimal bundle of `riff.proof` and `riff.public` is 410–413 bytes raw, requiring 2 MAVLink v2 packets at a frame overhead of 5.5%. A packed 32-byte binary encoding would reduce this to 297 bytes, still within 2 packets. The current PoC transmits a full 8-file bundle for auditability, which when encoded requires approximately 5 MAVLink v2 packets for internal fragmentation. However, this bundled data is transmitted as a single application-layer message over one logical send operation, so for the protocol loss model, it constitutes one message transmission attempt. By transmitting only the joint proof, fragmentation is reduced by a factor of approximately 2.4 without loss of verifiability. MAVLink v2 has no native fragmentation-reassembly mechanism, so application-layer segmentation logic is left for future work. Table 4.1 details the per-file breakdown.

File	Raw (bytes)	Base64 (bytes)	Required for verification
`identity.proof`	137	184	Joint circuit only
`identity.public`	155	208	Joint circuit only
`hardware.proof`	137	184	Joint circuit only
`hardware.public`	194-197	260-264	Joint circuit only
`policy.proof`	137	184	Joint circuit only
`policy.public`	81	108	Joint circuit only
`riff.proof`	137	184	**Required**
`riff.public`	273-276	364-368	**Required**
Minimal bundle	**410-413**†	**548-552**	2 MAVLink packets
Full bundle	**1,251-1,257**	**1,680-1,684**	5 MAVLink packets

Table 4.1: Proof bundle composition and MAVLink v2 fragmentation. Groth16 proof size is constant at 137 bytes regardless of circuit complexity. Public input file sizes vary with field element count (S1: 2, S2: 3, S3: 2, joint: 5) and value-dependent decimal text serialisation. †MAVLink packet counts are unaffected by the size range.

4.2. Circuit Implementation

The four statements of R_{IFF} are implemented in the libsnark gadgetlib1 framework [59] using the Groth16 proving system over the BN128 elliptic curve (ALT_BN128 in the libsnark build system). The circuit library is written in C++14 and compiled with `-O2 -march=native`. Each statement is compiled as a standalone binary for testing, while the joint R_{IFF} circuit is used during protocol execution. BN128 provides mature zk-SNARK tooling and efficient pairing performance, although its effective security level is approximately 100 bits following the Kim–Barbulescu attack [39], and it is not a NIST-approved curve. These limitations are discussed in Section 5.3.

All circuits use the Poseidon hash function [32] over the BN128 scalar field, chosen over SHA-256 due to substantially lower R1CS constraint cost and therefore reduced proving time. It is instantiated with $\alpha = 17$, $R_F = 8$, and $R_P = 31$. This choice is non-standard: mainstream Poseidon deployments typically use $\alpha = 5$ for 254-bit fields, and derive round counts accordingly.

The Poseidon parameterisation used in this PoC is not based on a published round-number derivation and deviates from standard MDS (maximum distance separable) constructions. It was chosen as a lightweight implementation to support functional evaluation for the protocol rather than to provide production-grade cryptographic assurances. Any security arguments in this thesis relying on collision-resistance or preimage-resistance should therefore be interpreted as applying to a correctly instantiated Poseidon, not this specific implementation. This parameterisation affects both security assumptions and performance results, since round count and field structure directly determine the number of constraints and thus proving time.

Using these building blocks, Statements 1, 3, and the joint circuit verify Merkle membership with a binary sparse Merkle tree (SMT) gadget of depth 4, supporting 16 leaves. At each level, the gadget enforces a boolean direction bit ($d(d-1) = 0$), conditional left and right child selection, and a Poseidon hash to compute the parent node. The resulting path is constrained to terminate at the registry root ϕ_{reg} or policy root ϕ_{pol}, respectively.

Statement 2 does not require a Merkle proof. Instead, it enforces two Poseidon hash equalities, as hardware binding only verifies that the measured hardware hash matches the enrolled reference contained in the registry leaf.

With the individual gadgets defined, the joint circuit combines the constraints of Statements 1–3 into a single R1CS. It verifies the identity leaf, hardware key, and policy credential against a shared public input vector, allocates separate SMT paths for the registry and policy trees, and enforces the epoch tolerance $\phi^{\mathsf{cred}}_{\mathsf{epoch}} \in \{\phi_{\mathsf{epoch}} - 1, \phi_{\mathsf{epoch}}\}$ using a product constraint. Table 4.2 summarises the resulting circuit metrics.

Statement	Constraints	Witness size	Proving key (B)	Verification key (B)
S1 Identity	3,462	4,956	914,428	1,179
S2 Hardware	1,152	1,650	299,631	1,220
S3 Policy	3,463	4,958	915,053	1,182
Joint R_{IFF}	8,077	11,563†	2,093,523	1,289

Table 4.2: Circuit complexity metrics. Witness size denotes the total number of quadratic arithmetic program (QAP) variables, including intermediate wire values. Each per-statement row includes the real Poseidon hash and 4-level Merkle-path gadgets, consistent with the joint circuit. The joint proving key scales linearly with the constraint count (≈260 B/constraint), giving approximately 2 MB. †The constraint count is the exact sum of the three sub-statements ($3{,}462 + 1{,}152 + 3{,}463 = 8{,}077$); folding grants no constraint-level saving, since $h_{\mathsf{hw}}^{\mathsf{enrol}}$ is consumed by a different constraint in each sub-statement (a hash pre-image in S1, an equality check in S2). The one-variable reduction in witness size relative to the naïve sum of 11,564 instead arises because $h_{\mathsf{hw}}^{\mathsf{enrol}}$, which a three-proof verifier would require as a public input in *both* S1 and S2 to check the two proofs agree, collapses into a single private witness wire shared by both sub-computations inside R_{IFF}, removing its public disclosure entirely.

Table 4.3 decomposes the 8,077 constraints by component. The two SMT path verifications contribute 57.2% of all constraints and thus dominate proving cost, motivating the depth-scaling analysis in Section 5.3. The per-statement figures in Table 4.2 report each statement instantiated with the same Poseidon and Merkle-path gadgets as the joint circuit; the standalone verification binaries used for the modular ablation (Section 4.3.8) employ simplified placeholder gadgets for isolation and are not performance-representative, serving only to demonstrate the structural cross-asset property of Theorem 3.1.

Constraint source	Constraints	Share
Poseidon: identity path ($H(ID_{priv})$, registry leaf)	1,150	14.2%
Poseidon: hardware binding ($H(k_{\mathsf{hw}})$, session commitment)	1,150	14.2%
Poseidon: policy path (attribute hash, policy leaf)	1,150	14.2%
Merkle registry path (4 levels)	2,312	28.6%
Merkle policy path (4 levels)	2,312	28.6%
Equality constraints (Statement 2)	3	$<0.1\%$
Total	**8,077**	**100%**

Table 4.3: Constraint breakdown of the joint R_{IFF} circuit.

Finally, Table 4.4 summarises the public inputs and proof sizes. The enrolled hardware hash $h_{\mathsf{hw}}^{\mathsf{enrol}}$ is an explicit public input in S1 and S2, where it serves as the verifier anchor. In the joint circuit, it is enforced internally through the registry SMT path and therefore does not appear as a primary input.

Statement	Public inputs	Count	Proof (B)	Public input file (B)
S1 Identity	$\phi_{\mathsf{reg}},\ h_{\mathsf{hw}}^{\mathsf{enrol}}$	2	137	155
S2 Hardware	$\phi_{\mathsf{nonce}},\ \phi_{\mathsf{hw}},\ h_{\mathsf{hw}}^{\mathsf{enrol}}$	3	137	194–197
S3 Policy	$\phi_{\mathsf{epoch}},\ \phi_{\mathsf{pol}}$	2	137	81†
Joint R_{IFF}	$\phi_{\mathsf{reg}},\ \phi_{\mathsf{nonce}},\ \phi_{\mathsf{epoch}},\ \phi_{\mathsf{hw}},\ \phi_{\mathsf{pol}}$	5	137	273–276

Table 4.4: Public inputs and proof sizes for each circuit. Groth16 proofs over BN128 are always 137 bytes. Public input files use libsnark's decimal text format, so file sizes vary with the encoded field values. †The S3 input file is small because $\phi_{\mathsf{epoch}} = 0$ during evaluation; larger epoch values produce files of approximately 160 bytes.

4.3. Experimental Methodology

This section describes the hardware platforms, network impairment model, flight scenarios, metrics, and test design used to evaluate R_{IFF}.

4.3.1. Experimental Setup

All experiments were conducted on a single x86-64 host running WSL2 (AMD Ryzen 7 4800H, 8 cores, Linux kernel 6.6.114). No hardware parallelism was used; within each authentication cycle, the prover,

verifier, and orchestration steps executed sequentially. UAV motion and telemetry are provided by FlyNetSim in direct mode, with inter-UAV distances computed from simulated GPS coordinates injected over ZMQ.

Network impairment is introduced at the Authentication Bridge layer using a configurable link model that applies per-message Gaussian-distributed delays and probabilistic drops. This operates at the protocol level rather than through kernel-level `tc netem`, isolating authentication performance from simulator or OS networking artefacts. The model assumes independent and identically distributed delay and loss, and does not capture temporal correlation or burst loss seen in real RF channels (e.g. the Gilbert–Elliott model). As a result, consecutive packet drops are less likely than in realistic wireless conditions, meaning the measured success rates are optimistic and true operational performance would likely be lower under bursty RF interference.

Each experiment is repeated over a limited number of cycles, with 30 runs per delay condition and 50 runs per loss condition. This choice is constrained by variance characteristics of the target platforms: on the ARM setup, prover latency exhibits a standard deviation of approximately 400 ms, so 30 cycles yields wide confidence intervals and limits interpretability of delay-sweep results (see Section 4.4.2). In contrast, on x86-64, where prover variance is below 25 ms, the same number of cycles is sufficient to estimate central tendency with reasonable stability.

4.3.2. Flight Scenarios

Four scenario profiles are defined to span encounter geometries relevant to federated UAV operations (Table 4.5): Proximate Hold isolates cryptographic cost under fixed close-range conditions; Transit Encounter models a high-speed pass that compresses the authentication window; Mission Separation stresses increasing link distance and delay under diverging trajectories; and Joint Operation combines multiple phases including approach, split, transit, loiter, and rerouting. Together, these scenarios cover mean inter-UAV distances from 4–160 m and baseline link delays from 0–79 ms.

Scenario	Description	Mean dist.	Base delay
Proximate Hold	Both UAVs stationary at 4 m; fixed link, no distance model	4 m	0 ms
Transit Encounter	UAV1 passes UAV0 at 13.7 m/s; separation 0–230 m	107 m	54 ms
Mission Separation	Both UAVs diverge to 300 m; increasing separation	160 m	79 ms
Joint Operation	Phased mission with approach, split, transit, loiter, reroute	117 m	64 ms

Table 4.5: Flight scenario profiles used in evaluation. Base delay is the mean per-cycle link delay from the distance model at sweep delay $d = 0$; it is not controlled by the delay sweep parameter.

4.3.3. Metrics

Metric	Definition and scope
Prover wall time (`r_iff_wall_ms`)	Elapsed time of the `prover_main` process from start to termination. Includes witness generation and Groth16 proof computation. Excludes network delay, ZMQ communication, and file I/O. Used for measuring cryptographic computation cost only.
Verifier wall time (`r_iff_verify_ms`)	Elapsed time of the `verifier_main` process. Includes only Groth16 proof verification via pairing checks.
End-to-end latency (`e2e_latency_ms`)	Time from nonce generation at the OD to receipt of confirmation by the secondary observer. Includes network delay, ZMQ messaging, process startup overhead, file I/O, and cryptographic computation. Used for measuring full system latency.
Authentication success rate	Fraction of cycles with `cycle_result = PASS` over all executed cycles, including those that timeout.
Timeout rate	Fraction of cycles with `cycle_result = TIMEOUT`. A timeout occurs if the protocol does not complete within 30 seconds.

Table 4.6: Metric definitions. Prover and verifier wall time isolate cryptographic cost, while end-to-end latency measures total system performance.

4.3.4. Test Design

Metrics are collected across three test categories (Table 4.7). The baseline configuration evaluates all four scenarios under zero impairment (no delay or loss), providing a reference point for cryptographic performance independent of network effects. Network impairment is then studied using two independent sweeps: delay and packet loss. Each sweep varies a single parameter while holding the other fixed at zero, allowing the effect of delay and loss to be characterised independently.

Adversarial tests differ in nature: they evaluate structural security properties of the protocol rather than performance. Each adversarial case targets a specific attack class and is executed as a single trial, so no cycle averaging is applied.

Delay sweeps use 30 cycles per condition. This is sufficient on x86-64, where prover variance is below 25 ms, but limits statistical precision on ARM, as discussed in Section 4.3.1.

Test	Variable	Range	Fixed	Cycles
Baseline	—	d=0, l=0	—	80/scenario
Delay sweep	One-way delay	0, 50, 100, 200 ms	l=0	30/condition
Loss sweep	Packet loss	0, 1, 2, 5%	d=0	50/condition
Adversarial T1	Re-verification check	—	—	single
Adversarial T2	Nonce replay detection	—	—	single
Adversarial T3	Proof tampering detection	—	—	single
Adversarial T4	VK impersonation attack	—	—	single
Adversarial T5	Epoch confusion attack	—	—	single
Adversarial T6	Cross-asset witness mix	—	—	single

Table 4.7: Test design. Baseline runs cover all scenarios under $(d = 0, l = 0)$. Total sweep evaluation consists of $4 \times 4 \times 30 = 480$ delay-sweep cycles and $4 \times 4 \times 50 = 800$ loss-sweep cycles, plus 80 baseline cycles per scenario. Adversarial tests T1–T6 are single-trial structural security checks and are excluded from cycle counts.

All latency results are computed over `PASS` cycles only. Success and timeout rates are computed over all cycles. Figures are generated from the sweep data described in Section 4.3.4.

4.3.5. Scenario Baseline Performance

We first evaluate system behaviour under zero impairment across all four scenarios. This isolates cryptographic and protocol execution costs from network effects.

Figure 4.3 shows the end-to-end authentication latency distributions. Proximate Hold exhibits the tightest distribution (median 1,829 ms, IQR $<$35 ms), since it has fixed 4 m separation and no distance-based link variation. The remaining scenarios show higher medians (1,988–2,121 ms) and wider spread, driven by per-cycle variation in inter-UAV distance and the resulting link delay. In Mission Separation, for example, the 160 m mean separation induces a baseline delay of 79 ms, contributing approximately $3 \times 79 \approx 237$ ms to total end-to-end latency relative to Proximate Hold.

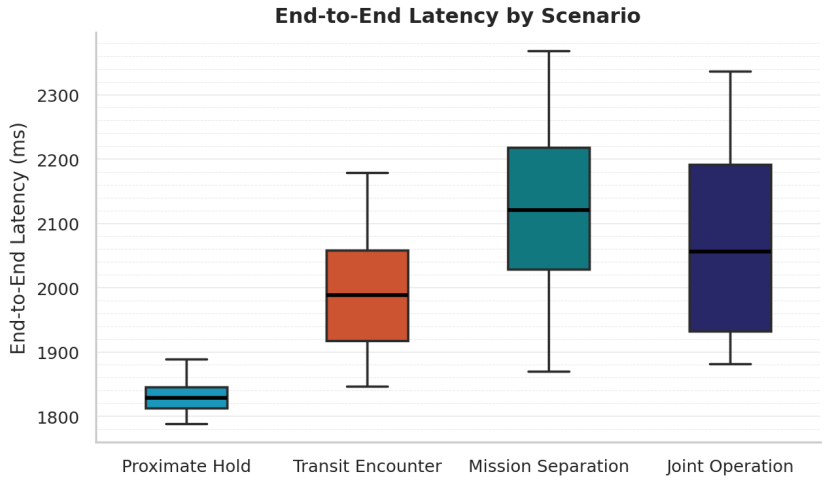

Figure 4.3: End-to-end authentication latency under baseline conditions. Boxes indicate IQR; whiskers extend to 1.5×IQR; outliers are shown individually. Variance in Proximate Hold reflects prover jitter only (59 ms peak-to-peak), while dynamic scenarios also include distance-induced variation.

To separate cryptographic cost from network effects, Figure 4.4 isolates prover and verifier execution time. Prover wall time[2] is stable at $\approx 963\text{ms}$ ($\sigma \approx 14.2\text{ms}$ ms; 95% CI for Proximate Hold [959.7, 965.9] ms), confirming that proving cost depends only on circuit complexity and is independent of flight geometry. The verifier completes in approximately 10.8 ms, yielding a roughly 89:1 asymmetry between prover and verifier cost. This is important as the observer node can verify multiple concurrent sessions with negligible computational load.

The gap between prover wall time and end-to-end latency is not dominated by network or messaging cost. Instrumenting the prover subprocess directly shows that approximately 93% of this gap on x86-64 lies inside the prover invocation, before the proof is transmitted. The reported prover wall time measures only the core `r1cs_gg_ppzksnark_prover` call; it excludes deserialisation of the ∼2 MB proving key, elliptic-curve parameter initialisation, and R1CS witness generation, all of which execute per cycle. On x86-64 this hidden cryptographic-path cost adds approximately 0.81 s, with the remaining ∼60 ms attributable to transmission, verification, and forwarding.

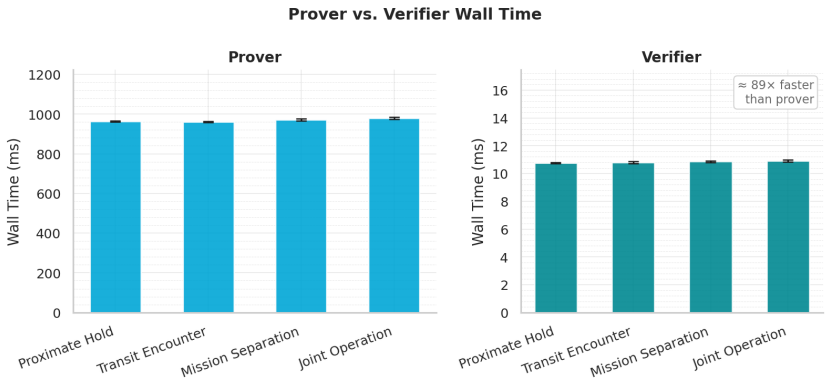

Figure 4.4: Prover and verifier wall time under baseline conditions. Error bars show 95% confidence intervals. Prover cost is scenario-independent; verifier cost is approximately 89× lower.

4.3.6. Delay Sensitivity

We next evaluate sensitivity to injected one-way network delay while holding packet loss at zero.

[2]Prover wall time measures the core proving call and excludes proving-key deserialisation, curve initialisation, and witness generation. It therefore understates the full per-cycle cryptographic-path cost by roughly 45% on both platforms. End-to-end latency and D_{crit}, which are computed from the full cycle time, are unaffected by this definition.

Figure 4.5 shows that end-to-end latency increases linearly with injected delay across all scenarios. The slope is approximately 3 ms per 1 ms of injected delay, consistent with the three-message protocol structure. Differences between scenarios appear only in the intercepts, which reflect baseline distance-dependent delay. The identical slopes indicate that delay sensitivity is determined by protocol structure and not mobility or geometry. At 200 ms injected delay, end-to-end latency ranges from approximately 2,432 ms (Proximate Hold) to 2,699 ms (Mission Separation).

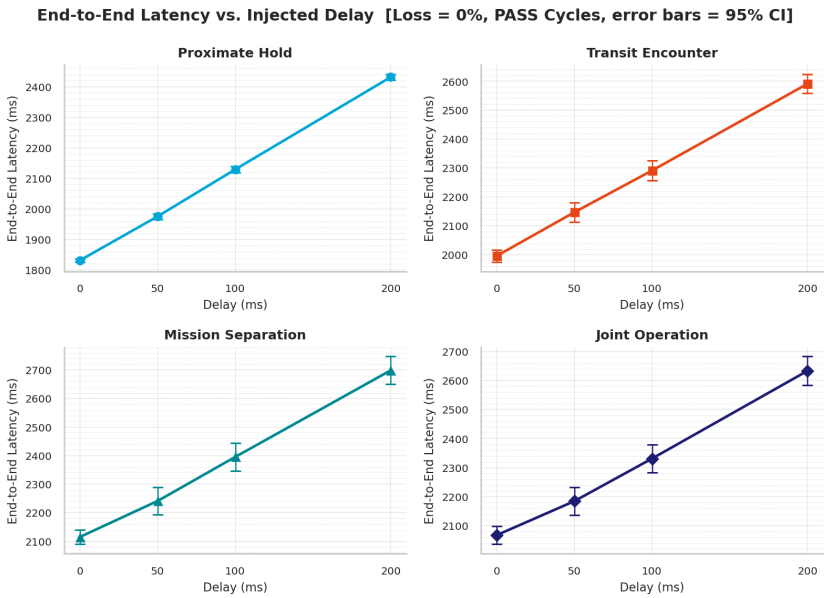

Figure 4.5: End-to-end latency versus injected one-way delay (zero packet loss). Error bars show 95% confidence intervals. The linear slope is consistent across all scenarios.

4.3.7. Packet Loss Sensitivity

We now evaluate robustness under increasing packet loss with zero injected delay.

Across all scenarios, success rates remain 100% at $l = 0\%$, above 96% at 1% loss, and between 86–94% at 5% loss (Figures 4.6–4.8).

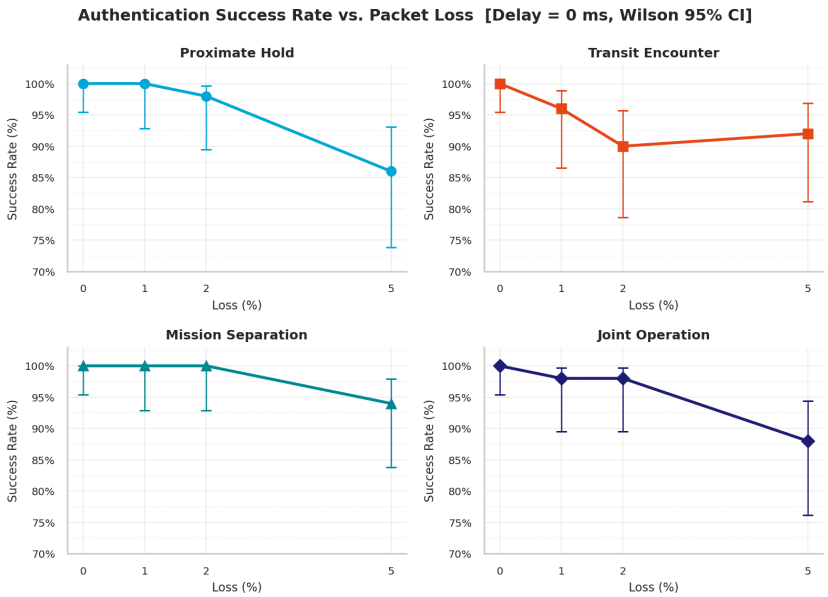

Figure 4.6: Authentication success rate versus packet loss (all cycles). Error bars show Wilson 95% confidence intervals.

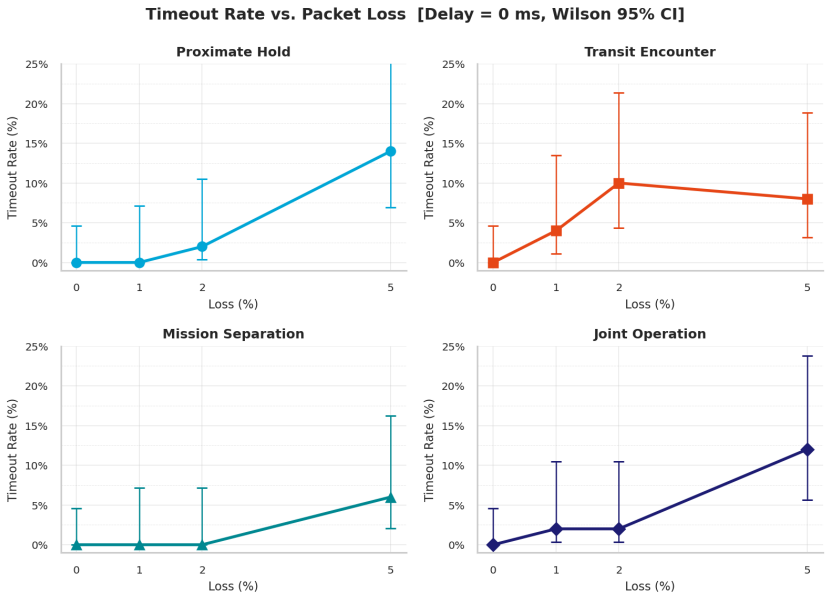

Figure 4.7: Timeout rate versus packet loss (all cycles). Error bars show Wilson 95% confidence intervals.

At 5% loss ($l = 0.05$), the three message exchanges (challenge, proof, verdict) succeed independently with probability $(1 - l)^3$. This gives $(0.95)^3 \approx 85.7\%$ success, i.e., a 14.3% failure rate, consistent with observed 6–14% timeouts due to early termination. Since the protocol does not implement retransmission, loss primarily reduces the number of completed cycles rather than increasing latency. This is confirmed in Figure 4.8, where end-to-end latency remains stable across loss conditions for successful cycles (mean variation $<$50 ms). In other words, packet loss removes executions but does not slow successful ones.

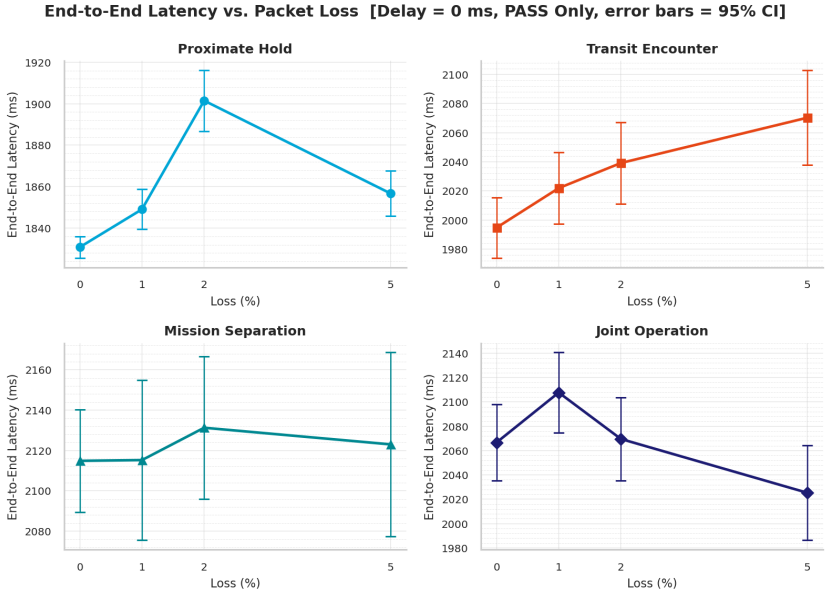

Figure 4.8: End-to-end latency under packet loss (PASS cycles only, zero delay). Latency remains stable across loss conditions; error bars show 95% confidence intervals.

4.3.8. Adversarial Security Tests

We now evaluate correctness under adversarial conditions. These tests target protocol-level security properties under the assumed cryptographic model (physical-layer attacks and side-channel analysis

are outside scope). Each test corresponds to a specific attack class from Section 2.4. T1–T5 validate protocol mechanisms, while T6 provides a constructive test of Theorem 3.1.

ID	Attack	Check	Result
T1	Secondary Observer Re-verification	Proof verified by OD; same bundle verified by secondary observer independently	**PASS**
T2	Nonce Replay Detection	Proof for nonce_A accepted with nonce_A	**PASS**
		Proof for nonce_A rejected with nonce_B	**PASS**
T3	Proof Tampering	Valid proof accepted	**PASS**
		Proof with corrupted bytes rejected	**PASS**
T4	Cross-VK Impersonation	Attacker proof accepted under own VK	**PASS**
		Attacker proof rejected under legitimate VK	**PASS**
T5	Epoch Confusion	Proof at epoch=3 accepted with epoch=3	**PASS**
		Proof at epoch=3 rejected with epoch=5	**PASS**
		Proof restored to epoch=3 accepted again	**PASS**
T6	Cross-Asset Witness Mixing	See Table 4.9	**FAIL**

Table 4.8: Adversarial test outcomes. FAIL in T6 is the expected correct outcome.

T1 (Secondary Observer Re-verification). A valid R_{IFF} proof was generated and verified by the OD. The proof tuple was forwarded to a secondary observer holding only the public verification key. The secondary observer independently executed `verifier_main` and obtained the same PASS result. This confirms that the forwarded proof bundle is self-contained and does not require interaction with the AO or OD.

T2 (Nonce Replay Detection). A valid proof generated under `nonce_A` was submitted to a verifier expecting `nonce_B`. The verifier compared the nonce embedded in the proof's public input against the expected nonce and detected the mismatch before running the Groth16 verification, rejecting the proof and returning exit code 1. This confirms that the nonce is checked as plaintext equality and that cross-session replay is prevented by the nonce field in the public input. It does not evaluate whether a captured, unaltered proof would be accepted by a secondary observer that never issued the nonce, which is the deferred-replay case discussed in Section 5.3.

T3 (Proof Tampering Detection). A valid serialised proof was modified by corrupting selected bytes and submitted for verification. The verifier returned exit code 1. This is expected, as any modification invalidates the Groth16 pairing equation.

T4 (Cross-VK Impersonation). An attacker generated a proof using an independent CRS and their own proving key. Verification under the legitimate verification key returned exit code 1. The rejection follows directly from Groth16 structure, which binds all proof elements to the verification key's embedded trapdoor.

T5 (Epoch Confusion). A valid proof generated with $\phi_{\mathsf{epoch}} = 3$ was submitted to a verifier configured for epoch 5. The anchor pre-check detected the mismatch and returned exit code 1. Restoring epoch 3 restores acceptance, consistent with Claim 3.5.2 (Context Binding) in Section 3.5.2.

T6 (Cross-Asset Witness Mixing). To evaluate Theorem 3.1, a joint-circuit witness was constructed by combining identity material from one enrolled asset ($\mathsf{id\_seed} = 0$) with hardware material from another ($\mathsf{hw\_seed} = 1$), both registered under the same authority as a controlled cross-asset case. To demonstrate the vulnerability addressed by this attack, we constructed a modular baseline with three independent proofs (`all` mode) and verified that T6 succeeds against it. The attack recombines an identity statement from one asset with a hardware statement from another. Because the modular baseline proves these statements independently, it contains no shared witness wire for $h_{\mathsf{hw}}^{\mathsf{enrol}}$ and therefore accepts the recombined pair. In contrast, R_{IFF} compiles identity and hardware into a single R1CS in which $h_{\mathsf{hw}}^{\mathsf{enrol}}$ is shared across the relevant constraints, so any cross-asset recombination violates the

constraint system and is rejected at proving time. This property derives from the shared-wire construction rather than the standalone membership logic and would hold even if each independent proof implemented full membership verification. Table 4.9 summarises the evaluation.

Table 4.9: T6: cross-asset witness-mixing test. Steps 1–2 validate individual credentials. Step 3 shows that modular `all` mode cannot detect cross-asset inconsistency. Step 4 confirms that the joint `riff` circuit rejects mismatched witnesses.

Step	Mode	What it shows	Result
1	`identity` (seed 0)	Identity credential for Asset A is valid	PASS
2	`hardware` (seed 1)	Hardware credential for Asset B is valid	PASS
3	`all` (seeds 0/1)	The two statements are proven independently with no shared wire binding identity to hardware; modular verification imposes no cross-statement constraint and accepts the recombined pair	PASS
4	`riff_mixed` (id 0, hw 1)	Joint circuit rejects mismatched witness at verification; witness construction succeeds, but `verifier_main` returns exit code 1 due to constraint violation	FAIL

Steps 1–2 confirm that both credentials are individually valid under their respective checks. Step 3 highlights a limitation of modular verification since independent evaluation does not enforce cross-asset consistency. Step 4 provides the main result where the joint circuit enforces global consistency at the R1CS level and rejects mismatched witnesses. This contrast isolates the benefit of the joint construction in preventing cross-asset composition errors that are not visible under separate circuit execution.

Across T1–T6, the evaluation is limited to structured protocol-level adversarial cases. More advanced threat models including malicious prover behaviour beyond witness mixing, CRS compromise, and Byzantine observer behaviour are left as future work in Section 5.3.

4.4. Embedded Hardware Evaluation

To evaluate deployability, the complete protocol stack is executed on embedded hardware representative of a small UAV-class platform. The system is deployed on a Raspberry Pi 3B+ running Ubuntu 24.04 (ARM Cortex-A53, 1 GB RAM), and directly compared to the x86-64 baseline from Section 4.3.5. The build is identical except for two platform-specific changes: `-DCURVE=ALT_BN128` selects the pure C++/GNU multiple precision arithmetic library (GMP) backend in place of x86-64 assembly optimisations, and `-DWITH_PROCPS=OFF` disables a monitoring dependency not available on ARM. All field arithmetic thus executes via GMP. All scenarios and sweep configurations from Section 4.3.4 were reproduced on the ARM platform using the same timeout parameters. To account for the greater variability in ARM proving times, the delay-sweep sample size was increased to 140 authentication cycles per condition (141 for the Proximate Hold scenario at $d = 50$). A 2 GB swap file is used only for compilation; runtime remains within physical RAM. As in Section 4.3.1, measurements reflect co-located execution. All results are reported using ALT_BN128 over GMP and are directly comparable to the x86-64 baseline.

4.4.1. Scenario Baseline Performance

Figure 4.9 shows end-to-end latency under zero impairment. Compared to x86-64, all scenarios exhibit a uniform upward shift in latency and substantially increased variance. Proximate Hold has a median of 25,321 ms with an IQR of 1,148 ms, while dynamic scenarios range from 25,625 ms (Transit Encounter) to 26,980 ms (Joint Operation). The increase relative to x86-64 (IQR 33–259 ms) is primarily driven by ARM CPU frequency scaling and higher prover variance. Early cycles show a visible dynamic voltage and frequency scaling (DVFS) ramp-up effect; these cold-start cycles are included in the statistics, making the reported baseline slightly conservative.

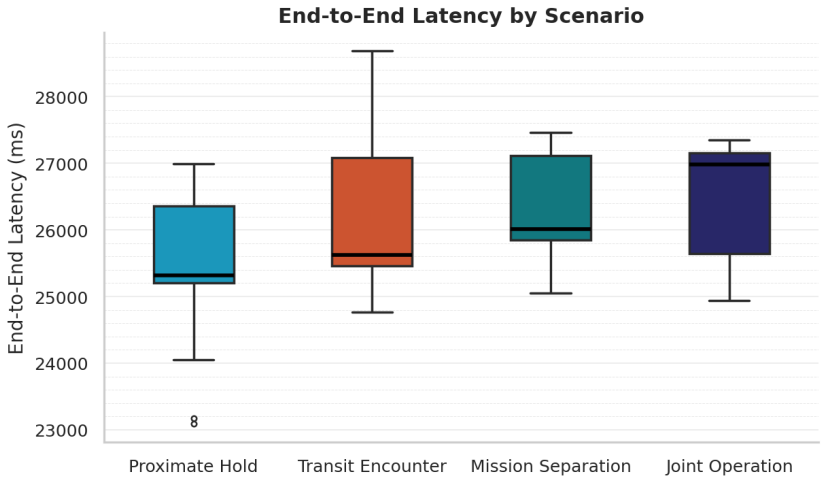

Figure 4.9: End-to-end latency on RPi ARM under baseline conditions. Variance is dominated by ARM CPU frequency scaling and prover jitter, resulting in wider distributions than x86-64.

To isolate cryptographic computation from system-level effects, Figure 4.10 reports prover and verifier wall times. Prover latency ranges from 13,970 ms to 14,341 ms, corresponding to a 14.5 to 14.8$\times$ slowdown relative to x86-64 ($\approx 963\text{ms}$). Verifier latency averages 58.4 ms, a 5.4$\times$ slowdown relative to x86-64 (10.8 ms). Despite this slowdown, the prover-to-verifier asymmetry remains strong at approximately 239:1, compared to 89:1 on x86-64. This preserves the key property that the observer node incurs negligible per-encounter verification cost.

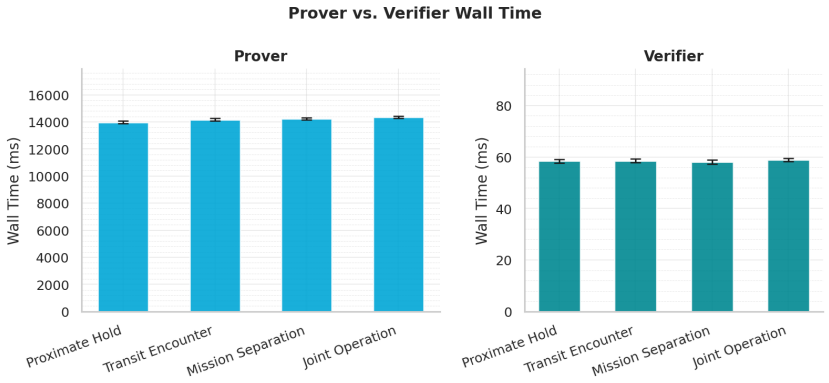

Figure 4.10: Prover and verifier wall time on ARM under baseline conditions. The prover remains dominant in computation cost, while verifier overhead remains negligible.

4.4.2. Delay Sensitivity

We next evaluate sensitivity to injected network delay. In principle, the same three-message structure implies a linear relationship with slope approximately 3 ms per 1 ms of injected delay, as observed on x86-64.

Figure 4.11 shows that this trend is not reliably observable on ARM. The prover standard deviation of 360 to 453 ms is comparable to the maximum delay contribution of 600 ms at 200 ms injection, meaning measurement noise dominates signal strength. As a result, different scenarios exhibit inconsistent or flattened slopes, even though all follow the same underlying protocol. Across all injected-delay conditions, all 1,681 delay-sweep cycles at $d > 0$ completed successfully, yielding a 100% PASS rate.

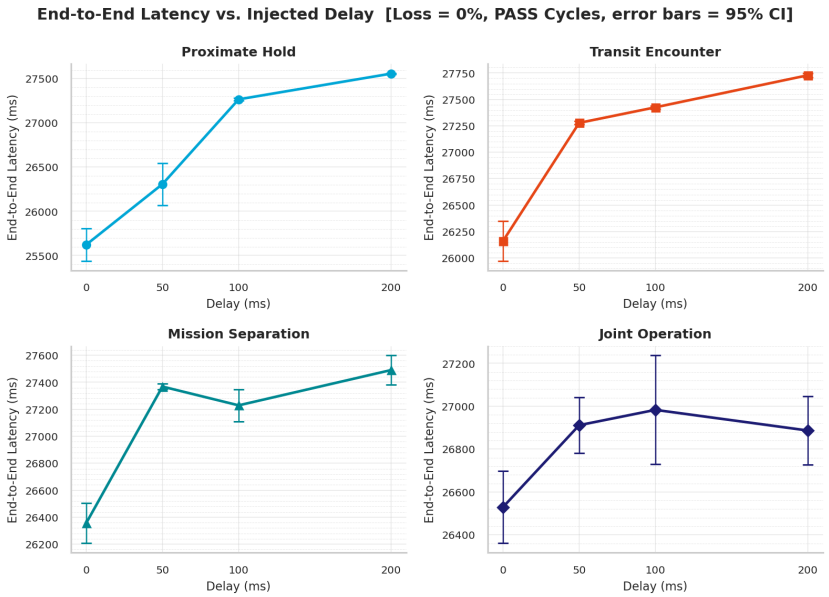

Figure 4.11: End-to-end latency versus injected delay on ARM. High prover variance limits observability of the theoretical 3 ms/ms slope seen on x86-64.

4.4.3. Packet Loss Sensitivity

Packet loss behaviour is consistent with the x86-64 results. All scenarios achieve 100% success at 0% loss. At 1%, success remains at or above 98%, and at 5% loss it ranges from 88% to 96%. Timeout rates remain below the theoretical three-message independent loss bound of 14.3%, confirming early termination effects. As on x86-64, loss reduces the number of completed executions but does not materially affect latency for successful runs.

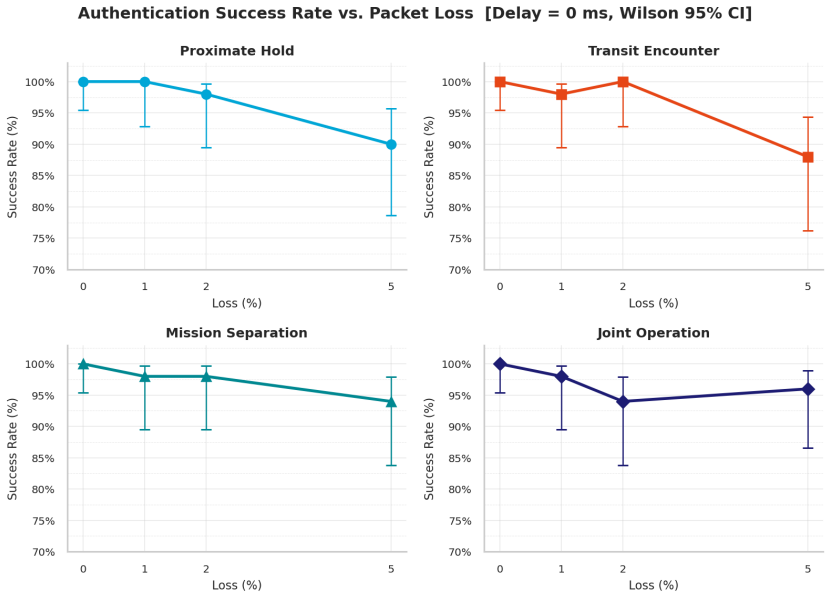

Figure 4.12: Authentication success rate on ARM under packet loss. Behaviour matches x86-64 trends, indicating architecture-independent protocol loss sensitivity.

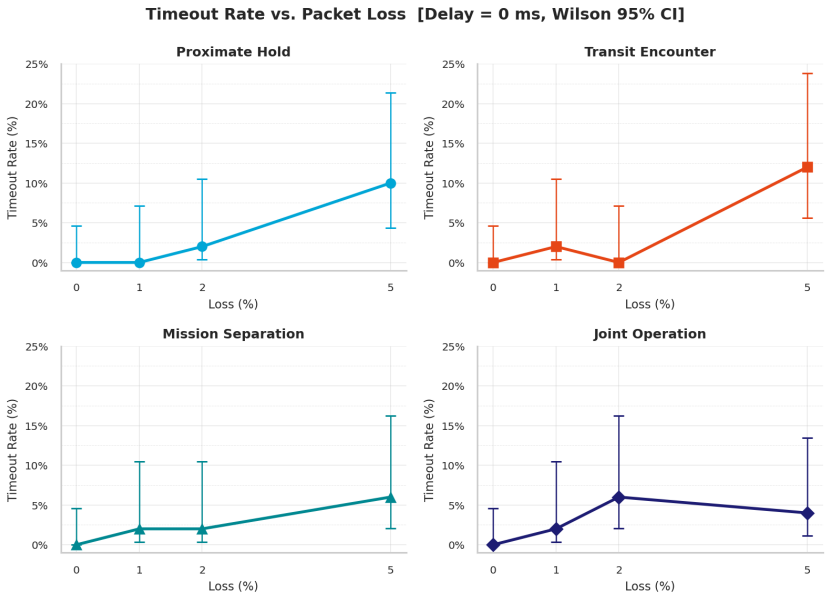

Figure 4.13: Timeout rate versus packet loss on ARM. Loss primarily affects completion probability rather than the latency of successful runs.

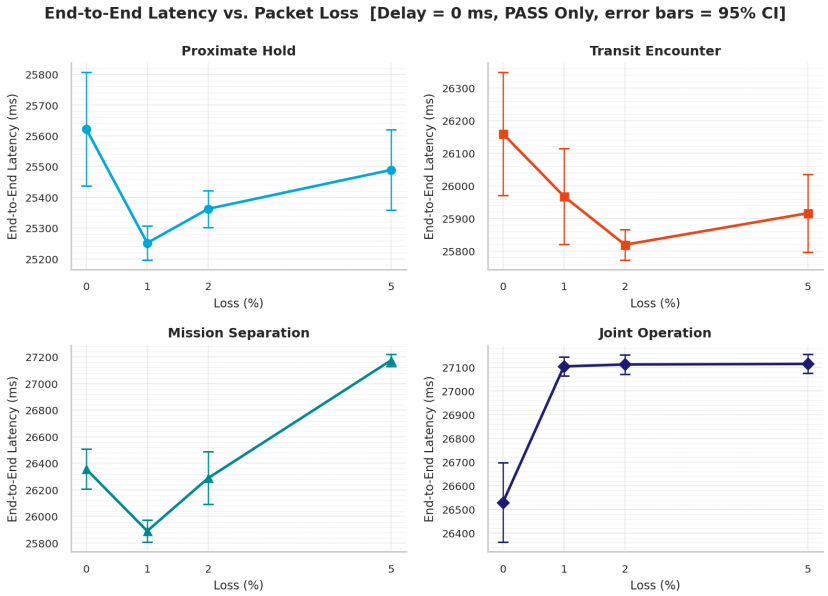

Figure 4.14: End-to-end latency versus packet loss on ARM. Latency is stable across loss conditions for successful cycles.

4.4.4. Adversarial Security Tests

Adversarial tests T1–T5 from Section 4.3.8 are repeated on the RPi under identical conditions and all return PASS. T6 is platform-independent because it evaluates algebraic satisfiability of the joint circuit rather than runtime behaviour; it is thus executed only once on x86-64. Across all cases, outcomes are determined by the Groth16 constraint system and verifier pre-check logic, both of which are independent of hardware architecture.

4.5. Cross-Platform Comparison

Table 4.10 summarises cryptographic performance across both evaluation platforms, while Figure 4.15 provides a visual comparison of baseline timing behaviour. Success and timeout rates are determined

by protocol message structure and are therefore included for completeness only.

Metric	x86-64	RPi ARM	Slowdown
Prover wall time	$\approx 963\text{ms}$	13,970 to 14,340 ms ($\sigma \approx 400$ ms)	14.5 to 14.8×
Verifier wall time	10.8 ms	58.4 ms	5.4×
Prover/verifier ratio	89:1	239:1	—
E2E latency (median)	1,829 to 2,121 ms	25,321 to 26,980 ms	12.3 to 13.8×
Success at l=1%	≥96%	≥98%	—
Success at l=5%	86 to 94%	88 to 96%	—
Adversarial tests	T1–T6 PASS	T1–T5 PASS	—

Table 4.10: Cross-platform performance comparison at zero impairment baseline. Both platforms use ALT_BN128 over GMP arithmetic. Slowdown is relative to the x86-64 host. T6 was executed on x86-64 only. 95% confidence intervals for Proximate Hold (d=0, $n = 80$): prover x86-64 [959.7, 965.9] ms, ARM [13871, 14070] ms; verifier x86-64 [10.7, 10.8] ms, ARM [57.7, 59.1] ms; E2E x86-64 [1825.6, 1836.0] ms, ARM [25438, 25807] ms.

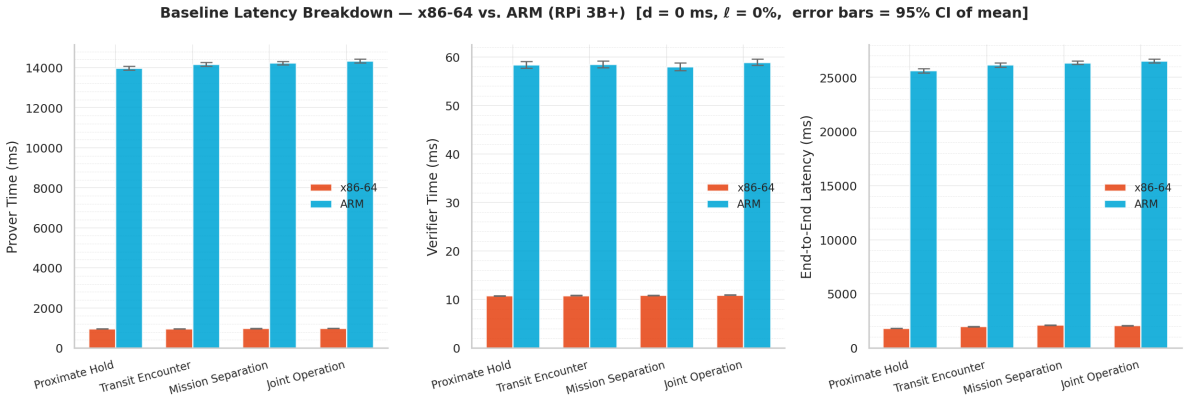

Figure 4.15: Side-by-side comparison of prover wall time, verifier wall time, and end-to-end latency for x86-64 and RPi ARM at baseline. Error bars show 95% confidence intervals of the mean. The prover slowdown of approximately 15× dominates system performance on the RPi, while the verifier slowdown of approximately 5× has a smaller absolute impact due to its already low baseline cost.

Across all metrics, the dominant difference between platforms is the cost of prover computation. On x86-64, prover wall time is $\approx 963\text{ms}$ ($\sigma \approx 14.2\text{ms}$, $n = 80$), increasing to 13,970–14,340 ms ($\sigma \approx 400$ ms) on RPi ARM, corresponding to a 14.5–14.8× slowdown. Verifier wall time increases from 10.8 ms to 58.4 ms (5.4× slowdown), while end-to-end latency rises from 1,829–2,121 ms to 25,321–26,980 ms (12.3–13.8× slowdown). The prover-to-verifier ratio increases from 89:1 to 239:1, reflecting the comparatively smaller scaling penalty on verification.

Despite these performance differences, protocol-level behaviour remains consistent across platforms. Success at $l = 1\%$ remains high in both cases (≥96% on x86-64 and ≥98% on ARM), and at $l = 5\%$ remains comparable (86–94% versus 88–96%). Adversarial robustness is also preserved, with T1–T6 passing on x86-64 and T1–T5 on ARM; T6 was not executed on ARM.

Differences in low-level arithmetic efficiency primarily explain the observed performance gap. The 14.5–14.8× prover slowdown arises from the absence of assembly-optimised big-integer arithmetic on ARM. In this configuration, ALT_BN128 relies on GMP for field operations, which performs substantially less efficiently on ARM than on x86-64 systems with SSE/AVX2 vectorisation support. The verifier exhibits a smaller 5.4× slowdown because pairing verification involves fewer field multiplications than proof generation, reducing sensitivity to arithmetic throughput.

These results have direct implications for real-time deployment. The measured 25–28 s end-to-end latency on RPi ARM exceeds the approach-phase window of time-critical scenarios such as Transit Encounter (approximately 16.8 s for $v = 13.7$ m/s at $D = 230$ m), resulting in zero completed authentication cycles. Real-time deployment on this platform would therefore require performance optimisations such

as assembly-level acceleration of BN128 pairing arithmetic on Cortex-A, a persistent prover daemon to remove per-cycle startup overhead, or offloading proof generation to a companion processor. These changes are performance-only and do not affect protocol correctness or security guarantees.

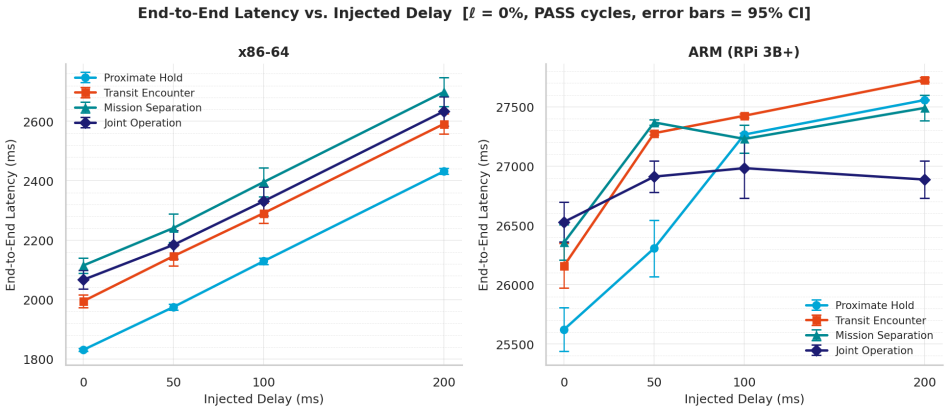

Figure 4.16: Delay sweep comparison across platforms. The x86-64 panel shows a stable linear gradient of approximately 3 ms/ms across all scenarios (Figure 4.5). On RPi, the same theoretical gradient holds by protocol structure but becomes less visually distinct due to higher prover variance, even at increased sample sizes ($n = 80$ at $d = 0$; $n = 140$ at $d = 50, 100, 200$ ms).

Figure 4.16 shows the delay sweep behaviour across impairment levels. On x86-64, latency increases linearly with a stable gradient of approximately 3 ms/ms across all scenarios, consistent with the baseline trend in Figure 4.5. On RPi ARM, the same underlying linear relationship is preserved by protocol design, but the trend is partially obscured by higher variance in prover execution times. This effect persists even with larger sample sizes ($n = 80$ at $d = 0$; $n = 140$ at $d = 50, 100, 200$ ms), indicating that variability drives the visual difference between platforms.

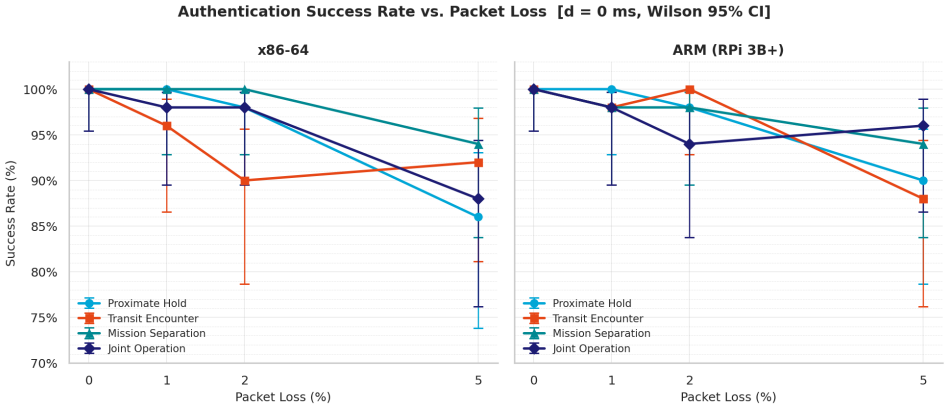

Figure 4.17: Loss sweep comparison across platforms. Success rate profiles are closely matched, confirming that packet-loss sensitivity is a property of the protocol's message structure rather than of cryptographic performance. Error bars show Wilson 95% confidence intervals.

Figure 4.17 presents the loss sweep comparison. Both platforms exhibit closely aligned success-rate curves across increasing packet loss, with overlapping confidence intervals throughout. This consistency confirms that loss sensitivity is governed by protocol message structure rather than platform-specific cryptographic execution, reinforcing that observed differences in latency do not translate into differences in reliability behaviour.

Further deployment implications are discussed in Chapter 5.

5

Discussion

This chapter interprets the experimental results from Chapter 4 with respect to the three sub-research questions, evaluates the architectural and implementation decisions that shaped these outcomes, and identifies the key limitations and open problems that constrain the scope of the contribution. The focus is on the implications of the measured behaviour for deploying zero-knowledge attestation in contested UAV swarm environments.

5.1. Interpretation of Results

Before interpreting individual findings, the role of R_{IFF} is clarified to ensure correct framing. It is not introduced as a new NIZK primitive, but as a deployment methodology for privacy-preserving coalition IFF. The cryptographic building blocks remain standard (Groth16, Poseidon, sparse Merkle trees); the contribution lies in their composition into a unified circuit, the evaluation under simulated network conditions across platforms, and the resulting deployability framework that links measured latency to encounter-level constraints. This distinction is important for correctly interpreting the claims supported by the results.

With this framing in place, the results are evaluated against three primary objectives: *(i)* maintaining low verifier-side computational overhead during encounter-based authentication, *(ii)* ensuring robustness under degraded and variable network conditions, and *(iii)* validating that the constructed relation resists malformed, replayed, and context-inconsistent proof tuples.

The main research question concerns whether zero-knowledge attestation can support IFF decisions without relying on identity-based authentication. The evidence indicates that this is feasible under the conditions evaluated in this work. In particular, verifier cost remains negligible across platforms, the relation demonstrates soundness against the evaluated attack classes, and authentication completes reliably within operationally relevant time windows on x86-64 hardware.

Open issues include prover feasibility on autopilot-class hardware beyond the Raspberry Pi 3B+ used in this evaluation, the relay attack described in Section 5.3, and revocation propagation latency in denied or disconnected environments. The following subsections hence interpret the results strictly within these constraints.

The satisfaction of each functional, security, and operational requirement is in Appendix A.

5.1.1. SRQ1: Zero-Knowledge Attestation for Control State Verification

The central question is whether zero-knowledge attestation can support IFF decisions without relying on identity-based authentication. The results indicate that this is achievable within the scope and assumptions of this work, and that the joint circuit design is the key architectural element enabling it.

R_{IFF} encodes three trust conditions, namely federation membership, possession of an enrolment-bound hardware secret, and policy authorisation, into a single Groth16 proof of 137 bytes. The verifier accepts

this proof as evidence that all three conditions hold simultaneously without revealing any of the underlying attributes. No persistent identifier, hardware secret, sector assignment, or role is disclosed to the OD. Instead, session-specific values ϕ_{nonce} and ϕ_{hw} change with each encounter. As established in Section 3.5.1, this provides within-federation unlinkability: sessions cannot be linked through the proof by any observer that does not hold the enrolment registry. At the same time, the shared roots ϕ_{reg}, ϕ_{pol}, and ϕ_{epoch} remain fixed within a mission epoch across all federation members. This allows a passive observer to detect that an authentication event occurred within a known coalition while preventing identification of the specific asset. Under the threat model used in this work, where federation membership is assumed to be observable by an adversary, this separation is acceptable because asset-level identity, hardware state, and authorisation attributes remain hidden.

While zero knowledge hides witness values, it does not hide the existence of communication. An adversary observing the RF channel can still detect that an authentication exchange took place, measure its duration, and infer approximate properties of the proof system from bundle size patterns. Repeated observations can leak higher-level information, including mission tempo, and cross-session timing correlation may allow reconstruction of swarm density without revealing individual identities. These effects are forms of traffic analysis that lie outside the zero-knowledge guarantee and thus require operational mitigations such as randomised timing and cover traffic. They are inherent limits of observable wireless communication rather than limitations of the protocol itself.

The choice of a joint circuit over three independent proofs matters most under realistic compromise scenarios. In coalition UAV operations, enrolled assets cannot be assumed to remain fully secure. A UAV may be physically captured, its hardware-bound key extracted through firmware compromise, or its identity credentials exfiltrated via a compromised provisioning channel, all without affecting other assets in the system. This risk is amplified under the threat model adopted in this work, in which credential material for a single asset is assumed to be provisioned through coalition channels instead of a single trusted operator. Where different custodians hold enrolment secrets, identity credentials, and policy material with independent security postures, a partial compromise of one asset cannot be excluded even when the rest of the federation remains secure. This work does not attempt to characterise the provisioning practices of any specific federation, and treats distributed custody as an assumption that motivates the design. The consequence for a modular design is direct. An adversary who obtains the identity credential of asset A and the hardware key of asset B through separate compromises could combine a valid S1 proof from A with a valid S2 proof from B, and because each proof verifies independently, the mismatch would go undetected. The joint construction of R_{IFF}, defined in Chapter 3, closes this gap by forcing both conditions onto the same witness through the shared $h_{\mathsf{hw}}^{\mathsf{enrol}}$ wire, so a cross-asset pairing has no valid registry leaf and the constraint system is unsatisfiable. This is a stronger guarantee than modular approaches, which can only compare independently verified outputs after proof generation.

Result T4, which concerns cross-VK impersonation, should be interpreted separately from the shared wire argument. T4 shows that a proof generated under an attacker-controlled CRS trapdoor is rejected when verified against the legitimate verification key, which is a standard Groth16 soundness property and does not depend on circuit structure. By contrast, the shared wire constraint is a property of the R_{IFF} circuit itself and remains valid even when the attacker has access to the correct proving key. Together, these results cover complementary aspects of security. T4 establishes that proofs cannot be forged under an incorrect CRS, while the joint circuit structure ensures that proofs cannot be constructed by combining credentials from different assets under a valid CRS. Both properties are required for the overall security argument and address distinct threat vectors.

Existing UAV-oriented ZK authentication schemes reviewed in this work do not provide the cross-asset binding established above. ZAPS [49] and Koulianos et al. [40] apply zk-SNARKs to location and authorisation proofs but do not include hardware binding or a forwarding model for secondary observers. Walshe et al. [77] demonstrate NIZK authentication under reduced-connectivity IoT conditions but likewise omit hardware attestation. As a result, none of these schemes would detect the partial-compromise scenario described above, since each evaluates its statements independently. In contrast, this work enforces hardware binding at the circuit level, but evaluates it only as an algebraic property. Test T6 demonstrates that the circuit rejects a mismatched hardware witness, confirming the binding logic but not a physical anchor. Although the $\mathsf{PUF}(d)$ call in Algorithms 1 and 2 denotes the intended hardware

source, k_{hw} is simulated rather than derived from a physical PUF. Consequently, any security property that depends on the unclonability of the hardware anchor is assumed by the design rather than empirically validated in this work.

A direct comparison across schemes is inherently limited because they differ in hardware assumptions, threat models, and proof system design. Nevertheless, Table 5.1 summarises the closest operationally comparable systems with respect to the properties most relevant for UAV deployment.

Scheme	Proof	Verifier (x86-64)	Verifier (ARM)	HW Binding	Fwd.
ZAPS [49]	128 B	∼3 to 5 ms[†]	—	No	No
Petzi et al. [53]	—[‡]	—	390 ms[§]	Yes (PUF)	No
This work (R_{IFF})	**137 B**	**10.8 ms**	**58 ms**	**Simulated**	**Yes**

Table 5.1: Comparison against prior UAV-oriented ZK authentication schemes. Proof sizes exclude public inputs. [†]ZAPS does not report standalone verification latency; values shown are standard Groth16 estimates over BN128 [33]. [‡]Interactive ZKP, so no succinct proof size applies. [§]Measured on LPCXpresso55S69 (150 MHz Cortex-M33 with ECC acceleration); without acceleration 3.24 s. ARM column for this work corresponds to ALT_BN128 over GMP on Raspberry Pi 3B+; ARM data for ZAPS is not reported.

On x86-64, verifier latency of 10.8 ms is comparable to ZAPS and lower than Petzi et al., while additionally supporting hardware binding and forwarding, which are not provided in prior schemes. On ARM, the 58 ms verification time remains within the same order of magnitude as Petzi et al.'s accelerated 390 ms result, despite the latter running on a significantly more constrained Cortex-M33 platform with hardware ECC support. The proof size of 137 B is consistent with ZAPS at 128 B and remains constant as a structural property of Groth16, independent of circuit complexity.

The prover-to-verifier ratios of 89:1 (x86-64) and 239:1 (ARM) reflect the asymmetry between constant-cost verification and circuit-dependent proving. This asymmetry aligns with the UAV authentication setting, where the OD must handle multiple incoming authentication events under tight encounter constraints, while the AO bears the computational burden of proof generation. Even on ARM, verifier throughput remains sufficient for realistic encounter rates given the low absolute verification cost.

R_{IFF} is defined using collision-resistant hash functions and Merkle path verification only, without embedding elliptic curve or pairing assumptions into the relation itself. This makes the protocol logic and deployment structure backend-agnostic in principle, allowing alternative SNARK instantiations without modifying the relation beyond the underlying hash gadget. However, backend substitution is not operationally neutral. As discussed later in Section 5.1.2, increased proving cost directly affects deployability through encounter-time constraints, meaning that any alternative backend must be evaluated not only for compatibility but also for its impact on real-time feasibility, particularly on ARM-class hardware where timing margins are already limited.

5.1.2. SRQ2: Computational, Latency, and Communication Constraints

The second question concerns the practical constraints of deploying zero-knowledge attestation on resource-constrained UAV platforms. The results are structured into three constraint categories across two hardware platforms, which together provide a more complete view of system behaviour than either platform alone.

On x86-64, prover wall time is stable at approximately $963\mathrm{ms}$ with a coefficient of variation below 3%. This stability simplifies timeout configuration and scheduling on the AO. The cost is driven by the 8,077 R1CS constraints in the joint circuit. Poseidon was selected because its field-native structure reduces the number of constraints per hash operation compared to SHA-256, which directly reduces proving cost.

On ARM, the 14.5 to 14.8$\times$ slowdown relative to x86-64 appears primarily attributable to the GMP fallback path for big-integer arithmetic. This suggests that the dominant limitation lies in the arithmetic backend and not in the ARM architecture itself, although this work does not isolate these effects experimentally. Existing implementations of assembly-optimised BN128 pairing for Cortex-A indicate that

such optimisations would substantially reduce proving time, although they were not evaluated in this study.

End-to-end latency scales linearly with RF link delay at approximately 3 ms per 1 ms of injected one-way delay d, where d is the per-crossing delay applied to the link and L_{base} is the latency measured at $d = 0$. The slope of three arises because three messages in each cycle, namely the challenge, the proof, and the verdict, each traverse the delayed link once and so each contributes one further d to the critical path. A fourth message, the forwarding of the proof tuple to the secondary observer, uses a separate fixed link delay and therefore adds a constant term. This gives the relationship $L_{e2e} \approx L_{\text{base}} + 3d$, which holds across all measured scenarios on x86-64 at delay conditions of 0, 50, 100, and 200 ms. A linear fit to the measured cycle times yields a slope between 2.87 and 3.01 across the four scenarios with $R^2 \geq 0.997$, confirming that exactly three delay-bearing crossings sit on the critical path. The relationship allows direct estimation of latency under varying network conditions.

On ARM, the same protocol structure implies the same 3 ms per ms gradient. However, this effect is not reliably observable against prover variance. The prover exhibits a standard deviation between 360 and 453 ms, which is large relative to the maximum delay contribution of 600 ms at $d = 200$ ms. As a result, delay-induced variation is initially masked at small sample sizes.

An initial measurement set of 30 cycles per condition produced apparent differences of up to 6% between delay scenarios. Increasing the sample size to 140 cycles per condition reduces this variation, with all scenarios converging to a cycle time of approximately 26 s at $d = 0$. Across both sample sizes, a consistent proportional effect remains. The Proximate Hold ratio of cycle times, $T_{\text{cycle}}(d{=}200)/T_{\text{cycle}}(d{=}0)$, equals 1.076 and is identical at both $n = 30$ and $n = 140$. A bootstrap analysis of the mean difference $T_{\text{cycle}}(d{=}200) - T_{\text{cycle}}(d{=}0)$ yields a 95% confidence interval of [1752, 2114] ms, confirming that the effect is statistically non-zero. This corresponds to a 7.6% increase in cycle time from 200 ms of injected delay, which is small in absolute terms. On ARM-class hardware, reducing network delay yields almost no operational benefit, whereas reducing prover cost reduces the minimum separation distance needed for a successful encounter at every closure speed.

The Transit Encounter scenario provides an operational interpretation of these latency results. At a closure speed of 13.7 m/s, two UAVs remain within 230 m of each other for approximately 33 s. However, authentication must be completed during the approach phase (before closest approach), which lasts approximately 16.8 s. On x86-64, a single authentication cycle completes in under 2.2 s, allowing multiple attempts within this window. On ARM, a single cycle requires approximately 25–28 s, exceeding the 16.8 s approach phase and resulting in zero completed authentication cycles before closest approach. To generalise encounter feasibility beyond this scenario, the *minimum separation distance* required to complete at least one authentication cycle is defined as

$$D_{\text{crit}} = v \cdot T_{\text{cycle}}$$

where v is the closure speed and T_{cycle} is the measured cycle time. This formulation assumes constant closure speed, no manoeuvring, immediate back-to-back attempts without backoff, and link conditions independent of distance. The formulation assumes that the authentication cycle begins at the moment the two platforms are separated by distance D. In a real deployment, detection, non-cryptographic pre-classification, and link establishment all precede the first proof message, so the effective stand-off required is larger than D_{crit} predicts. The values reported here should thus be read as a lower bound on the separation needed to complete one cycle, and a deployed system would require additional margin to absorb these pre-cycle costs. The full breakdown across scenarios and delay conditions is reported in Appendix B.

Across all scenarios, D_{crit} differs significantly between platforms due to prover cost. On x86-64, values remain below 110 m even at 40 m/s closure speed. On ARM, AO has to be detected 350 metres away and start the authentication cycle immediately, while on x86-64, the AO can be only 25 metres.

Figures 5.1 and 5.2 translate the D_{crit} constraint into a more intuitive quantity: the number of complete authentication cycles available before two platforms meet, defined as $N_{\text{attempts}} = \lfloor (D/v)/T_{\text{cycle}} \rfloor$. Because N_{attempts} increments by one each time the available time window crosses another multiple of T_{cycle}, the result is a step function. On x86-64, where $T_{\text{cycle}} \approx 1.83$ s, the steps are fine-grained, a few

metres or a few m/s wide. On ARM, where $T_{\text{cycle}} \approx 26$ s, the steps are hundreds of metres or tens of m/s wide.

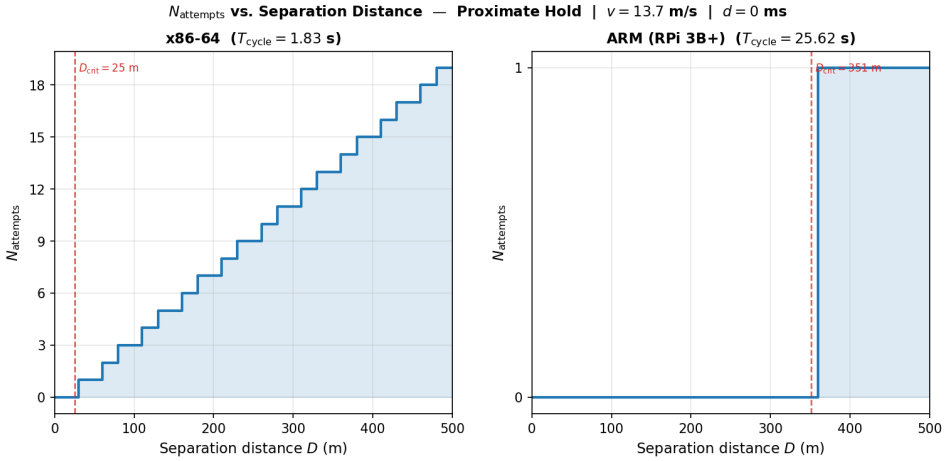

Figure 5.1: Number of complete authentication cycles $N_{\text{attempts}} = \lfloor (D/v)/T_{\text{cycle}} \rfloor$ as a function of initial standoff distance D at fixed closure speed 13.7 m/s, Proximate Hold scenario, $d = 0$. On x86-64, at least one cycle is feasible from approximately 25 m, and multiple cycles are feasible beyond 125 m. On ARM, a single cycle becomes feasible only beyond approximately 350 m.

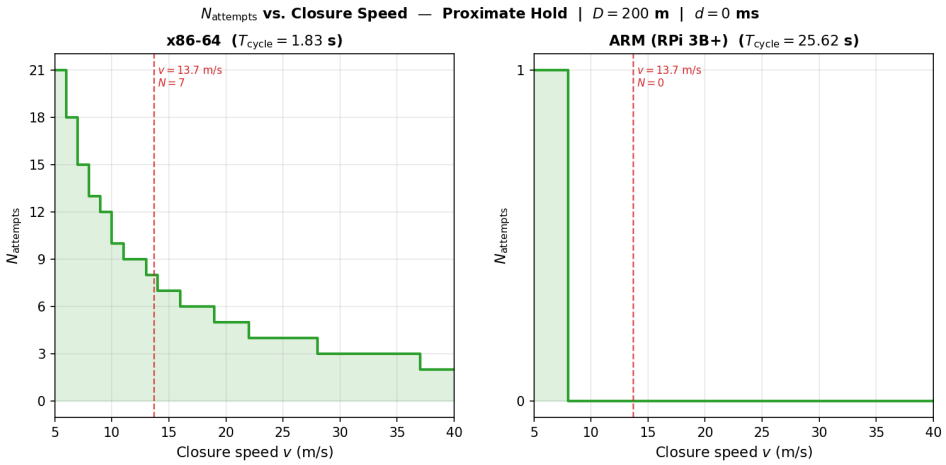

Figure 5.2: Number of complete authentication cycles $N_{\text{attempts}} = \lfloor (D/v)/T_{\text{cycle}} \rfloor$ as a function of closure speed v at fixed separation distance 200 m, Proximate Hold scenario, $d = 0$. On x86-64, at least one cycle remains feasible across the evaluated speed range. On ARM, feasibility is lost above approximately 7.8 m/s.

At the representative Transit Encounter geometry ($D = 230$ m, $v = 13.7$ m/s, encounter window 16.8 s), the x86-64 implementation permits eight complete authentication opportunities while the ARM implementation permits none, since a single cycle takes 25 to 28 s. This follows directly from $N_{\text{attempts}} = \lfloor (D/v)/T_{\text{cycle}} \rfloor$ and requires no assumptions beyond measured cycle time.

This finding reveals two qualitatively different operational regimes. When $N_{\text{attempts}} \geq 1$, as on x86-64, failed cycles can be retried, providing robustness against packet loss. When $N_{\text{attempts}} = 0$, as on ARM at this geometry, reliability analysis is irrelevant since no attempt completes within the encounter window regardless of link conditions. The key question is therefore not how reliable the protocol is but whether it can complete at all, and that question is answered by latency.

The approximately 14$\times$ slowdown of the ARM prover, rather than any single fixed overhead, is what places the protocol in the $N_{\text{attempts}} = 0$ regime for this geometry. Decomposing the measured Proximate Hold cycle at $d = 0$ shows that, of the total 25.6 s, approximately 14.0 s is spent generating the joint proof, while the remaining 11.65 s is attributable to process invocation, messaging, verification, and

other auxiliary computation. This decomposition indicates the limit of prover-only optimisation. Even if proof generation were instantaneous, the authentication cycle would still require approximately 11.65 s, giving $D_{\text{crit}} \approx 160$ m at a closure speed of 13.7 m/s, substantially higher than the 25 m achieved on x86-64. Reaching comparable performance on ARM therefore requires reducing both proving time and the remaining protocol overhead. Published assembly-optimised BN128 arithmetic for Cortex-A processors typically achieves a 1.3–4$\times$ speedup over the GMP fallback used in this prototype, while a persistent prover daemon can eliminate per-cycle startup overhead. Neither optimisation was implemented in this work, and their combined impact remains future work. Since part of the 11.65 s residual is also computational, the 160 m estimate should be regarded as a conservative upper bound rather than a hard lower limit on achievable performance.

Communication overhead is bounded and predictable. The 137-byte proof size is a fixed property of Groth16 over BN128, independent of circuit complexity. The minimal proof bundle is 410–413 bytes raw, fitting within 2 MAVLink v2 packets. A packed binary encoding reduces this to 297 bytes, still within 2 packets. Unlike latency, communication cost does not vary with platform or scenario and does not affect N_{attempts}.

Packet loss sensitivity is consistent across both platforms and is governed by message structure rather than prover performance. Success rates at 1% loss remain at or above 96% on x86-64 and 98% on ARM. For system design, timeout configuration should be driven by worst-case prover latency, and application-layer retransmission is needed to recover dropped cycles under degraded link conditions.

5.1.3. SRQ3: Multi-Observer Verification Properties

This section evaluates whether the proposed design satisfies the four properties defined in Section 3.4.1. Each property is assessed through an analytical argument that is supported by empirical results from both the x86-64 and ARM platforms.

Consistency follows directly from the determinism of the verification function $\mathrm{Verify}(\mathsf{vk}, \phi, \pi)$. Any two observers that hold identical anchor values and the same verification key reach the same outcome for a given proof tuple, and they do so without coordinating with one another. The experimental results support this property. Across all 3,881 sweep cycles, comprising 1,280 on x86-64 and 2,601 on ARM, no completed cycle produced a disagreement. A disagreement would therefore signal stale or corrupted anchor values rather than a cryptographic failure. That is an architectural property that holds as long as anchor provisioning is synchronised at deployment, an assumption that matches the pre-mission distribution model used by FMN.

Locality holds because verification runs entirely on state that is already present on the verifying node, so it needs no network access, no external lookup, and no interaction with the prover. This was shown by running `verifier_main` as a standalone process, with no connection to either the prover or the primary observer, under all test conditions. The measured verifier times of 10.8 ms on x86-64 and 58 ms on ARM therefore reflect pure local computation. This property is important because any reliance on external connectivity would block authentication in exactly the environments where infrastructure is absent. Local execution instead allows any node holding valid anchor values to evaluate trust independently of ground control or neighbouring assets.

Correctness of forwarding means that passing the pair (π, ϕ) from one observer to another preserves its validity and adds no new trust assumptions, because any change to either component makes the Groth16 pairing equation fail. Two adversarial tests confirm this. Test T3 shows deterministic rejection of corrupted proofs, and test T4 shows that a proof made under a different common reference string is rejected regardless of whether its witness was valid. The forwarding experiments produced the same outcome. Across every completed cycle, on both platforms and under all sweep conditions, the secondary observer re-verified each forwarded bundle as a PASS. Correctness is therefore enforced by the algebra of the proof system rather than by application logic. The node that forwards a proof does not need to be trusted, since any tampering is rejected by construction. This sets the design apart from conventional authentication schemes, where re-verification depends on reaching the original prover or a trusted intermediary.

The final property concerns availability under degraded network conditions. On x86-64, every completed cycle produced a PASS at delays of up to 200 ms. The loss sweep gives success rates of 96 %

or higher at 1 % loss, and ARM behaves in much the same way. For the cycles that succeed, end-to-end latency stays roughly constant as the loss level rises on both platforms, which indicates that packet loss removes whole cycles rather than slowing the ones that finish. A further advantage is that each completed cycle delivers a trust decision to all secondary observers at once, and that decision stays valid until the nonce expires or the registry revokes it. Once a single cycle succeeds, trust is preserved across observers, so the need to re-authenticate during intermittent connectivity falls.

These availability figures carry an important qualification. They describe only whether a cycle that has started will finish, not whether a cycle can be both started and finished inside an operational time window. Section 5.1.2 shows that this precondition fails on ARM at realistic encounter speeds, because no complete cycle fits inside the Transit Encounter window. In that regime, the availability results carry no weight, whatever the delay or loss.

Taken together, the four properties show that the protocol behaves as a trust artefact that can be distributed. A traditional interactive protocol needs the prover to stay available throughout verification, whereas a protocol built on non-interactive zero-knowledge proofs separates the availability of the prover from the operation of the verifier. In contested wireless UAV settings with intermittent connectivity, this separation adds operational robustness without any change to the underlying cryptographic construction.

Three limitations bound these conclusions. The secondary observer runs on the same physical machine as the primary observer, so forwarding never crosses a real network link, which means that correctness of forwarding and locality hold at the cryptographic level whatever the topology, but the measured availability reflects a controlled setup. Consistency also assumes synchronised anchor values, and the effect of stale or divergent anchors across a distributed swarm was not evaluated. Finally, each adversarial test was run once. The outcomes are deterministic given the structure of the Groth16 pairing equation, so repeating them would not change the results, yet only a single empirical run is reported.

5.2. Implications for Practice

The results have several practical implications for deploying zero-knowledge authentication in federated UAV operations, including operational role assignment, infrastructure requirements, policy maintenance, privacy, and communication constraints.

The verification-to-prover cost imbalance observed in SRQ2 suggests a natural separation of responsibilities within a heterogeneous swarm. A single OD can verify proofs from many assets with negligible overhead while assets perform the more expensive proving operation, allowing verification capacity to scale with little additional computational cost. Computation alone does not determine role assignment, however, because the relationship $D_{\text{crit}} = v \cdot T_{\text{cycle}}$ shows that proving latency constrains encounter feasibility. Assigning the prover role to an ARM-class platform increases the minimum separation distance required for successful authentication by an order of magnitude compared to x86-64, independent of which device verifies. Platform choice therefore determines operational range as much as computational throughput.

Prover cost also has an energy component that is relevant for battery-powered platforms. The energy required to generate a proof can be estimated as $E = P \cdot t$, where P is the platform's active power draw and t is the measured proving time. On the x86-64 host, a Ryzen 7 4800H completes a proof in approximately 0.97 s; using a full 45 W TDP as a conservative upper bound yields about 44 J, likely overestimating single-core energy consumption. On the Raspberry Pi 3B+, which draws approximately 4.5 W under load, proof generation takes 13–14 s, corresponding to roughly 60 J. Although the embedded platform consumes considerably less power, its longer execution time results in a higher energy cost per proof under this model. For comparison, a commercial platform such as the DJI Matrice 300 RTK [20], with a dual-battery capacity of approximately 274 Wh, would expend only around 0.01% of a full charge on proof generation. In practice, computation is not the dominant energy cost. Each authentication cycle requires the aircraft to maintain its position while the proof is generated, so the propulsion energy consumed during the 25–28 s proving interval on the ARM platform is expected to exceed the computational energy by a substantial margin. Consequently, reducing proving latency lowers both the energy required for proof generation and the additional flight energy associated with loitering.

This computational constraint is counterbalanced by an infrastructure advantage. The locality property established in SRQ3 removes the need for runtime connectivity. Trust decisions rely entirely on anchor values provisioned before deployment, allowing operation in contested or remote environments. Revocation is handled through updates to the Merkle root, where removal of a compromised leaf invalidates its associated proofs once the updated root is distributed. This differs from certificate-based systems that depend on online validation or continuously accessible revocation lists. Revocation latency is thus bounded by infrastructure availability rather than cryptographic design, and the epoch mechanism reduces risk for policy freshness without removing this dependency for revocation events.

Epoch length becomes a deployment parameter. The circuit enforces a bounded epoch tolerance as confirmed in T5, preventing reuse of credentials across outdated epochs. Short epochs improve security by limiting the window in which a credential remains valid, but they raise the risk that legitimate assets fail verification if policy updates cannot propagate in time. Longer epochs reduce this operational risk but weaken freshness guarantees. Because revocation depends on the same out-of-band channel that may be unavailable mid-mission, the epoch window is in practice the only bound on how long a compromised but unrevoked credential remains usable in a disconnected environment, which raises the weight of the security side of this trade-off. The epoch tolerance is fixed in the circuit at proving time rather than adjustable at runtime, so the appropriate value depends on expected policy propagation delay and must be set as a mission-planning decision before deployment.

Zero-knowledge proofs prevent disclosure of asset-level attributes, but they do not conceal the existence or timing of an authentication exchange. An observer can still infer when authentication occurs, measure its latency, and extract structural information from public inputs such as federation membership and epoch state. This residual side channel is inherent to observable wireless communication, so mitigating it requires complementary communication-layer protections. Zero-knowledge conceals the authentication data, not the existence of the authentication exchange.

Integration into federated systems requires agreement on shared setup parametres before deployment. The relation is hardware-agnostic, allowing instantiation on different secure hardware substrates such as PUFs, TPMs, or secure enclaves, provided a stable hardware-bound secret can be derived. Interoperability under FMN requires agreement on the registry root, policy root, CRS, and epoch schedule, all established before deployment. During operation, the protocol exchanges bounded messages of approximately 500 bytes within the encounter window. The protocol assumes bidirectional communication, with one participant issuing a nonce challenge and the other returning a proof. Where both participants must authenticate each other, this bidirectionality is intrinsic. Where one participant is receive-only, the protocol in its current form does not apply, and an alternative architecture would be needed.

5.3. Limitations of This Work

Several limitations bound the scope of the results and claims in this thesis. They fall into four categories: the evaluation environment, the performance and scalability of the current implementation, remaining security questions, and the design trade-offs accepted by R_{IFF}'s architecture.

The most basic limitation is the evaluation environment. All experiments were run on a single x86-64 host with simulated telemetry and software-injected network conditions, without a real RF channel, physical UAV motion, or a hardware PUF. The results therefore characterise protocol logic and cryptographic cost and not RF behaviour, embedded scheduling effects, interrupt latency, or PUF error-correction overhead. The network impairments were also modelled in a simplified way. Delay and loss were evaluated independently, so their combined effects remain unknown. Loss was modelled as independent per-message drops, whereas real wireless losses tend to occur in correlated bursts. This assumption is optimistic and would likely underestimate the number of protocol cycle timeouts in a real deployment, making the reported loss results a best-case estimate.

On embedded hardware, the ARM prover ran 14.5–14.8$\times$ slower than x86-64. This slowdown is likely dominated by the arithmetic backend rather than the ARM architecture itself, since the prototype uses `libsnark`, whose big-integer arithmetic on ARM falls back to GMP without assembly-optimised BN128 support. Although the backend and hardware contributions are not isolated, assembly-optimised BN128 pairings for Cortex-A exist and would likely reduce proving time, as would a persistent prover

daemon that avoids per-request key loading. As discussed in Section 5.1.2, prover optimisation reduces only the proving component of the cycle: the measured 25.6 s ARM cycle decomposes into roughly 14.0 s of proving and an 11.65 s non-proving residual, so even eliminating proving entirely leaves $D_{\text{crit}} \approx 160$ m at the Transit Encounter baseline (13.7 m/s), still well above the 25 m achieved on x86-64. The measured slowdown is therefore better explained by the current software stack than an inherent limitation of the Cortex-A53. More modern platforms (e.g., Raspberry Pi 4/5 or Jetson boards) combined with optimised BN128 arithmetic would likely further improve feasibility at Transit Encounter speeds, but evaluating this is outside the scope of this work. Similarly, toolchains such as `arkworks`, `gnark`, and `halo2` provide more efficient field arithmetic and could materially change absolute timings, especially on ARM. Benchmarking R_{IFF} in these frameworks remains a prerequisite for operational assessment.

The registry also uses a fixed, small sparse Merkle tree. The evaluation considers a depth-4 tree supporting 16 leaves with seed-derived synthetic witnesses. To assess scalability, Table 5.2 reports measured constraint counts, QAP sizes, and x86-64 prover wall times for depths 4, 8, 10, and 12, corresponding to federations ranging from 16 to 4,096 enrolled assets.

Table 5.2: Measured scaling with registry depth on x86-64. Constraints increase by approximately 578 per tree, while prover time grows more slowly because the QAP size increases only at power-of-two boundaries. Most additional levels therefore add Poseidon evaluations without increasing QAP cost. Even at depth 12 (4,096 enrolled assets), proving remains below 2 s on x86-64.

Depth	Leaves	Constraints	QAP size	Prove (ms)
4	16	8,077	8,192	963
8	256	12,701	16,384	1,522
10	1,024	15,013	16,384	1,660
12	4,096	17,325	32,768	1,927

Two observations are noteworthy. First, proving at depth 12 stays under 2 s on x86-64, so the depth-4 prototype overstates x86-64 deployability by at most a factor of about 1.9 even for 4,096 assets. Depths 8 and 10 share the same QAP size of 16,384 because QAP size changes only at power-of-two boundaries, so prover time grows more slowly than constraint count. On ARM the picture is harder, because any depth increase adds to an already high prover cost.A depth-8 circuit (12,701 constraints) would increase the ARM proving component by roughly 57%, further raising D_{crit}; the exact figure depends on how much of the non-proving residual is fixed, and measuring it on ARM is left for future work. Measuring these depth-scaling figures on ARM, and under combined delay and loss, is left for future work.

The most serious open security question is the relay, or wormhole, attack. Because the proof tuple (π, ϕ) is non-interactive and anonymous, it acts as a bearer token that any verifier will accept within the current nonce and epoch window. An adversary positioned between the AO and OD, or able to relay RF traffic in real time, can forward a legitimate AO's fresh proof to the OD and receive a friend classification without holding any enrolled credential. This is not a replay attack, because the nonce still prevents reuse across sessions. The problem is that the nonce does not bind the proof to a specific physical sender or channel. Hardware binding does not prevent this, because k_{hw} remains inside the witness, so the verifier cannot tell whether the transmitting airframe is the one that generated the proof. Closing this gap needs mechanisms outside the current design, such as distance bounding [17], channel binding, or proof-of-proximity, each of which adds latency and protocol complexity. This remains the primary direction for future protocol work.

A related and equally unmitigated weakness follows directly from the verification procedure. The nonce cache $\mathcal{N}$ in Algorithm 3 is maintained per verifier, and the freshness test rejects only a nonce the verifier has already seen. It does not test whether the verifier issued that nonce. A secondary observer therefore has an empty cache for any nonce it did not challenge, so a captured valid tuple (ϕ, π) presented to a fresh observer within the same epoch passes the freshness test and verifies. This is a deferred replay to a secondary observer, distinct from the real-time relay attack above. The nonce prevents reuse only at the issuing observer, and the forwarding model gives a secondary observer no means of

distinguishing a legitimately forwarded proof from an adversary-replayed one. Closing this gap requires either shared nonce state across observers or the same class of proximity or channel binding needed to prevent relay, none of which is evaluated here.

A related weakness appears under degraded connectivity. The design treats the absence of a verifiable proof as grounds for an unverified classification, and under the evaluated loss conditions, 6 to 14% of cycles time out at 5% per-message loss. In an IFF setting, those assets would be classified unverified. Misclassifying a friendly asset as hostile risks friendly fire or unnecessary exclusion from coalition coordination. The protocol does not prescribe how unverified assets should be handled, which is a matter of operational doctrine, but the measured rate should inform retry logic and human-in-the-loop review thresholds so that friendly fire risk stays within acceptable bounds for the target environment. Because the loss model is optimistic, the real rate is likely higher, which strengthens the case for conservative thresholds.

The security claims rely on analytical arguments and adversarial testing. No model was built in tools such as ProVerif, Tamarin, or Scyther. The related work surveyed in Chapter 2 repeatedly treats formal verification as a strength, so its absence here is a relative weakness. A model of the nonce-binding and epoch-binding properties is feasible given the compact message structure of the protocol, and is left for future work. The properties evaluated in T2–T5 are structural and deterministic, so repeated trials add nothing, but a formal model would give machine-checked guarantees for the replay-resistance and context-binding properties under a Dolev-Yao or symbolic adversary. Minimal ProVerif or Tamarin model covering nonce and epoch binding is feasible given the compact message structure and is identified as near-term future work.

Groth16 also brings a trusted-setup dependency. The CRS is generated from a trapdoor, and any party that learns the trapdoor can forge proofs for the relation indefinitely. In practice, this is handled by an MPC ceremony in which the trapdoor is destroyed as long as at least one participant is honest. For a multi-national coalition, deciding who organises and audits the ceremony, and under which legal framework, is both a security and political question. This work assumes an honestly run ceremony and treats it as a first-class trust dependency. Where that assumption is unacceptable, transparent systems such as STARKs [9] avoid the trusted setup, but their proofs run to tens or hundreds of kilobytes. This is two to three orders of magnitude larger than the 137-byte Groth16 proof and would require hundreds of MAVLink packets instead of two, so for the constrained links assumed here the change may be prohibitive, and proving times are longer as well. This alternative was not evaluated.

The curve choice carries a similar caveat. The prototype uses Groth16 [33] over BN128 (ALT_BN128), which is not a NIST-approved curve and offers roughly 100 bits of security, lower than the approximately 128 bits of BLS12-381. Neither curve is NIST-approved, so moving to BLS12-381 would raise the security margin but would not on its own satisfy a NIST-approval requirement. BN128 was chosen for its mature tooling and efficient proving on constrained hardware. The relation R_{IFF} carries no curve-specific assumptions, since it is defined entirely over collision-resistant hash functions and Merkle-path verification. Migrating to an approved or institutionally endorsed curve would therefore require regenerating the CRS and proving and verification keys, but not changing the protocol or relation. This is a deployment-level requirement for any operationally certified version of the work.

Finally, the design intentionally makes three trade-offs. Merkle roots were chosen instead of zero-knowledge credential schemes for fast revocation and efficiency on constrained devices, at the cost of making federation membership and policy epoch visible to channel observers in the public inputs. Hardware-derived secret was prioritised over full federation anonymity, so the OD must first identify a candidate AO through pre-classification before the hardware anchor check can run, which makes this a design for attribute privacy. Non-interactive proofs enable multi-hop verification without prover recomputation, which means that proof timing and bundle transmission are observable as traffic metadata. Countermeasures for this are outside the scope of this work.

6

Conclusion

This thesis asked whether non-interactive zero-knowledge proofs can replace identity-based authentication as the basis for privacy-preserving IFF among UAVs in contested environments. Rather than revealing who an asset is, the approach lets an asset prove that its control state is valid without disclosing any underlying detail. To answer the question, the work produced a formal protocol relation R_{IFF}, a complete cross-platform proof-of-concept implementation, and an empirical evaluation covering baseline performance, sensitivity to network impairment, adversarial security, and an analytical assessment of deployability.

6.1. Contributions

This thesis makes four contributions, which together move from the cryptographic construction, to how trust spreads across a swarm, to how the design performs in practice, and finally to what its measured performance means for real encounters.

1. **A joint-circuit, hardware-bound NIZK attestation relation.** R_{IFF} encodes three trust conditions, namely federation membership, possession of an enrolment-bound hardware secret, and policy authorisation, as a single Groth16 proof. Because the circuit reuses $h_{\mathsf{hw}}^{\mathsf{enrol}}$ as a shared wire across Statements 1 and 2, the same hardware secret must satisfy both checks at once. This prevents an adversary from combining credentials taken from different assets, and it does so without revealing any private value to the verifier.
2. **A forwarding-capable, multi-observer verification model.** Because the proof is non-interactive, any observer that holds the verification key and locally provisioned anchor values can re-verify a forwarded proof on its own, without contacting the original prover. This lets a single authentication decision spread across a swarm without any runtime connection to ground infrastructure.
3. **A cross-platform empirical evaluation under impaired network conditions.** The protocol was tested across four operational flight scenarios on both x86-64 and ARM (Raspberry Pi 3B+) hardware, with controlled delay and packet-loss injection and six adversarial test vectors. This provides a latency and reliability characterisation of a joint hardware-bound zero-knowledge IFF relation on a capable constrained platform.
4. **An analytical framework relating authentication latency to encounter feasibility.** The relationship $D_{\mathsf{crit}} = v \cdot T_{\mathsf{cycle}}$ converts measured proving time into the minimum standoff distance needed to finish one authentication cycle at a given closing speed. This turns a raw timing figure into an operational limit and gives future optimisation work a concrete numerical target.

6.2. Answers to the Research Questions

Main Research Question

To what extent can zero-knowledge attestation support IFF decisions among UAVs without relying on identity-based authentication in contested environments?

The evidence shows that zero-knowledge attestation can support UAV IFF decisions without identity-based authentication under the evaluated conditions. The proof itself reveals none of the asset's private values. The exchange provides within-federation unlinkability rather than full anonymity. Asset-level attributes remain hidden, and sessions cannot be linked through the proof by any observer without the enrolment registry. Any association between a proof and a physical airframe arises from the non-cryptographic sensing channel rather than from the proof itself. What the design removes is the need to authenticate by disclosed identity. And it hides the set of private attributes behind the decision. Within that boundary, the approach is operationally viable on x86-64 platforms, while current ARM performance imposes encounter-duration constraints that need further engineering before the same claim can extend to embedded autopilot-class hardware.

On x86-64, the full authentication cycle completes in under 2.2 s across all four evaluated flight scenarios, with no failures at delays up to 200 ms and at least 96% success under 1% per-message packet loss. The verifier cost of 10.8 ms is negligible for any realistic encounter rate. The 137-byte proof fits within a single MAVLink v2 packet, while the full proof-and-public-input bundle (410–413 B) requires two packets; both therefore meet the communication constraints without protocol modification. The relation is sound against all evaluated attack classes.

On ARM (Raspberry Pi 3B+), the prover runs about 14 times slower than on x86-64, which raises the cycle time to 25 to 28 s. At the Transit Encounter baseline speed of 13.7 m/s, this requires a standoff of roughly 350 m to complete a single cycle, rising to over 1 km at 40 m/s. The slowdown appears consistent with the use of GMP's generic arithmetic in the absence of assembly-optimised BN128 support, though this work does not isolate the two factors directly. The protocol is therefore not yet deployable on ARM-class autopilot hardware for fast-encounter geometries, although the deployability assessment suggests it remains feasible for slower encounters.

Three open questions bound the answer, namely prover feasibility on real autopilot-class hardware beyond the evaluated Raspberry Pi, the relay attack identified in Section 5.3 which is structurally possible within the current design, and the latency of revocation in fully denied environments.

SRQ1: Control State Attestation Without Persistent Identifiers

SRQ1 is answered positively. A single proof attests federation membership, possession of an enrolment-bound hardware secret, and policy validity, without revealing any persistent identifier in the proof tuple. Adversarial evaluation confirmed resistance to cross-asset credential composition across all tested attack classes, since the shared-wire binding between Statements 1 and 2 stops an adversary from pairing one asset's identity credential with another asset's hardware credential. The design provides within-federation unlinkability. Asset-level attributes remain hidden, and consecutive proofs from the same asset cannot be linked through the proof by any observer without the enrolment registry.

SRQ2: Computational, Latency, and Communication Constraints

SRQ2 is answered with a platform-dependent finding. On x86-64, all three constraint categories are met within the evaluated envelope. On ARM, computation is the dominant constraint, and the roughly 14 times prover slowdown shifts the critical question from whether a proof verifies once it arrives to whether a proof can be generated and verified inside the encounter window at all. Network delay is a minor factor on ARM by comparison, since its contribution is small next to the absolute prover cost, so reducing delay yields almost no operational benefit while reducing prover cost yields proportional benefits at every closure speed. Communication overhead is the same on both platforms and satisfies the MAVLink constraint throughout.

SRQ3: Multi-Observer Verification Properties

SRQ3 is answered positively within the scope of the evaluation. All four properties hold, namely consistency, locality, forwarding correctness, and availability under degraded connectivity. No disagreement

was recorded across 3,881 sweep cycles; the secondary observer verified every forwarded bundle on its own with no connection to the prover, and tests T3 and T4 confirmed that tampering and cross-CRS forgery are rejected at the algebraic level.

The main unresolved security limitation is relayability. The protocol binds a proof to the prover's enrolled hardware key inside the circuit, but it does not bind the transmission of that proof to the physical platform that generated it. A valid proof can therefore be forwarded by an intermediary adversary without causing verification to fail. This does not affect the correctness of attestation, but it prevents the current design from offering any assurance of proximity, and it remains the most important direction for future protocol work.

6.3. Future Work

The most immediate performance bottleneck is the ARM prover slowdown, which is consistent with GMP relying on a generic arithmetic backend. A direct improvement would be to replace this with assembly-optimised BN128 field arithmetic for Cortex-A53 and Cortex-A72 cores, as already demonstrated in other zk-SNARK systems. This change alone is expected to yield substantial gains. Because prover optimisation reduces only the proving component of the cycle, a conservative $5\times$ improvement applied to the $\sim$14.0 s ARM proving component, holding the 11.65 s non-proving residual fixed, reduces the Transit Encounter D_{crit} from 351 m to approximately 198 m (Section 5.1.2). Among all optimisations considered, this has the most direct effect on the operational envelope, though the non-proving residual bounds the achievable gain: reaching x86-64-class stand-off would require reducing that residual as well.

Once proving performance is improved, the next limiting factor becomes protocol robustness under adversarial communication. In particular, the current design does not explicitly prevent relay or wormhole attacks. Two complementary mitigations are relevant. Distance bounding [17] enforces a physical-layer timing constraint that makes relaying infeasible even if valid proof material is forwarded correctly. Channel binding further restricts validity to a specific communication endpoint, preventing reuse of a proof across different RF links. Both mechanisms, however, introduce additional latency and design complexity, and their interaction with the non-interactive bearer-token structure of R_{IFF} requires careful integration, particularly to ensure that security gains do not undermine timing assumptions.

With these timing and adversarial constraints in place, the choice of proving backend becomes a second-order but still important design axis. Since R_{IFF} avoids elliptic curve and pairing assumptions, it is compatible in principle with post-quantum SNARK constructions. Candidate replacements include STARK-based systems [9] and lattice-based SNARKs [15]. However, because $D_{\text{crit}} = v \cdot T_{\text{cycle}}$ ties proving latency directly to a physical minimum separation distance, backend substitution cannot be evaluated in isolation. It must be assessed under the same four flight scenarios used throughout this work, since even modest changes in proving time translate directly into operational range constraints.

A separate but equally important limitation concerns privacy leakage through public inputs. The values ϕ_{reg} and ϕ_{pol} expose federation membership and policy epoch to any observer, which is undesirable in contested environments. This can be addressed using anonymous credential systems, such as BBS+ signatures [14] or accumulator-based constructions [56]. Both approaches preserve verifiability while concealing membership and policy state. The main trade-off is revocation handling: Merkle-root-based revocation is operationally simple and immediate once the updated root is distributed, whereas accumulator-based schemes require witness updates and more complex synchronisation, which may be challenging in UAV swarm settings where connectivity is intermittent.

Beyond these core architectural directions, several refinements would improve practicality and confidence in deployment. A persistent prover daemon would eliminate per-cycle process startup overhead, producing a bounded but measurable reduction in ARM latency. Formal verification of the nonce-binding and epoch-binding sub-protocols using tools such as ProVerif or Tamarin would strengthen assurance for Claims 3.5.1 and 3.5.2 by providing machine-checked correctness guarantees (Section 3.5.2). In addition, the depth-scaling results in Table 5.2 should be re-evaluated on ARM hardware under combined delay and packet-loss conditions, since the current orthogonal sweep model may underestimate degradation when both effects interact, particularly for larger registry sizes and their impact on D_{crit} and availability.

Finally, more realistic RF conditions combining high latency and high loss should be evaluated jointly rather than independently, as nonlinear effects are likely in contested environments. A further incremental improvement would be to incorporate a credential issuance timestamp into the policy leaf. This would strengthen freshness guarantees by reducing the one-epoch replay window, although its feasibility depends on the accuracy of time synchronisation available across the federation. Relatedly, epoch duration is currently fixed at provisioning, but a deployed system would benefit from allowing mission planners to set it per operation, trading freshness against the expected propagation delay of policy updates in the target environment.

6.4. Broader Impact

This work connects cryptographic authentication with real-time autonomous systems, where correctness guarantees must be interpreted alongside timing, communication constraints, and governance requirements. The implications therefore extend beyond the specific construction to the design of verifiable authentication mechanisms in constrained wireless environments.

From a security perspective, the main contribution is that cross-asset credential composition can be enforced at the relation level through shared-wire binding in the joint circuit. This strengthens the integrity of the control state abstraction introduced earlier, where trust is derived from a composite condition and not the identity alone. More generally, the deployability framework formalises the relationship between cryptographic latency and operational encounter geometry, making it relevant to wireless attestation settings in which verification must occur under strict time and communication constraints. Within this setting, relay attack analysis highlights a structural limitation of non-interactive zero-knowledge authentication: while proofs can certify control state without revealing identity, they do not inherently bind to the physical transmitter, meaning that origin assurance cannot be achieved purely at the cryptographic layer.

These properties also have ethical implications due to the dual-use nature of privacy-preserving authentication in autonomous systems. The same zero-knowledge guarantees that protect coalition assets from adversarial identification also reduce visibility for external oversight. As a result, operational trust is shifted toward the governance of the federation registry and CRS, which define what constitutes a valid control state. The protocol is therefore positioned as a research system requiring independent security evaluation, regulatory approval, and formal governance before deployment. It is not intended to support autonomous kinetic decisions without human oversight. In line with emerging norms on meaningful human control of autonomous systems [57], responsibility for classification outcomes remains with human operators. A further ethical concern arises from degraded communication conditions: conservative failure modes may misclassify legitimate coalition assets as unverified, introducing a false-foe risk that must be explicitly addressed in operational doctrine.

The societal and economic implications follow from the operational gap identified earlier. Existing identification systems depend on infrastructure, connectivity, or identity disclosure, none of which can be guaranteed in contested environments. By enabling verification of control state using a pre-mission distributed trust anchor and lightweight communication (approximately 500 bytes per authentication cycle), the proposed framework supports interoperability across heterogeneous coalition systems without requiring shared hardware platforms or centralised coordination. This directly addresses the requirement for local, autonomous trust decisions in degraded environments while preserving confidentiality of identity and mission context.

The primary limitation to deployment is computational, as the ARM prover latency remains the dominant bottleneck on embedded UAV hardware. This prevents efficient execution on many currently fielded autopilot platforms despite the suitability of the cryptographic design. Addressing this constraint is essential for practical adoption in low-power autonomous systems.

7

Reflection

This chapter reflects on the research process, its relation to the broader Computer Science field, and the implications of the work. It also addresses the use of Generative AI tools in the thesis project, in accordance with TU Delft guidelines.

7.1. Reflection on Research in the Broader CS Field

This work contributes to research at the intersection of privacy-preserving authentication and autonomous systems. It shows that federation membership, hardware provenance, and mission authorisation can be combined in a single attestation without revealing underlying attributes. A recurring issue in applied cryptography is the gap between formal security guarantees and deployability under operational constraints. Latency, communication reliability, and physical encounter dynamics are as important as formal security properties in determining system viability.

The work also contributes to a clearer understanding of what non-interactive zero-knowledge authentication can and cannot achieve. Non-interactive proofs enable forwarding and multi-observer verification without prover involvement, which is advantageous in intermittently connected swarm environments, but also prevents binding a proof to its physical transmitter. This tension between privacy, forwarding capability, and proximity assurance is not specific to this construction but is a structural property of non-interactive anonymous bearer tokens in general. Clarifying this boundary is a contribution to the deployment understanding of NIZK authentication in physical settings.

The work suggests that composing individually secure mechanisms may introduce vulnerabilities if their relationships are not enforced within a single cryptographic statement. This observation is particularly relevant in coalition environments, where trust material is distributed across multiple parties with differing security assumptions.

7.2. Implications and Applicability

R_{IFF} applies to coalition drone operations and distributed systems requiring local trust establishment without continuous connectivity. Similar privacy requirements arise in civilian applications such as autonomous logistics and cooperative robotics, where multiple operators may need to verify each other's authorisation without disclosing sensitive mission or fleet information. The design uses generic primitives, allowing instantiation with future quantum-resistant systems. This may ease migration toward quantum-resistant constructions as such systems mature.

Practical deployment will depend on three conditions being met. First, proving performance on embedded platforms must improve (via optimised arithmetic or hardware support) to meet UAV realistic encounter-time constraints. Second, a mechanism to mitigate relay attacks must be integrated, as the current design cannot provide proximity assurance without additional physical-layer or channel-binding mechanisms. Third, the governance question of who organises and audits the trusted setup ceremony for a multi-national deployment must be resolved at the policy level before the protocol can be opera-

tionally certified.

A key operational risk is the classification of unverified assets. The protocol's conservative design treats any failure to complete authentication as an unverified outcome, which is the correct security posture but introduces fratricide risk for legitimate assets that fail authentication due to link degradation rather than adversarial activity. Operational procedures must account for this asymmetry, and human oversight should remain part of any decision-making process based on trust classifications.

7.3. Personal Learning and Growth

The main challenge was bridging the gap between a theoretically sound design and a working implementation on resource-constrained hardware. While the formal model suggests a direct pipeline from relation definition to circuit compilation, proof generation, and verification, the implementation required a substantial number of engineering decisions whose effects only became visible at runtime. This experience reinforced that applied security research requires sustained engagement with implementation concerns that formal models deliberately abstract away. A related lesson came from the simulation infrastructure: adapting an existing UAV network simulator to the protocol's requirements required substantially more effort than expected, particularly in integrating the authentication harness with the FlyNetSim's messaging system.

The project also changed how I think about evaluation methodology. Different claims require different forms of evidence. Security arguments rely primarily on formal reasoning and adversarial analysis, whereas performance and reliability claims require empirical evaluation, sufficient sample sizes, and careful treatment of variance. Maintaining a clear separation between these evidence types required some discipline throughout the project.

The most significant shift was in how I viewed the relay attack limitation. At first, I considered it a narrow edge case, but I came to realise that it defines a fundamental boundary of what non-interactive anonymous authentication can achieve in wireless environments. Recognising and clearly articulating this boundary became a contribution in itself, rather than simply a limitation to defend against.

Another challenge was preparing the work for submission to the Information Security Conference (ISC 2026). Condensing a full thesis into a page-limited paper required deciding what to emphasise, what to omit, and how to present the claims with enough precision for peer review. This process improved my ability to write concisely and prioritise the most important aspects of the work.

7.4. Use of Generative AI Tools

Three generative AI tools were used during this thesis project.

Claude (Anthropic) was the primary AI tool used. It assisted with improving clarity, consistency of terminology, and structural coherence across chapters. All draft text was critically reviewed and substantially revised by the author before incorporation. It was not used to generate research findings, experimental results, security arguments, or formal proofs. Claude Code was used to assist with the Python orchestration layer, the experimental harness, the data analysis and plotting scripts, and the architecture diagrams. All code was reviewed, tested, and validated.

ChatGPT (OpenAI) was used occasionally for editorial feedback on argument structure and clarity. No text produced by ChatGPT was incorporated into the thesis.

Anara was used as a reading aid for research papers already identified through the author's own literature search, helping to locate specific definitions, figures, or results within long documents. All literature identification, evaluation, and citation was performed by the author independently.

All AI-assisted content was verified before use. Writing assistance was verified by re-reading and cross-checking against source material. Code assistance was verified through review and comparison against independently computed reference values. No AI tool was used to generate experimental data, statistical analysis, formal proofs, or research claims. All AI-generated suggestions were treated as drafts requiring independent verification.

References

[1] Piyush Agarwal, Sachin Sharma, and Priya Matta. “Security techniques in Unmanned Air Traffic Management System”. In: *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, Jan. 2023, pp. 641–646. DOI: 10.1109/icssit55814.2023.10061115.

[2] Ridwane Aissaoui et al. “Evaluating post-quantum key exchange mechanisms for UAV communication security”. In: *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*. IEEE, Sept. 2024, pp. 1–10. DOI: 10.1109/dasc62030.2024.10749304.

[3] Nitish Andola et al. “Spychain: A lightweight blockchain for authentication and anonymous authorization in IOD”. In: *Wireless Personal Communications* 119.1 (Feb. 2021), pp. 343–362. DOI: 10.1007/s11277-021-08214-8.

[4] Frederik Armknecht et al. “A Formalization of the Security Features of Physical Functions”. In: *2011 IEEE Symposium on Security and Privacy*. IEEE, 2011, pp. 397–412. DOI: 10.1109/SP.2011.10.

[5] Yucel Aydin et al. “Group authentication for drone swarms”. In: *2021 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, Oct. 2021, pp. 72–77. DOI: 10.1109/wisee50203.2021.9613831.

[6] Edoukou Berenger Ayebie et al. “Decentralized Post-quantum Authentication Protocol for Drone Communication Networks”. In: *2025 IEEE/ACS 22nd International Conference on Computer Systems and Applications (AICCSA)*. IEEE, Oct. 2025, pp. 1–5. DOI: 10.1109/aiccsa66935.2025.11315326.

[7] Sabur Baidya, Zoheb Shaikh, and Marco Levorato. “FlyNetSim: An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot”. In: *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM. 2018, pp. 37–45.

[8] Michael Bailey et al. “The Menlo Report”. In: *2012 IEEE Symposium on Security and Privacy Workshops*. IEEE, 2012, pp. 58–73. DOI: 10.1109/SPW.2012.25.

[9] Eli Ben-Sasson et al. *Scalable, Transparent, and Post-Quantum Secure Computational Integrity*. IACR Cryptology ePrint Archive, Report 2018/046. 2018. URL: https://eprint.iacr.org/2018/046.

[10] Jens Bender et al. “The PACE|AA protocol for Machine Readable Travel Documents, and its security”. In: *Financial Cryptography and Data Security — FC 2012*. Lecture Notes in Computer Science. Springer, 2012, pp. 344–358. DOI: 10.1007/978-3-642-32946-3_25.

[11] Basudeb Bera, Ashok Kumar Das, and Biplab Sikdar. “Quantum Resistant Lattice-Based Access Control Scheme for UAV-Assisted Internet-of-Drones Applications”. In: *IEEE Transactions on Mobile Computing* 25.3 (2026), pp. 4077–4090. DOI: 10.1109/TMC.2025.3622654.

[12] Basudeb Bera et al. “Access control protocol for battlefield surveillance in drone-assisted IoT environment”. In: *IEEE Internet of Things Journal* 9.4 (Feb. 2022), pp. 2708–2721. DOI: 10.1109/jiot.2020.3049003.

[13] Manuel Blum, Paul Feldman, and Silvio Micali. “Non-Interactive Zero-Knowledge and Its Applications”. In: *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[14] Dan Boneh, Xavier Boyen, and Hovav Shacham. “Short Group Signatures”. In: *Advances in Cryptology — CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 41–55. DOI: 10.1007/978-3-540-28628-8_3.

[15] Cecilia Boschini et al. “Efficient Post-quantum SNARKs for RSIS and RLWE and their applications to privacy”. In: *Post-Quantum Cryptography — PQCrypto 2020*. Lecture Notes in Computer Science. Springer, 2020, pp. 247–267. DOI: 10.1007/978-3-030-44223-1_14.

[16] Sean Bowe, Ariel Gabizon, and Matthew D. Green. “A Multi-Party Protocol for Constructing the Public Parameters of the Pinocchio zk-SNARK”. In: *Financial Cryptography and Data Security — FC 2018 Workshops*. Vol. 10958. Lecture Notes in Computer Science. Springer, 2019, pp. 64–77. DOI: 10.1007/978-3-662-58820-8_5.

[17] Stefan Brands and David Chaum. “Distance-Bounding Protocols”. In: *Advances in Cryptology — EUROCRYPT ’93*. Vol. 765. Lecture Notes in Computer Science. Springer, 1994, pp. 344–359. DOI: 10.1007/3-540-48285-7_30.

[18] Jan Camenisch and Anna Lysyanskaya. “An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation”. In: *Advances in Cryptology — EUROCRYPT 2001*. Ed. by Birgit Pfitzmann. Vol. 2045. Lecture Notes in Computer Science. Springer, 2001, pp. 93–118. DOI: 10.1007/3-540-44987-6_7.

[19] Dharmendra Chauhan et al. “Nation’s defense: A comprehensive review of anti-drone systems and Strategies”. In: *IEEE Access* 13 (2025), pp. 53476–53505. DOI: 10.1109/access.2025.3550338.

[20] DJI. *Matrice 300 RTK Support*. 2026. URL: https://www.dji.com/nl/support/product/matrice-300 (visited on 07/05/2026).

[21] European Commission. *Action Plan on Drone and Counter-Drone Security*. European Commission. Feb. 2026. URL: https://digital-strategy.ec.europa.eu/en/library/action-plan-drone-and-counter-drone-security.

[22] European Commission. *New Plan to Counter Drone Threats*. European Commission. Feb. 2026. URL: https://commission.europa.eu/news-and-media/news/new-plan-counter-drone-threats-2026-02-11_en.

[23] European Commission. *Recommendation on a Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography*. Tech. rep. European Commission, Apr. 2024. URL: https://digital-strategy.ec.europa.eu/en/library/recommendation-coordinated-implementation-roadmap-transition-post-quantum-cryptography.

[24] FIDO Alliance and World Wide Web Consortium. *Web Authentication: An API for Accessing Public Key Credentials (WebAuthn) Level 1*. W3C Recommendation. Mar. 2019. URL: https://www.w3.org/TR/webauthn-1/.

[25] Ihor Filimonov et al. “Breaking unlinkability of the ICAO 9303 standard for e-passports using bisimilarity”. In: *Computer Security — ESORICS 2019*. Lecture Notes in Computer Science. Springer, 2019, pp. 577–594. DOI: 10.1007/978-3-030-29959-0_28.

[26] Frank Gardner. *How realistic is the plan to build a “drone wall” against Russia?* Nov. 2025. URL: https://www.bbc.com/news/articles/crkl3d6pegpo.

[27] Blaise Gassend et al. “Silicon Physical Random Functions”. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*. ACM, 2002, pp. 148–160. DOI: 10.1145/586110.586132.

[28] Rosario Gennaro et al. “Quadratic Span Programs and Succinct NIZKs without PCPs”. In: *Advances in Cryptology — EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 626–645. DOI: 10.1007/978-3-642-38348-9_37.

[29] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. “The Knowledge Complexity of Interactive Proof Systems”. In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208. DOI: 10.1137/0218012.

[30] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. “A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks”. In: *SIAM Journal on Computing* 17.2 (1988), pp. 281–308. DOI: 10.1137/0217017.

[31] Michael T. Goodrich, Roberto Tamassia, and Jasminka Hasic. “An Efficient Dynamic and Distributed Cryptographic Accumulator”. In: *Information Security — ISC 2002*. Vol. 2433. Lecture Notes in Computer Science. Springer, 2002, pp. 372–388. DOI: 10.1007/3-540-45811-5_29.

[32] Lorenzo Grassi et al. “Poseidon: A New Hash Function for Zero-Knowledge Proof Systems”. In: *30th USENIX Security Symposium*. USENIX Association, 2021, pp. 519–535. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/grassi.

[33] Jens Groth. “On the Size of Pairing-Based Non-interactive Arguments”. In: *Advances in Cryptology — EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 305–326. DOI: 10.1007/978-3-662-49896-5_11.

[34] Linlin He et al. “A Post-Quantum Authentication and key agreement scheme for drone swarms”. In: *Electronics* 14.17 (Aug. 2025), p. 3364. DOI: 10.3390/electronics14173364.

[35] Hannah Hollander. *What is grey zone deterrence?* Feb. 2021. URL: https://ras-nsa.ca/what-is-grey-zone-deterrence/.

[36] Hans Niklas Jacob et al. “faultTPM: Exposing AMD fTPMs’ Deepest Secrets”. In: *32nd USENIX Security Symposium*. USENIX Association, 2023, pp. 1835–1852. URL: https://arxiv.org/abs/2304.14717.

[37] Muhammad Asghar Khan et al. “Future-proofing security for UAVs with Post-Quantum Cryptography: A Review”. In: *IEEE Open Journal of the Communications Society* 5 (2024), pp. 6849–6871. DOI: 10.1109/ojcoms.2024.3486649.

[38] Jane Kim et al. “PqRID: Compact, anonymous remote identification and post-quantum authentication for Drones”. In: *IEEE Internet of Things Journal* (2026), pp. 1–15. DOI: 10.1109/jiot.2025.3650221.

[39] Taechan Kim and Razvan Barbulescu. “Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case”. In: *Advances in Cryptology – CRYPTO 2016*. Vol. 9814. Lecture Notes in Computer Science. Springer, 2016, pp. 543–571. DOI: 10.1007/978-3-662-53018-4_20.

[40] Athanasios Koulianos et al. “Enhancing Unmanned Aerial Vehicle Security: A Zero-knowledge proof approach with zero-knowledge succinct non-interactive arguments of knowledge for authentication and location proof”. In: *Sensors* 24.17 (Sept. 2024). DOI: 10.3390/s24175838.

[41] Leslie Lamport. “The Part-Time Parliament”. In: *ACM Transactions on Computer Systems* 16.2 (1998), pp. 133–169. DOI: 10.1145/279227.279229.

[42] Leslie Lamport, Robert Shostak, and Marshall Pease. “The Byzantine Generals Problem”. In: *ACM Transactions on Programming Languages and Systems* 4.3 (1982), pp. 382–401. DOI: 10.1145/357172.357176.

[43] Ingo Liersch. “Electronic passports – from secure specifications to secure implementations”. In: *Information Security Technical Report* 14.2 (May 2009), pp. 96–100. DOI: 10.1016/j.istr.2009.06.005.

[44] MAVLink Development Team. *MAVLink Micro Air Vehicle Communication Protocol*. https://mavlink.io. 2024.

[45] Ralph C. Merkle. “A Digital Signature Based on a Conventional Encryption Function”. In: *Advances in Cryptology — CRYPTO ’87*. Ed. by Carl Pomerance. Lecture Notes in Computer Science. Springer, 1988. DOI: 10.1007/3-540-48184-2_32.

[46] Emmanouel T. Michailidis and Demosthenes Vouyioukas. “A Review on Software-Based and Hardware-Based Authentication Mechanisms for the Internet of Drones”. In: *Drones* 6.2 (Feb. 2022). DOI: 10.3390/drones6020041. URL: https://doi.org/10.3390/drones6020041.

[47] Daniel Moghimi et al. “TPM-FAIL: TPM Meets Timing and Lattice Attacks”. In: *29th USENIX Security Symposium*. USENIX Association, 2020, pp. 2057–2073. URL: https://tpm.fail.

[48] National Institute of Standards and Technology. *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard; FIPS 204: Module-Lattice-Based Digital Signature Standard; FIPS 205: Stateless Hash-Based Digital Signature Standard*. Tech. rep. National Institute of Standards and Technology, Aug. 2024. URL: https://csrc.nist.gov/pqc-standardization.

[49] Shayesta Naziri et al. *ZAPS: A Zero-Knowledge Proof Protocol for Secure UAV Authentication with Flight Path Privacy*. arXiv:2508.17043 [cs.CR]. 2025. arXiv: 2508.17043 [cs.CR]. URL: https://arxiv.org/abs/2508.17043.

[50] NIST. *Sybil Attack*. National Institute of Standards and Technology (NIST). URL: https://csrc.nist.gov/glossary/term/sybil_attack.

[51] Ravikanth Pappu et al. "Physical One-Way Functions". In: *Science* 297.5589 (2002), pp. 2026–2030. DOI: 10.1126/science.1074376.

[52] Rohini Poolat Parameswarath and Biplab Sikdar. "Privacy-Preserving Mutual Authentication Protocol for Drone Delivery Services". In: *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. 2023, pp. 2166–2171. DOI: 10.1109/GLOBECOM54140.2023.10436751.

[53] Lukas Petzi, Alexandra Dmitrienko, and Ivan Visconti. "PUF-based authentication in IoT against strong physical adversary using zero-knowledge proofs". In: *2024 IEEE Security and Privacy Workshops (SPW)*. IEEE, May 2024, pp. 312–319. DOI: 10.1109/spw63631.2024.10918871.

[54] Sérgio Ramos, Tiago J. Cruz, and Paulo Simoes. "Security and safety of Unmanned Air Vehicles: An overview". In: *Proceedings of the European Conference on Information Warfare and Security*. June 2021. DOI: 10.34190/ews.21.027.

[55] Parya Derakhshan Roodsari, Siavash Bayat Sarmadi, and Hatame Mosanaei Boorani. "Post-quantum authentication and communication security for drone networks". In: *IEEE Transactions on Dependable and Secure Computing* (2025), pp. 1–12. DOI: 10.1109/tdsc.2025.3636937.

[56] Michael Rosenberg et al. "ZK-creds: Flexible anonymous credentials from zkSNARKs and existing identity infrastructure". In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2023, pp. 790–808. DOI: 10.1109/sp46215.2023.10179430.

[57] Stefka Schmid, Thea Riebe, and Christian Reuter. "Dual-Use and Trustworthy? A Mixed Methods Analysis of AI Diffusion Between Civilian and Defense R&D". In: *Science and Engineering Ethics* 28.2 (2022), p. 15. DOI: 10.1007/s11948-022-00364-7.

[58] Savio Sciancalepore. "Privacy and confidentiality issues in drone operations: Challenges and road ahead". In: *IEEE Network* 38.6 (Nov. 2024), pp. 227–233. DOI: 10.1109/mnet.2024.3432730.

[59] SCIPR Lab. *libsnark: A C++ Library for zkSNARK Proofs*. GitHub. 2014. URL: https://github.com/scipr-lab/libsnark.

[60] Muhammet A. Sen, Saba Al-Rubaye, and Antonios Tsourdos. "Design of Secure Communication Networks for UAV platform empowered by Lightweight Authentication Protocols". In: *Electronics* 15.4 (Feb. 2026), p. 785. DOI: 10.3390/electronics15040785.

[61] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.

[62] Nigel P. Smart. *Cryptography Made Simple*. Information Security and Cryptography. Springer, 2016. ISBN: 978-3-319-21935-6. DOI: 10.1007/978-3-319-21936-3.

[63] Tore Steiro et al. "Toward Federated Mission Networking in the Tactical Domain". In: *IEEE Communications Magazine* 53.10 (2015), pp. 52–57. DOI: 10.1109/MCOM.2015.7295463.

[64] Emanuele Strieder, Christoph Frisch, and Michael Pehl. "Machine learning of physical unclonable functions using helper data". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (Feb. 2021), pp. 1–36. DOI: 10.46586/tches.v2021.i2.1-36.

[65] Daryna Sydorenko. *Moscow's creeping escalation: Hybrid pressure on Poland and the Alliance*. Nov. 2025. URL: https://warsawinstitute.org/moscows-creeping-escalation-hybrid-pressure-on-poland-and-the-alliance/.

[66] Nawar Hayder Tawfeeq et al. "Lightweight and quantum-resistant authentication for the internet of drones (IOD) using dilithium signatures." In: *International Journal of Innovative Research and Scientific Studies* 8.2 (Apr. 2025), pp. 2842–2853. DOI: 10.53894/ijirss.v8i2.5825.

[67] The MITRE Corporation. *MITRE ATT&CK for Industrial Control Systems*. 2023. URL: https://attack.mitre.org/matrices/ics/.

[68] The MITRE Corporation. *T0814: Denial of Service*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0814/.

[69] The MITRE Corporation. *T0816: Device Restart/Shutdown*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0816/.

[70] The MITRE Corporation. *T0830: Adversary-in-the-Middle*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0830/.

[71] The MITRE Corporation. *T0842: Network Sniffing*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0842/.

[72] The MITRE Corporation. *T0849: Masquerading*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0849/.

[73] The MITRE Corporation. *T0856: Spoof Reporting Message*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0856/.

[74] The MITRE Corporation. *T0862: Supply Chain Compromise*. MITRE ATT&CK for ICS. 2023. URL: https://attack.mitre.org/techniques/T0862/.

[75] Trusted Computing Group. *TPM Library Specification, Family 2.0*. Tech. rep. Published as ISO/IEC 11889. Trusted Computing Group, 2019. URL: https://trustedcomputinggroup.org/resource/tpm-library-specification/.

[76] Vilmantas Venckūnas. *Lithuania's govt declares nationwide emergency over smuggler balloons from Belarus*. Dec. 2025. URL: https://www.lrt.lt/en/news-in-english/19/2770083/lithuania-s-govt-declares-nationwide-emergency-over-smuggler-balloons-from-belarus.

[77] Marcus Walshe et al. "Non-interactive zero knowledge proofs for the authentication of IoT devices in reduced connectivity environments". In: *Ad Hoc Networks* 95 (Dec. 2019), p. 101988. DOI: 10.1016/j.adhoc.2019.101988.

[78] Wei Wu et al. "Blockchain based zero-knowledge proof of location in IoT". In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE, June 2020, pp. 1–7. DOI: 10.1109/icc40277.2020.9149366.

[79] Tao Xia et al. "A quantum-resistant identity authentication and key agreement scheme for UAV Networks based on Kyber algorithm". In: *Drones* 8.8 (July 2024), p. 359. DOI: 10.3390/drones8080359.

[80] Jean-Paul Yaacoub et al. "Security analysis of drones systems: Attacks, Limitations, and recommendations". In: *Internet of Things* 11 (Jan. 2020), p. 100218. DOI: 10.1016/j.iot.2020.100218.

[81] Muktar Yahuza et al. "Internet of drones security and privacy issues: Taxonomy and open challenges". In: *IEEE Access* 9 (Apr. 2021), pp. 57243–57270. DOI: 10.1109/access.2021.3072030.

[82] Shuo Zhang et al. "A Lightweight Authentication Protocol for UAVs Based on ECC Scheme". In: *Drones* 7.5 (May 2023), p. 315. DOI: 10.3390/drones7050315.

A

Requirements Traceability

This appendix maps each system requirement from Section 1.2.1 to the evidence.

Requirement	Description	Evidence
Functional Requirements		
FR-1	Local Trust Establishment	No GCS contact required; all verification uses locally provisioned anchor values (Section 5.1.3)
FR-2	Control State Verification	Protocol encodes federation membership (Statement 1), hardware binding (Statement 2), and policy (Statement 3) jointly in R_{IFF} (Section 3.3)
FR-3	Revocation Support	Merkle root updates invalidate removed assets (Section 3.3)
FR-4	Multi-Observer Forwarding	Non-interactive proofs enable forwarding; secondary observer verification tested in SRQ3 evaluation with 100% re-verification success across 3,881 cycles (Section 5.1.3)
FR-5	Standardisation Compatibility	137-byte proof fits in two MAVLink v2 packets; measured communication cost in Section 5.1.2
Security Requirements		
SR-1	Replay Resistance	Nonce bound to proof's public input; plaintext equality check prevents cross-session replay at issuing verifier (T2); deferred replay to secondary observers acknowledged as unmitigated in Section 5.3
SR-2	Unlinkability	Session-specific ϕ_{nonce} and ϕ_{hw} provide cryptographic unlinkability; fixed roots ϕ_{reg}, ϕ_{pol}, ϕ_{epoch} are observable but do not reveal asset identity (Section 5.1.1)
SR-3	Soundness Under Impersonation	Joint circuit with shared $h_{\mathsf{hw}}^{\mathsf{enrol}}$ wire prevents cross-asset credential composition; empirically validated by T4 (cross-CRS rejection) and T6 (mismatched witness rejection) (Section 4.3.8)
SR-4	Hardware Binding	Binding enforced at circuit level and evaluated as an algebraic property via T6; hardware secret is simulated, not derived from PUF (Section 5.1.1)
SR-5	Context Binding	Epoch check prevents reuse across expired policy windows (T5); sector and role are encoded in the policy Merkle leaf
SR-6	Post-Quantum Readiness	R_{IFF} defined entirely over hash functions and Merkle paths with no pairing assumptions (Section 3.3); current instantiation uses Groth16/BN128 and is not post-quantum (Section 5.3)
SR-7	Zero-Knowledge Privacy	Groth16 zero-knowledge property guarantees witness hiding under computational hardness assumption (Section 2.4)
Operational Requirements		
OR-1	Verification Latency	10.8 ms on x86-64, 58 ms on ARM; negligible relative to 2.2 s (x86) and 25–28 s (ARM) end-to-end cycle time (Section 5.1.2)
OR-2	Communication Overhead	137-byte proof plus public input fits in 2 MAVLink v2 packets (410–413 bytes raw, 297 bytes binary) (Section 5.1.2)
OR-3	Resource Compatibility	x86-64 achieves 2.2 s completion, compatible with encounter windows; ARM requires 25–28 s, infeasible above 7.8 m/s (Section 5.1.2)
OR-4	Infrastructure Independence	No external connectivity required; all trust decisions use provisioned anchors; design validated in local-only evaluation (Section 5.1.3)
OR-5	Degraded Connectivity Tolerance	Success rates 96%+ at 1% loss on x86-64, 98%+ on ARM; packet loss treated as cycle timeout, not protocol failure (Section 5.1.2)
OR-6	Heterogeneous Interoperability	Multi-nation federation supported via shared registry and policy roots provisioned pre-mission (Section 3.3)

B

Critical Encounter Distance Tables

This appendix reports the critical separation distance $D_{\text{crit}} = v \cdot T_{\text{cycle}}$ for both platforms across all closure speeds, referenced from Section 5.1.2. D_{crit} is the minimum starting separation at which at least one complete authentication cycle fits before two platforms closing at speed v meet, evaluated from 5 m/s (loitering ISR) to 40 m/s (fast intercept). In the table below, T_{cycle} is the mean end-to-end latency over PASS-only trials at $l = 0\%$ packet loss, and the bold column is the Transit Encounter baseline speed (13.7 m/s). The ARM figures use $n = 140$ trials per condition, except Proximate Hold at $d = 50$, which uses $n = 141$. The corresponding speed sweep is plotted in Figures 5.1 and 5.2. This is an analytical assessment derived from the empirical latency measurements of Section 5.1.2, not a new experiment.

Scenario	d (ms)	T_{cycle} (s)	D_{crit} (m) at closure speed v (m/s)						
			5	10	**13.7**	20	25	30	40
x86									
Proximate Hold	0	1.831	9.2	18.3	**25.1**	36.6	45.8	54.9	73.2
	50	1.975	9.9	19.7	**27.1**	39.5	49.4	59.2	79.0
	100	2.129	10.6	21.3	**29.2**	42.6	53.2	63.9	85.2
	200	2.432	12.2	24.3	**33.3**	48.6	60.8	73.0	97.3
Transit Encounter	0	1.995	10.0	19.9	**27.3**	39.9	49.9	59.8	79.8
	50	2.146	10.7	21.5	**29.4**	42.9	53.7	64.4	85.8
	100	2.291	11.5	22.9	**31.4**	45.8	57.3	68.7	91.6
	200	2.591	13.0	25.9	**35.5**	51.8	64.8	77.7	103.7
Mission Sep.	0	2.115	10.6	21.1	**29.0**	42.3	52.9	63.4	84.6
	50	2.241	11.2	22.4	**30.7**	44.8	56.0	67.2	89.6
	100	2.396	12.0	24.0	**32.8**	47.9	59.9	71.9	95.8
	200	2.699	13.5	27.0	**37.0**	54.0	67.5	81.0	108.0
Joint Operation	0	2.067	10.3	20.7	**28.3**	41.3	51.7	62.0	82.7
	50	2.184	10.9	21.8	**29.9**	43.7	54.6	65.5	87.4
	100	2.331	11.7	23.3	**31.9**	46.6	58.3	69.9	93.2
	200	2.634	13.2	26.3	**36.1**	52.7	65.8	79.0	105.3
ARM									
Proximate Hold	0	25.622	128.1	256.2	**351.0**	512.4	640.6	768.7	1024.9
	50	26.307	131.5	263.1	**360.4**	526.1	657.7	789.2	1052.3
	100	27.266	136.3	272.7	**373.5**	545.3	681.7	818.0	1090.7
	200	27.557	137.8	275.6	**377.5**	551.1	688.9	826.7	1102.3
Transit Encounter	0	26.160	130.8	261.6	**358.4**	523.2	654.0	784.8	1046.4
	50	27.278	136.4	272.8	**373.7**	545.6	681.9	818.3	1091.1
	100	27.425	137.1	274.2	**375.7**	548.5	685.6	822.7	1097.0
	200	27.728	138.6	277.3	**379.9**	554.6	693.2	831.8	1109.1
Mission Sep.	0	26.356	131.8	263.6	**361.1**	527.1	658.9	790.7	1054.3
	50	27.369	136.8	273.7	**375.0**	547.4	684.2	821.1	1094.7
	100	27.229	136.1	272.3	**373.0**	544.6	680.7	816.9	1089.2
	200	27.491	137.5	274.9	**376.6**	549.8	687.3	824.7	1099.6
Joint Operation	0	26.529	132.6	265.3	**363.4**	530.6	663.2	795.9	1061.2
	50	26.911	134.6	269.1	**368.7**	538.2	672.8	807.3	1076.5
	100	26.983	134.9	269.8	**369.7**	539.7	674.6	809.5	1079.3
	200	26.887	134.4	268.9	**368.3**	537.7	672.2	806.6	1075.5