

Article

Secure and Decentralised Swarm Authentication Using Hardware Security Primitives

Sagir Muhammad Ahmad  and Barmak Honarvar Shakibaei Asli * 

Faculty of Engineering and Applied Sciences, Cranfield University, Cranfield MK43 0AL, UK;
sagir.ahmad@cranfield.ac.uk

* Correspondence: barmak@cranfield.ac.uk

Abstract

Autonomous drone swarms are increasingly deployed in critical domains such as infrastructure inspection, environmental monitoring, and emergency response. While their distributed operation enables scalability and resilience, it also introduces new vulnerabilities, particularly in authentication and trust establishment. Conventional cryptographic solutions, including public key infrastructures (PKI) and symmetric key protocols, impose computational and connectivity requirements unsuited to resource-constrained and external infrastructure-free swarm deployments. In this paper, we present a decentralized authentication scheme rooted in hardware security primitives (HSPs); specifically, Physical Unclonable Functions (PUFs) and True Random Number Generators (TRNGs). The protocol leverages master-initiated token broadcasting, iterative HSP seed evolution, randomized response delays, and statistical trust evaluation to detect cloning, replay, and impersonation attacks without reliance on centralized authorities or pre-distributed keys. Simulation studies demonstrate that the scheme achieves lightweight operation, rapid anomaly detection, and robustness against wireless interference, making it well-suited for real-time swarm systems.

Keywords: swarm drones; PUF; decentralized; lightweight

1. Introduction

Autonomous drone swarms are rapidly becoming a cornerstone of next-generation cyber-physical systems, particularly in the domains of critical infrastructure testing, inspection, and remote surveillance. These systems offer substantial benefits in terms of scalability, responsiveness, and cost-effectiveness, enabling coordinated missions in environments that are hazardous, large-scale, or logistically complex [1,2]. However, as drone swarms become more autonomous and mission-critical, ensuring the security and legitimacy of each node within the swarm becomes paramount. Unlike conventional networks, swarm drones operate in highly distributed, resource-constrained, and potentially adversarial environments. They must communicate, authenticate, and coordinate without constant supervision or persistent connectivity to trusted authorities. Traditional security mechanisms—such as centralized key distribution, PKI, or symmetric encryption protocols—pose significant limitations in this context. These techniques are not only computationally demanding but also vulnerable to single points of failure and synchronization challenges [3,4]. More critically, they offer limited protection against hardware-level attacks, such as cloning or physical tampering, which are particularly relevant in field-deployable systems [4]. To address these challenges, we explore a fundamentally different approach based on HSPs—notably,



Academic Editors: Umberto Papa,
Giuseppe Del Core, Salvatore Ponte
and Gennaro Ariante

Received: 24 December 2025

Revised: 13 January 2026

Accepted: 17 January 2026

Published: 18 January 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

PUFs and TRNGs. HSPs leverage manufacturing-induced physical randomness to generate device-unique, non-replicable responses that are lightweight, tamper-evident, and naturally suited for embedded environments. These primitives have shown promise in secure identification and key generation in IoT and embedded systems [5,6]. However, their potential to enable decentralized authentication protocols in cooperative mobile drone networks remains underexplored [7].

The novelty of this work lies in synthesizing HSPs (PUFs/TRNGs) with statistical, multi-round trust evaluation in a decentralized protocol. Unlike prior PUF-based schemes designed primarily for static networks, our approach introduces iterative seed evolution and randomized response delays to realize a dynamic, lightweight authentication mechanism tailored to the scale, mobility, and adversarial setting of autonomous drone swarms, operating without external trusted infrastructure or pre-shared keys.

In this work, we propose an authentication scheme tailored for HSP-equipped swarm drones. The protocol follows a master-initiated, multi-round challenge–response process in which each drone uses its internal HSP and an evolving cryptographic seed to compute device-consistent delays and round-bound responses. By monitoring response patterns across rounds, the swarm derives statistical trust and detects anomalies such as cloning, replay, or impersonation attempts without relying on heavyweight cryptography or external verification services. This approach offers several key advantages:

- It enables fully decentralized authentication, eliminating dependence on trusted third parties.
- It provides inherent resistance to cloning and replay attacks, due to the unique and time-evolving nature of the HSP responses.
- It ensures low computational overhead, making it feasible for real-time deployment on low-power microcontrollers.
- It supports scalability, with performance demonstrated up to dozens of swarm participants.

The primary contributions of this study are as follows:

- We develop a lightweight authentication protocol that leverages HSP-based randomness and multi-round challenge–response interactions to establish mutual trust within the swarm. This provides a conceptual overview of this core challenge–response mechanism between a master and slave drones, where the slave’s response is intrinsically tied to its unclonable hardware.
- We develop a mathematical model of the protocol, capturing seed evolution, randomized delay generation, and statistical response validation.
- We conduct extensive simulations to evaluate the protocol’s performance in terms of scalability, tamper detection, and computational cost.
- We benchmark our approach against classical RSA-based and static PUF-only schemes, demonstrating its advantages in real-time, swarm-oriented scenarios.

The remainder of the paper is structured as follows: Section 2 reviews related work in drone security and hardware-based authentication. Section 3 describes the research methods and methodology used in the development of the paper, including the main research question and the overall evaluation approach. Section 4 formulates the problem and outlines our design objectives. Section 5 describes the proposed authentication protocol in detail, while Section 6 presents its mathematical underpinnings. Section 7 reports the simulation results, and Section 8 discusses the security properties and benchmarking evaluation. Section 9 outlines the limitations and directions for future work, followed by concluding remarks.

2. Background and Related Work

Swarm drone technology has advanced swiftly, enabling applications in surveillance, emergency response, precision farming, and monitoring infrastructure [8]. These systems utilize collective intelligence and distributed coordination to perform tasks that would be impossible for a single UAV [7]. However, this shift in approach brings forth security issues that are markedly different from those in traditional networked systems [9]. In contrast to static or centrally controlled networks, drone swarms operate in highly dynamic, external infrastructure-free environments where nodes are mobile, intermittently connected, and vulnerable to unauthorized access [10]. These traits heighten the risk of identity spoofing, cloning, and unauthorized involvement, as traditional authentication models are built on the assumption of stable connectivity and centralized trust points [11]. Additionally, the inherent resource limitations of UAV platforms, such as restricted power, memory, and processing capabilities, limit the practicality of robust cryptographic protocols [10,12]. Before tackling these obstacles, it is crucial to comprehend how operational characteristics specific to swarms, like decentralized control, ad hoc changes in topology, and real-time coordination needs, influence the development of authentication and trust frameworks [11].

2.1. Security Challenges in Swarm Drone Systems

Swarm drones represent a highly dynamic subset of cyber–physical systems (CPS), designed to operate autonomously and cooperatively in decentralized environments [8]. However, their distributed nature introduces complex security challenges [12,13]. Unlike traditional networks, where nodes are often stationary and operate under central supervision, swarm drones are mobile, intermittently connected, and physically exposed, increasing their susceptibility to attacks such as a lack of persistent infrastructure intermittent communication links resource constraints, and physical exposure [14,15]. Figure 1 summarizes key constraints and threat vectors relevant to drone swarms and the Internet of Drones (IoD), including limited battery/computation/storage, ad hoc network formation, high mobility, and attacks such as eavesdropping, spoofing, replay, and DoS.

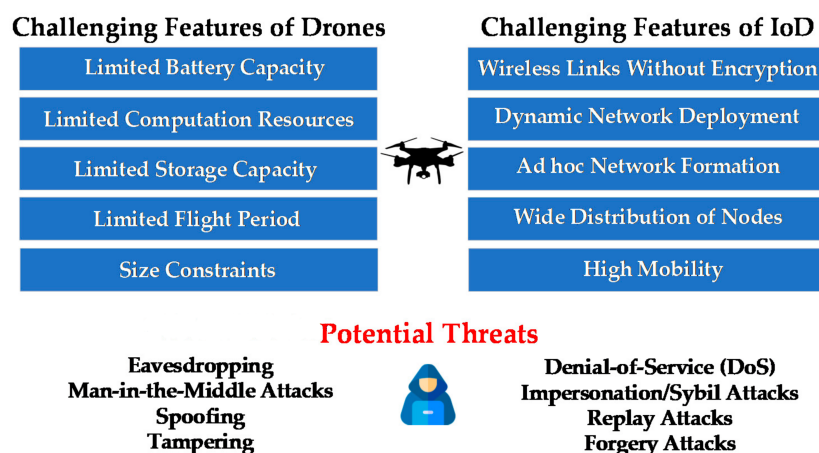


Figure 1. Challenging features and potential threats in drone and IoD environments [16].

Key vulnerabilities arise from a lack of persistent infrastructure (lack of centralized authentication authority), intermittent communication links, resource constraints in power, memory, and computational capacity, and physical exposure that allows adversaries to capture, reverse-engineer, or clone individual drones [11,12]. As swarm drones are increasingly used in critical missions such as infrastructure inspection, emergency response, or military reconnaissance, the need for resilient, real-time, and decentralized authentication mechanisms becomes imperative [17].

2.2. Existing Authentication Approaches

Numerous security frameworks have been suggested to secure communication and identification within UAV networks, each designed to maintain confidentiality, integrity, and authenticity of the information exchanged [18,19]. These methods vary in their foundational assumptions, computational demands, and resilience against adversarial scenarios [20]. Generally, they can be classified into several groups based on their design concepts and trust models, as illustrated in Figure 2.

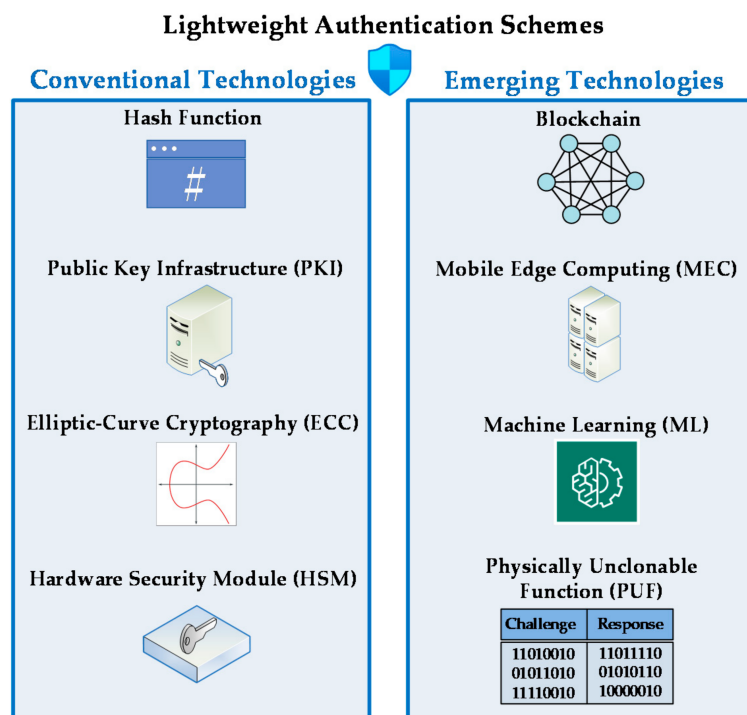


Figure 2. Classification of lightweight authentication schemes: conventional technologies versus emerging technologies [16].

2.2.1. Centralized Public Key Infrastructure

Conventional methods for UAV authentication often rely on certificate-based public key cryptography, executed through PKI [16]. PKI offers strong cryptographic guarantees by facilitating secure key exchanges and identity verification using digital certificates [21]. Nevertheless, its applicability in swarm drone contexts is significantly constrained due to various reasons [16]. Firstly, PKI entails considerable computational and storage demands, which are impractical for low-power flight controllers with limited processing power and memory resources [22]. Secondly, PKI presumes persistent connectivity with certificate authorities for the issuance and revocation of certificates, an assumption that does not hold in mobile, ad hoc drone swarms operating in external infrastructure-free settings [23]. Thirdly, the centralized structure of PKI creates a single point of vulnerability; the compromise or inaccessibility of the certificate authority can disrupt authentication across the entire swarm [21]. These drawbacks render PKI-based solutions inappropriate for highly dynamic, decentralized UAV networks where resilience, scalability, and real-time functionality are essential [17,24].

2.2.2. Symmetric Key-Based Protocols

Symmetric encryption is often considered as a viable alternative to public key cryptography in UAV networks due to its lower computational demands and diminished energy consumption [25]. These features make it appealing for resource-limited drones that cannot

afford the high overhead of asymmetric operations. However, symmetric schemes pose significant challenges in key distribution and life-cycle management, especially in dynamic swarm environments where drones often join or exit the network [26]. Establishing secure channels for initial key exchanges without a trusted infrastructure is complex, and maintaining forward and backward secrecy becomes increasingly challenging as membership fluctuates [27]. Moreover, the compromise of a single node can have dire implications if shared group keys are reused among multiple drones, allowing an adversary to decrypt all future communications [16]. This vulnerability is exacerbated in physically exposed environments where drones can be captured and reverse-engineered [25]. While hierarchical key management and periodic rekeying have been proposed to mitigate these risks, such mechanisms often incur additional communication overhead and synchronization delays, which conflict with the real-time operational requirements of swarm missions [28].

2.2.3. Behavioral or Trust Models

Behavioral or trust-oriented models provide an alternative method for authentication in UAV networks by deducing the legitimacy of nodes based on detected communication patterns and operational actions [29]. These systems commonly track metrics like message forwarding rates, flight paths, and collaboration in routing tasks to develop trust scores over time [10]. Although they are conceptually attractive for decentralized settings, these strategies encounter several challenges [30]. Firstly, they often necessitate extended observation periods to gather enough behavioral data, which can delay authentication and is not suitable for real-time swarm operations [29]. Secondly, the computational demands of trust assessment models—especially those using probabilistic or machine learning approaches—can overwhelm the limited processing capabilities of UAV systems [31]. Thirdly, these techniques are inherently susceptible to advanced adversarial tactics, such as emulating legitimate behaviors or collusion among compromised nodes to artificially boost trust scores. Such threats jeopardize the trust inference's reliability and may result in widespread compromise [6]. Moreover, the ever-changing topology and sporadic connectivity of drone swarms add to the difficulty in maintaining accurate trust metrics, as frequent node movements disrupt the continuity of behavioral observation [30]. Therefore, while behavioral models present a promising avenue for anomaly detection, their practical application in large-scale, resource-limited swarms remains challenging [32].

2.2.4. PUF Integrated Scheme

PUFs have recently emerged as a robust hardware-level security mechanism for creating unique device identifiers, which act as unchangeable identity anchors in lightweight authentication frameworks [22]. Unlike traditional cryptographic keys kept in memory, PUFs take advantage of inherent manufacturing discrepancies in integrated circuits to generate outputs that are nearly impossible to duplicate, even by the original producer [6]. This characteristic makes PUFs exceptionally resistant to cloning and physical interference, providing a solid basis for device authentication in resource-limited settings [5]. As shown in Figure 3, current uses include mutual authentication in static IoT networks, secure key provisioning for sensor devices, and establishing trust in embedded systems [33,34].

However, most existing research presumes relatively stable network architectures and centralized coordination, which do not correspond with the dynamic, external infrastructure-free characteristics of drone swarms [9]. Few investigations have looked into decentralized, swarm-scale authentication methods that utilize PUFs alongside time-dependent randomness and adaptive trust metrics [26]. Implementing mechanisms such as evolving challenge-response pairs, randomized delay scheduling, and behavior-based statistical trust evaluation could greatly improve resilience against replay, cloning, and

collusion attacks [18,26]. These gaps underscore the necessity for innovative protocols that merge PUF-based identity with distributed decision-making to realize lightweight, scalable, and tamper-resistant authentication in hostile environments.

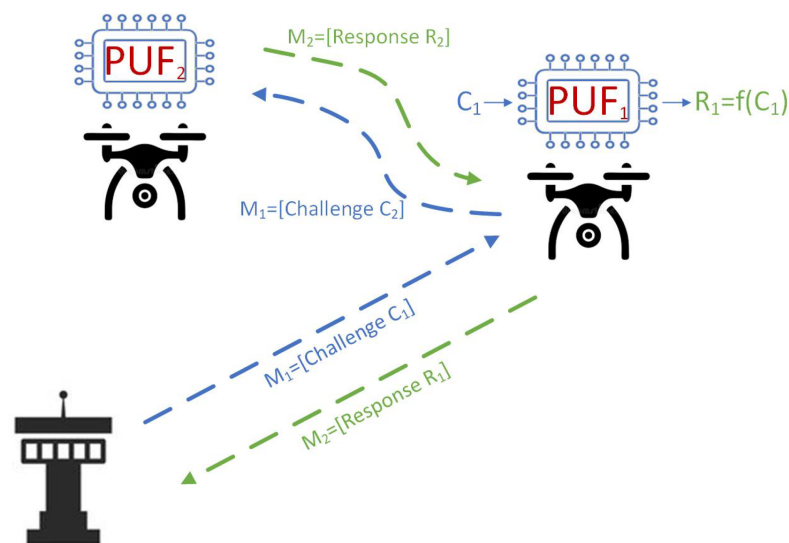


Figure 3. Mutual authentication protocol leveraging PUFs for UAV communication [19].

To better understand the strengths and limitations of current authentication approaches for drone swarms, Table 1 compares their key characteristics across critical security and implementation dimensions. The analysis considers decentralization capability, resistance to cloning and replay attacks, computational requirements, and overall suitability for swarm operations. The results highlight significant limitations in conventional approaches, particularly those dependent on centralized infrastructure or static identity mechanisms. In contrast, the proposed scheme leverages dynamic, hardware-rooted PUF randomness in a fully decentralized manner—making it inherently aligned with the operational demands of autonomous drone swarms.

2.3. Hardware Security Primitives

HSPs are physical components that utilize the unique, irreproducible traits of a device’s manufacturing process to generate secure responses or high-quality entropy [8]. These primitives take advantage of minute variations introduced during production—such as discrepancies in transistor thresholds or path delays—that are nearly impossible to replicate, even by the original manufacturer [16]. By transforming these distinct physical traits into cryptographic material, HSPs facilitate lightweight authentication and key generation without the need to store secrets in non-volatile memory, thereby diminishing vulnerability to invasive attacks [35]. Notable examples include PUFs, which generate device-specific challenge–response pairs, and TRNGs, which obtain randomness from unpredictable physical phenomena [8]. Together, these mechanisms offer a strong foundation for decentralized security in resource-limited systems such as UAV swarms, where traditional cryptographic methods are not feasible due to computational and connectivity restrictions [36].

2.3.1. Physical Unclonable Function

PUFs function by creating challenge–response pairs (CRPs) that are extremely sensitive to slight variations in the manufacturing process of semiconductors, ensuring that each response is unique and unpredictable for a particular device [6]. This inherent uniqueness establishes a solid basis for device authentication without the need for stored cryptographic keys, as illustrated in Figure 4 [19,37].

Table 1. Comparison of Authentication Methods for Drone Swarms.

Reference	Method	Decentralized Operation	Cloning Resistance	Replay Attack Resistance	Computational Overhead	Key Management	Suitability for Swarms
[17,24]	PKI-Based Auth.	✗ Requires infrastructure	✓ With strong crypto	✓	✗ High	✓ Certificates	✗ Limited due to centralization
[17,24]	Symmetric Key Auth.	✗ Group key sharing	✗ If key compromised	✓ With nonce	✓ Low	✓ Pre-distribution	✗ Risky in dynamic swarms
[28]	Trust-Based Models	✓ Inferred trust	✗ Behavior-only	✗ Easily mimicked	✗ Medium to high	✗ Not key-based	✗ Promising but fragile
[32,33]	PUF-Based Static ID	✓ One-time response	✓ Unclonable hardware	✗ Static response reusable	✓ Very low	✗ No keys	✗ Limited without renewal
This paper	PUF with Seed Evolution (This Work)	✓ Fully decentralized	✓ High (PUF + randomness)	✓ Dynamic response	✓ Low	✗ None	✓ Optimized for swarms

Note: Green checkmarks (✓) indicate favorable characteristics; red crosses (✗) indicate limitations or unsuitability.

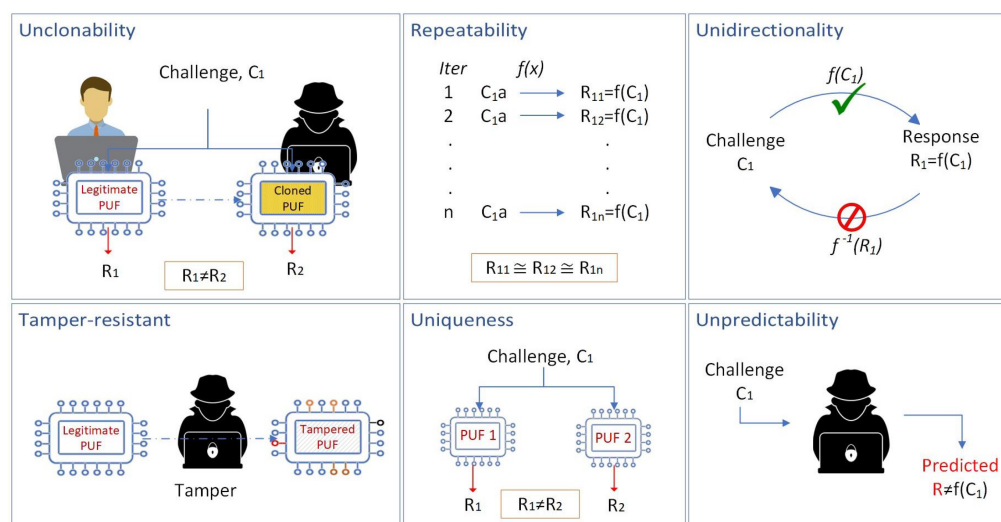


Figure 4. Fundamental properties of PUFs [19].

Figure 4 summarizes the fundamental properties of PUFs that are commonly required for robust device authentication. *Uniqueness* captures how strongly responses differ across distinct devices for the same challenge, reflecting the desired inter-device variability that enables device discrimination [37]. *Reliability* (also known as reproducibility) captures the stability of the responses of a given device's in repeated evaluations of the same challenge, despite environmental and operational variations (noise, voltage, temperature) and is typically associated with low intra-device variability [6]. *Unpredictability* indicates that an adversary should not be able to infer responses to unseen challenges—even after observing prior challenge–response pairs—thereby limiting modeling and impersonation attempts [33]. *Unclonability* denotes the practical infeasibility of reproducing the same physical behavior in another device, either through physical replication or sufficiently accurate emulation [8]. Finally, *tamper-evidence* arises because invasive probing or modification tends to perturb the underlying physical characteristics that generate the responses, causing measurable deviations and signaling compromise [36]. Together, these properties explain why PUFs can act as a lightweight hardware root of trust without storing long-term secret keys in non-volatile memory.

Additionally, PUFs are naturally tamper-evident since any attempt to physically probe or measure them invasively alters the circuit characteristics, thus modifying the response and indicating a breach [38]. Moreover, mechanisms based on PUFs are remarkably lightweight and energy-efficient, necessitating no dedicated cryptographic engine or secure key storage, making them suitable for platforms with limited resources, such as embedded systems and microcontrollers [18,37]. These characteristics position PUFs as an attractive alternative to conventional cryptographic primitives in settings where low power usage, minimal hardware demands, and resistance to physical attacks are critical [33].

2.3.2. True Random Number Generators

On the other hand, TRNGs extract entropy from fundamentally unpredictable physical phenomena like thermal noise, oscillator jitter, or metastability within electronic circuits [39]. In contrast to Pseudo-Random Number Generators (PRNGs), which depend on deterministic algorithms and can be susceptible to state compromise or prediction, TRNGs yield non-deterministic outputs, making them crucial for cryptographic functions in low-trust environments [40,41]. This unpredictability is vital for secure key generation, session initiation, and nonce creation, especially in decentralized systems where attackers might try to deduce or replay protocol states [42]. In the context of swarm drones, TRNGs can be

utilized to enhance or initiate randomness in authentication protocols, ensuring freshness and reducing the risk of replay attacks without incurring high computational costs. When paired with PUFs, TRNGs create a synergistic basis for hardware-rooted security [16,43]. PUFs provide unique identity anchors for devices, while TRNGs ensure dynamic unpredictability, facilitating lightweight authentication methods that resist cloning, tampering, and side-channel attacks [36]. This synergy is particularly beneficial in environments with limited resources where protecting memory, energy efficiency, and resilience against physical breaches are crucial for mission-critical tasks [44].

2.4. Research Gap

While the utility of HSPs for device-level identity has been well-demonstrated in IoT applications, their potential to enable fully decentralized, multi-node trust in mobile drone swarms remains largely untapped. Existing research rarely integrates statistical trust inference from swarm response patterns, dynamic seed evolution with response delay randomization, or multi-round decentralized protocols that operate without pre-shared keys or centralized oversight. This study addresses that gap by designing and evaluating a secure swarm authentication protocol grounded in HSPs, optimized for resource-constrained, external infrastructure-free, and adversarial deployment environments.

3. Research Methods and Methodology

This study is guided by a design-science research paradigm, aiming to create and evaluate a novel artifact—a decentralized authentication protocol—to address a defined class of problems in swarm security. The primary research question is: How can a drone swarm, operating in a decentralized and adversarial environment, verify the authenticity of its members in a scalable, lightweight, and tamper-resilient manner—without relying on pre-shared keys or centralized verification services?

To answer this, our methodology follows three interconnected phases:

1. **Conceptual design and protocol Formulation:** We first analyze the limitations of existing approaches (Section 2) and define the specific constraints and security goals for drone swarms (Section 5). Based on the capabilities of HSPs, we architect a novel protocol that integrates master-initiated collective token broadcasting, iterative HSP seed evolution, randomized response delays, and statistical trust evaluation. The detailed protocol specification is presented in Section 6.
2. **Mathematical modeling and security Analysis:** To establish the theoretical soundness of the design, we develop formal models of the protocol's core components: seed evolution as a Markov process, delay randomization, and the statistical anomaly detection mechanism (Section 7). This phase involves a systematic threat analysis to map potential attack vectors to the protocol's defensive properties, providing a reasoned argument for its security under a defined adversarial model.
3. **Simulation-based evaluation and benchmarking:** We implement a discrete-event simulation of the protocol in Python 3.9 to empirically validate its performance and security claims under controlled, scalable conditions (Section 8). The simulation assesses critical metrics such as uniformity, tamper detection rate, authentication latency, and scalability. The results are benchmarked against relevant schemes from the literature to contextualize the protocol's contributions and limitations.

This structured approach ensures the research is both theoretically grounded and practically relevant, moving from problem identification through artifact design to rigorous evaluation.

4. Problem Statement and Objectives

This section outlines the fundamental challenges associated with secure authentication in decentralized drone swarms and establishes the objectives that guide our proposed solution. The paper begins by defining the problem in the context of resource-constrained, external infrastructure-free environments, followed by a description of the system model and its operational constraints. Finally, we present the research objectives that shape the design of a lightweight, scalable, and tamper-resilient authentication protocol.

4.1. Problem Definition

Swarm drones deployed for infrastructure testing operate in unstructured, external infrastructure-free environments where real-time coordination, resilience, and autonomous decision-making are essential. In such distributed systems, ensuring that each drone is authentic, unaltered, and trustworthy is critical to mission safety and data integrity. However, authentication in drone swarms presents unique challenges: there is no centralized authority to manage credentials, validate nodes, or distribute session keys; drones are resource-constrained, with strict limits on processing power, memory, and energy; the physical exposure of drones makes them susceptible to cloning, tampering, and device impersonation; and traditional cryptographic mechanisms are computationally heavy and assume persistent connectivity to secure infrastructure—both of which are unrealistic in dynamic swarm deployments. Thus, the core problem can be formulated as follows: How can a drone swarm, operating in a decentralized and adversarial environment, verify the authenticity of its members in a scalable, lightweight, and tamper-resilient manner—without relying on pre-shared keys or centralized verification services?

4.2. System Model and Constraints

We consider a swarm $S = \{D_1, D_2, \dots, D_N\}$ consisting of N autonomous drones. Each drone is equipped with an HSP, such as a PUF or a TRNG, and operates with limited computational resources typical of embedded microcontrollers. Drones can exchange short messages within a local communication radius and participate in a multi-round authentication process initiated by a designated rotating coordinator (master drone), which may be rotational or elected for each session. Importantly, the system assumes no pre-installed symmetric keys or public certificates. Adversaries are assumed to possess the capability to physically clone or tamper with captured drones and may attempt impersonation, replay, or injection attacks to disrupt the authentication process. However, they are considered unable to reverse-engineer or replicate the unique internal characteristics of genuine HSPs.

As illustrated in Figure 5, the rotating coordinator initiates each authentication round by broadcasting a fresh public token, and each responder uses its HSP-rooted internal state to update its seed, compute a bounded delay, and transmit a round-bound response. The master validates received responses and aggregates multi-round observations (winner-frequency behavior) to support anomaly detection and identify compromised or cloned drones.

4.3. Research Objectives

To address the above problem, this paper aims to design and evaluate a fully decentralized authentication scheme that can be deployed in real-time swarm missions. The specific objectives are:

- Develop a lightweight authentication protocol that leverages HSP-based randomness and multi-round challenge–response interactions to establish mutual trust within the swarm.

- Introduce a secure method for evolving cryptographic seeds over time to produce non-repeatable, non-replayable responses without relying on synchronized clocks or external randomness sources.
- Integrate a statistical monitoring mechanism to evaluate the legitimacy of drones over repeated interactions, allowing detection of cloned or tampered nodes through abnormal response patterns.
- Ensure that the protocol scales to tens of drones and operates within microsecond-level computational bounds, making it suitable for real-time embedded environments.

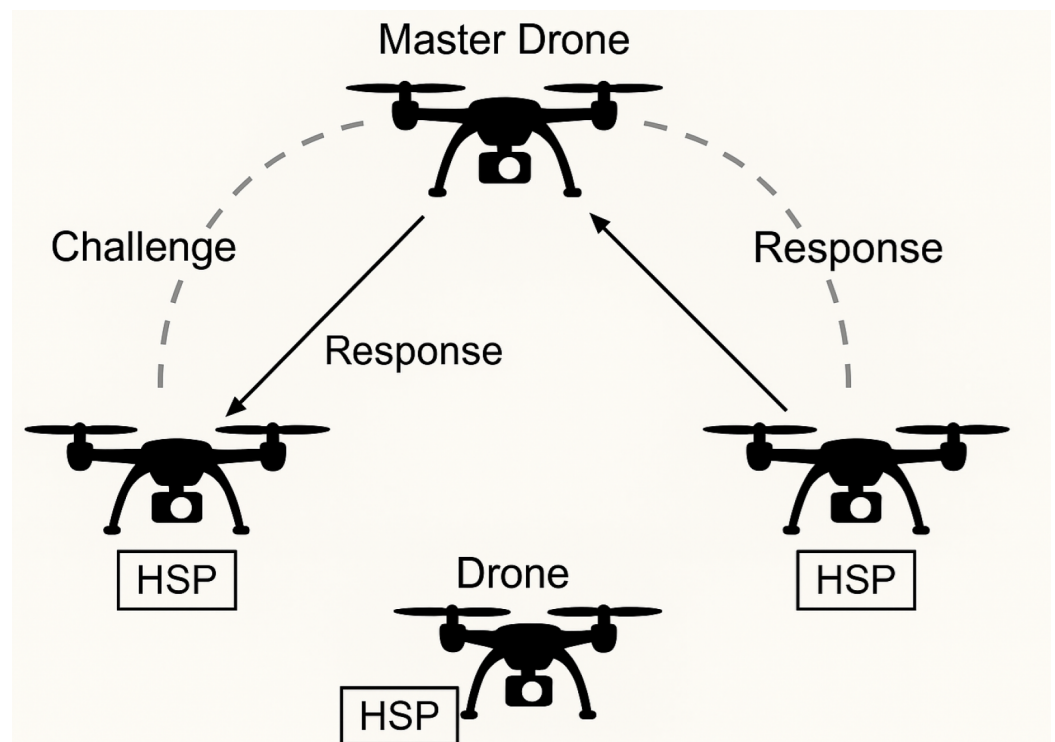


Figure 5. Challenge–response authentication between master and slave drones using HSP.

4.4. Security Goals

The proposed scheme is designed to satisfy several critical security properties. Authenticity ensures that only legitimate drones equipped with valid HSPs can generate correct and verifiable responses as shown in Figure 5.

Cloning resistance prevents duplicated devices from emulating the correct challenge–response behavior or dynamic HSP seed evolution. Replay protection is achieved through evolving seeds and time-varying responses, which eliminate the possibility of reusing old messages. Additionally, the protocol exhibits tamper-evident behavior, as any modification to a drone’s internal state or HSP alters its statistical response pattern, flagging it as anomalous. Finally, decentralization is maintained by eliminating reliance on a central authority or pre-shared secrets, instead building trust collaboratively through multi-round protocol interactions.

5. Authentication Protocol

This section describes the proposed decentralized authentication scheme based on HSPs, designed for secure identity verification and anomaly detection in swarm drone networks. The protocol is optimized for dynamic, resource-constrained environments where central key management is infeasible. It leverages a multi-round, master-initiated

challenge–response process, incorporating HSP seed evolution, randomized response timing, and statistical trust analysis.

5.1. System Model and Notation

Consider a swarm of N drones $\{D_1, D_2, \dots, D_N\}$. Time is divided into authentication rounds indexed by $k \in \mathbb{N}$. In each round, one drone D_m acts as master and all others act as responders. Every drone D_i maintains an internal state per-device $S_i(k)$ seeded by its HSP (PUF or TRNG-derived initialization). We assume an unreliable broadcast channel with bounded propagation and queueing delays relative to the protocol's timing resolution, and a lossy medium that may drop or collide packets; the protocol is robust to such loss by design. Let $H(\cdot)$ denote a cryptographic hash function modeled as a one-way, collision-resistant primitive. The operator $\oplus$ is bitwise XOR, and $\parallel$ denotes concatenation. Unless otherwise stated, all computations are performed over bit-strings with standard conversions to integers where needed.

5.2. Round Structure

Each authentication round comprises four conceptual stages: token emission, HSP seed evolution, delay-based scheduling, and winner selection with statistical evaluation.

5.2.1. Token Emission

The master D_m samples a public, round-specific token T_k from a high-entropy source (HSP-backed TRNG or equivalent) and broadcasts it to the swarm. The token is observable and non-secret; its sole role is to inject fresh entropy per-round derived MAC key and to advance device states.

5.2.2. Seed Evolution

Upon receipt of T_k , each drone updates its hidden internal state as

$$S_i(k+1) = H(S_i(k) \oplus T_k). \quad (1)$$

where $S_i(k)$ is the seed of drone D_i at round k , T_k is the challenge token, $H(\cdot)$ is a hash function, and $\oplus$ denotes XOR.

This evolution ensures forward progression even without clock synchronization and prevents replay of prior states, since recovering $S_i(k)$ or $S_i(k+1)$ from observations of T_k and protocol messages is computationally infeasible under the hash assumption.

5.2.3. Delay Computation and Response Scheduling

Each drone derives a randomized—but device-consistent—transmission delay from its updated state. Let $\text{map}(\cdot)$ be a deterministic extractor mapping the bit-string $S_i(k+1)$ to an integer in $\{0, 1, \dots, b-1\}$. The delay is computed as

$$d_i(k) = a \cdot \sin(S_i(k) \bmod b) + c. \quad (2)$$

where a is the amplitude, b is the modulus, c is the offset, and $S_i(k)$ is the seed of drone D_i at round k .

We emphasize that the specific choice of $\sin(\cdot)$ in Equation (2) is *not* security-critical; it is only one instantiation of a deterministic, bounded delay-shaping function. What the protocol requires is a device-consistent pseudorandom mapping from the secret evolving state to a bounded delay domain with good dispersion (to reduce systematic clustering

and collisions). In particular, a practical embedded implementation can replace the sine map with an integer-only construction such as

$$d_i(k) = c + (\text{PRF}(S_i(k+1)) \bmod b), \quad (3)$$

(or equivalently using a keyed hash/KDF seeded by $S_i(k+1)$), which guarantees cross-MCU reproducibility and avoids floating-point overhead. Optionally, to further reduce exact-delay collisions under dense swarms, we apply *jitter shaping* by deriving an additional small offset from independent PRF bits (e.g., intra-slot jitter) or using rejection/permutation shaping over $\{0, \dots, b-1\}$ while preserving the same bounded range.

The use of a bounded trigonometric function (sine) was selected to generate a smooth, quasi-uniform distribution of delays while minimizing the probability of two drones computing an identical delay value—a risk higher with simple linear modulo operations. Although the formulation uses floating-point operations, in a practical embedded implementation, this can be efficiently approximated using fixed-point integer arithmetic and a pre-computed lookup table for the sine function, avoiding the computational overhead of floating-point units on low-power microcontrollers. The core security requirement is that the function $f(\cdot)$ be deterministic for the genuine device and produce outputs that are unpredictable and uncorrelated for an adversary.

With design parameter $a > 0$, $b \in \mathbb{N}$, and c chosen to ensure bounded, non-negative delays, the argument of the $\sin(\cdot)$ function is scaled to lie within $[-1, 1]$, guaranteeing smooth variation and preventing overflow. The use of a nonlinear function (sine function) introduces unpredictable response times, making timing analysis attacks significantly harder. This design ensures that the delay remains deterministic for the genuine device while appearing unpredictable to an adversary lacking the internal state. After waiting for $d_i(k)$, the drone transmits its response; the delay value itself is never disclosed, adding an extra layer of security.

5.2.4. Response Generation and Broadcast

After waiting $d_i(k)$, drone D_i broadcasts a response

$$R_i(k) = H(S_i(k+1) \parallel k). \quad (4)$$

where $R_i(k)$ is the response of drone D_i at round k , $S_i(k+1)$ is the updated seed, k is the round index, $H(\cdot)$ is a hash function, and $\parallel$ indicate concatenation.

Including the round index k binds responses to their round and precludes trivial reordering or replay across rounds. Since $S_i(k+1)$ is hidden and device-specific, valid responses are infeasible to forge or predict without the genuine hardware state.

5.2.5. Winner Selection and Statistical Trust Evaluation

The master records the identity of the first valid response and designates that drone as the round winner:

$$w_k = \arg \min_i d_i(k). \quad (5)$$

where w_k represents the winner of round k , $\arg \min_i$ selects the index i that minimizes the value, $d_i(k)$ is the delay of drone D_i at round k , and N is the total number of drones.

More generally, winner identities within a sliding window of W rounds can be modeled as a multinomial distribution with baseline win probabilities $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_N)$ to account for persistent benign biases (e.g., hardware/radio asymmetries). Let O_i denote the observed

number of wins of drone i in the window. The expected count is then $E_i = W \hat{p}_i$ (instead of the uniform special case $E_i = W/N$), and the Pearson chi-square statistic becomes

$$\chi^2 = \sum_{i=1}^N \frac{(O_i - E_i)^2}{E_i}. \quad (6)$$

In this work, we validate fairness under ideal symmetric conditions where $\hat{p}_i = 1/N$; we propose learning $\hat{\mathbf{p}}$ during a trusted commissioning phase (or tracking it online using EWMA) as a practical enhancement for real deployments.

Under benign conditions and in the absence of manipulation, the winner identities $\{w_k\}$ should be approximately uniform over $\{1, \dots, N\}$, up to channel loss and natural variance.

The choice of G trades off liveness and throughput: if G is too small, valid but distant (or temporarily delayed) responses may be missed, and rounds may be incorrectly marked inconclusive; if G is too large, round duration increases and throughput decreases. A practical static configuration can be obtained by bounding each term in Equation (7) from deployment geometry and platform specifications. In particular, $d_{\max}$ follows directly from the delay function parameters (as an upper bound on $d_i(k)$), Δ_{prop} can be bounded from the swarm diameter as $\Delta_{\text{prop}} \approx R_{\max}/c$ where $R_{\max}$ is the maximum inter-drone distance to the master and c is the speed of light, and Δ_{proc} can be set from measured worst-case processing/MAC jitter on the flight controller and radio stack. The safety term Δ_{margin} may be chosen as a percentage of the deterministic components; e.g., $\Delta_{\text{margin}} = \rho (d_{\max} + \Delta_{\text{prop}} + \Delta_{\text{proc}})$ with $\rho \in [0.1, 0.3]$ (20% as a typical initial value).

To avoid overly conservative static settings and improve throughput under changing conditions, G can also be configured dynamically. The master can maintain a sliding window of the most recent valid round completion times (or earliest-response arrival times) and update the guard interval using a high-percentile estimate:

$$G_{t+1} = \max(G_{\min}, Q_p(\{\tau\}_{t-W+1:t}) + \Delta_{\text{margin}}), \quad (7)$$

where τ denotes the observed time between token broadcast and the first valid response in a round, $Q_p(\cdot)$ is the p -th percentile (e.g., $p = 0.95$ or 0.99) over a window of W rounds, and $G_{\min}$ is a lower bound ensuring at least $d_{\max} + \Delta_{\text{prop}} + \Delta_{\text{proc}}$ is covered. This adaptation keeps G tight during stable periods (improving throughput) while expanding it when the observed response spread increases due to mobility, interference, or transient MAC delays, thereby reducing false inconclusive rounds.

To detect manipulation that biases winner selection, the master applies the Pearson chi-square test defined in Equation (6) over a sliding window of W rounds, using expected counts $E_i = W \hat{p}_i$. Under benign symmetric conditions, this reduces to the uniform special case $\hat{p}_i = 1/N$ (i.e., $E_i = W/N$).

A test at significance level α triggers an alert if χ^2 exceeds the $(1 - \alpha)$ -quantile of the χ^2_{N-1} distribution. Windows may be advanced with overlap to balance sensitivity and timeliness. In lossy channels, the expected counts can be normalized by the number of rounds with at least one valid reception, or stratified by link-quality estimates, to reduce false positives due to exogenous interference.

5.2.6. Master Role, Decentralization, and Robustness

To avoid single points of failure and reduce predictability, the master drone serves as a *rotating coordinator* and the master role rotates across rounds using either a deterministic schedule or a token-derived election. Two common strategies are:

$$\text{Round-robin: } m = k \bmod N, \quad (8)$$

where m is the master index for round k , k is the round number, and N is the total number of drones.

$$\text{Token-based rotation: } m = \text{map}(H(T_k)) \bmod N, \quad (9)$$

where m is the master index for round k , $\text{map}(\cdot)$ maps the hash output to an index space, $H(\cdot)$ is a hash function, T_k is the challenge token, and N is the total number of drones.

Rotation ensures that no single node controls challenge issuance for extended periods, thereby mitigating targeted jamming or the capture of a persistent coordinator.

The protocol tolerates packet loss and collisions by allowing rounds to be marked inconclusive if no valid response is received within a guard interval; such rounds are excluded from statistical tallies or down-weighted. Designers should select $d_{\min}$, and the guard interval to dominate propagation and MAC-layer contention times, ensuring that winner selection reflects the intended randomized scheduling rather than channel artifacts. Because state evolution in Equation (1) depends only on public tokens and local secrets, the protocol does not require global time synchronization, long-term keys, or secure key distribution.

5.2.7. State Recovery and Synchronization

The HSP seed evolution in Equation (1) requires all drones to process the same public token sequence $\{T_k\}$. In lossy wireless channels, missed tokens can desynchronise a drone's evolving state $S_i(k)$; we therefore add a lightweight recovery mechanism that does not assume PKI, pre-shared keys, or external infrastructure.

Public Token Chaining

We define tokens as a publicly verifiable hash chain; e.g.,

$$T_k = H(T_{k-1} \parallel R_{w,k-1} \parallel \text{salt}), \quad (10)$$

where $R_{w,k-1}$ is the first valid response observed in round $k-1$ and salt is a domain-separation constant. A token is accepted only if it satisfies Equation (10) given the last validated token.

A drone D_j detects a gap when token indices are non-consecutive (e.g., it has T_{k-1} but receives T_{k+1}) or when the chain check fails. It broadcasts a Catch-up Query with its last validated index/token; e.g., $(j, k-1, T_{k-1})$. Any synchronized neighbor may respond with the missing chain elements (e.g., $(T_k, R_{w,k-1}), \dots$). The requester verifies recovery data locally by recomputing hash-chain continuity from T_{k-1} and discarding any element that does not match.

After validating missing tokens, D_j recomputes its state by iterating the normal update rule over the missed rounds. For example, if $S_j(r) = H(S_j(r-1) \oplus T_r)$, then $S_j(k) \leftarrow H(S_j(k-1) \oplus T_k)$, and so on until the current round. This restores liveness after transient loss without trusted reset channels; neighbors only forward public chain elements, while correctness is enforced by deterministic chain verification.

5.2.8. Security Implications of Master Compromise

Although the master role rotates, a compromised master may broadcast a malicious round token T'_k , which can temporarily desynchronize nodes that receive different token values in round k . In our protocol, tokens are publicly verifiable as a hash-chained sequence; thus, before updating its state, each drone checks chain continuity and rejects any token that does not satisfy the token update rule (e.g., Equation (10)). Consequently, a compromised master cannot silently desynchronize honest drones with an arbitrary T'_k ; at worst it can cause a failed/inconclusive round (liveness disruption) by transmitting an invalid or

inconsistently received token. In our protocol, however, state progression is not solely determined by the master: after token broadcast, drones compute delays and compete to respond, and the first valid responder (the round winner) broadcasts its response, which is adopted as the next-round token. Therefore, the worst-case impact of a single adversarial master is primarily a one-round liveness disruption: honest drones that evolve their state with inconsistent tokens will generate inconsistent responses, leading to an inconclusive round (no verifiable winner) or failed verification of the master-declared outcome. This disruption is detectable because the response set and winner selection become inconsistent with the expected validation behavior and statistics.

Crucially, because the next token is produced via competition by the round winner rather than being permanently dictated by the master, a compromised master cannot create irreversible long-term divergence with a single malicious broadcast. Under normal conditions, the subsequent token T_{k+1} is emitted by the winner (typically honest) and re-establishes a common public reference for continued evolution. Moreover, due to the one-way nature of HSP seed evolution (Equation (1)), the adversary cannot roll back the swarm state or deduce past internal states from observed tokens and responses. As a result, any divergence caused by T'_k is bounded in duration and does not compromise historical secrecy or allow reconstruction of prior seeds.

Regarding denial-of-service, a malicious master can at most waste the duration of one round (bounded by the guard interval G) by broadcasting a disruptive token that increases the likelihood of an inconclusive round. Sustained long-term DoS would require repeated compromise of successive masters or persistent channel jamming. Both are mitigated by master rotation and by excluding (or down-weighting) inconclusive rounds from the chi-square winner-frequency statistics, ensuring that transient liveness failures do not translate into false compromise decisions. Where partial reception causes some drones to miss the winner-emitted token, the catch-up recovery mechanism enables resynchronization from neighboring honest drones without requiring any external trusted infrastructure or global reset.

5.3. Security Intuition and Complexity

Security stems from three complementary properties: (i) unpredictability of the hidden, HSP-seeded state $S_i(k)$, which underpins response authenticity in Equation (4) and delay determinism in Equation (2) for genuine devices; (ii) public tokens that advance state without revealing it, preventing replay and desynchronization attacks; and (iii) statistical anomaly detection over multiple rounds that exposes cloning or bias attempts manifesting as winner distribution skew.

An adaptive adversary who eavesdrops tokens cannot reconstruct $S_i(k)$ or precompute valid $R_i(k)$ without the device's HSP; attempts to bias the ordering through deterministic injections or selective jamming yield detectable deviations in the winner statistics. The protocol is lightweight: per-round derived MAC key, each drone performs a constant number of hash evaluations and scalar operations, waits a bounded delay, and transmits a single short response. Memory overhead is $\mathcal{O}(1)$ per device (current state and counters) as shown in Algorithm 1.

Communication consists of one broadcast token and at most one response per drone per round derived MAC key, supporting scalability to large swarms under typical MAC constraints.

Algorithm 1 Drone HSP Legitimacy Testing**Require:** N = Total drones, $S_i(0)$ = Initial seeds, K = Iterations**Ensure:** Authentication Status (Legitimate/Compromised)

```

1: procedure MASTER-INITIATED AUTHENTICATION
2:   Master Drone (MD) generates  $T_0 = \text{HSP\_Generate}()$ 
3:   MD broadcasts  $T_0$  to all drones
4:   for  $k = 1$  to  $K$  do
5:     for each drone  $D_i$  do
6:       Update seed:  $S_i(k) = \text{Hash}(S_i(k-1) \oplus T_{k-1})$ 
7:       Compute random delay:  $d_i(k) = \text{RandomDelayFunction}(S_i(k))$ 
8:       Compute random response:  $R_i(k) = \text{HSP\_Generate}(S_i(k))$ 
9:     end for
10:    First drone to finish sends  $R_i(k)$  as new token  $T_k$ 
11:    All drones receive  $T_k$  and update seeds
12:  end for

13: Master observes patterns:
14: if patterns match expected statistics then
15:   return "LEGITIMATE"
16: else
17:   return "COMPROMISED"
18: end if
19: end procedure

```

6. Mathematical Model and Security Analysis

This section formalizes the mathematical foundations of the authentication protocol described in Section 5 and examines its security properties. Although Section 5 details the operational flow, here we interpret and model the behavior of its key components HSP seed evolution, randomized delay, response computation, and detection of statistical anomalies detection—under realistic assumptions.

6.1. Seed Evolution Dynamics

As defined in Equation (1), each drone D_i updates its internal cryptographic seed $S_i(k)$ using a combination of its prior state and the challenge token T_k . This recursive formulation injects entropy and ensures state progression across rounds without requiring global synchronization. The HSP seed sequence $\{S_i(k)\}_{k=0}^K$ can be modeled as a Markov process:

$$\Pr(S_i(k+1) \mid S_i(0:k), T_{0:k}) = \Pr(S_i(k+1) \mid S_i(k), T_k). \quad (11)$$

$S_i(k)$ stands for the seed of drone D_i at round k , T_k is the challenge token, and $\Pr(\cdot)$ indicates probability, where each new state depends solely on the immediate past value and the current token. This design ensures non-repeatability across rounds due to token entropy, unlinkability of responses across sessions, and tamper sensitivity, since any alteration in internal state affects all subsequent outputs.

The hash function $H(\cdot)$ is assumed to behave as a random oracle, preserving unpredictability under adversarial observation.

In our simulations, we model hashing using SHA-256 truncated to 64 bits to reflect a lightweight implementation and to improve simulation efficiency. This choice provides a conservative, lower-bound analysis: truncation reduces the output space, thereby making brute-force preimage or collision attempts strictly easier than they would be with wider outputs. The security of the HSP seed evolution nevertheless rests on the combined entropy of the input, since the truncated output is derived from a secret internal seed

$S_i(k)$ whose entropy originates from the high-entropy HSP initialization. For a resource-constrained adversary attempting real-time brute-force recovery or forgery within the protocol time scale (milliseconds per round), exhaustive search over 2^{64} candidates remains computationally infeasible in practice for a single round. For production systems, we recommend using a wider output (e.g., 128-bit or the full 256-bit SHA-256 output) to provide a larger security margin against offline, well-resourced attackers.

6.2. Delay Randomization Behavior

The randomized delay function in Equation (2) maps the internal state to a bounded real-valued delay $d_i(k)$. Although deterministic for a given drone (due to fixed HSP properties and constants a, b, c), its output is opaque to adversaries, preventing prediction or synchronization. We interpret $d_i(k)$ as a pseudo-random variable drawn from a quasi-sinusoidal distribution, ensuring approximately uniform delay distribution across legitimate drones over many rounds, low probability of collisions under typical system parameters, and detectable anomalies when malicious nodes attempt to bias or shortcut delays.

6.3. Response Computation Semantics

Equation (4) binds each drone's response $R_i(k)$ to its updated internal state and the current round index. From an information-theoretic perspective, this produces a non-replayable response stream and a unique fingerprint per device per round derived MAC key. Legitimate responses are indistinguishable from uniformly random strings to external observers because of the properties of the hash function. This cryptographic binding is critical for preventing impersonation and pre-computation attacks.

6.4. Statistical Winner Modeling and Anomaly Detection

As introduced in Equation (5), the drone with the minimum delay in a given round is selected as the winner w_k . Over a sliding window of W rounds, the empirical win frequency of each drone D_i is:

$$f_i(t) = \frac{1}{W} \sum_{k=t-W+1}^t \mathbf{1}(w_k = i), \quad (12)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Expected Multinomial Baseline

Rather than assuming perfect uniformity, we model winner identities in a window as an expected multinomial distribution with baseline win probabilities $\hat{\mathbf{p}}(t) = (\hat{p}_1(t), \dots, \hat{p}_N(t))$ that capture persistent benign asymmetries (e.g., hardware/radio differences). Let

$$O_i(t) = \sum_{k=t-W+1}^t \mathbf{1}(w_k = i) \quad (13)$$

denote the observed win count of drone i in the window, and define the expected count as

$$E_i(t) = W \hat{p}_i(t). \quad (14)$$

We then compute the Pearson chi-square statistic

$$\chi^2(t) = \sum_{i=1}^N \frac{(O_i(t) - E_i(t))^2}{E_i(t)}. \quad (15)$$

Uniformity is a special case when $\hat{p}_i(t) = 1/N$ for all i .

We propose estimating $\hat{\mathbf{p}}$ during a trusted commissioning phase and using it as the null model for monitoring. Alternatively, $\hat{\mathbf{p}}(t)$ can be tracked online using EWMA to accommodate slow benign drift, in which case $\chi^2(t)$ is used as a monitoring statistic with a calibrated alarm threshold.

An anomaly is flagged when $\chi^2(t)$ exceeds the threshold for $N - 1$ degrees of freedom at the chosen significance level (e.g., 95%), which may indicate cloning or systematic delay manipulation. This mechanism is lightweight and does not require PKI or centralized oversight.

6.5. Security Considerations and Attack Surface Analysis

Tokens are broadcast publicly, allowing adversaries to monitor them freely. However, this does not compromise security: the root of trust lies in each drone's private HSP state $S_i(k)$, which is unclonable and hidden. The update rule as per Equation (1) ensures that even full knowledge of all tokens does not enable prediction of future seeds, delays, or responses. Tokens serve as public randomness, while hidden HSP states provide the security anchor.

Potential side-channel vectors merit consideration. Monitoring tokens alone is harmless, but adversaries may attempt timing correlation or token injection. Replay attacks fail because seeds evolve each round. Token injection could disrupt synchronization, but this risk is mitigated by strict acceptance of tokens from the designated master and by master rotation. Physical side-channels such as power analysis and EM probing remain a threat if drones are captured; these can be mitigated through standard hardening measures such as shielding, noise injection, and obfuscation. Table 2 summarizes these threats and the corresponding mitigations.

Table 2. Mapping attack vectors to threats and mitigation techniques.

Attack Vector	Threat Description	Mitigation Strategy
Replay attack	Reuse of old tokens or responses to impersonate drones	Seeds evolve each round; old values are invalid and rejected
Token injection	Fake tokens desynchronize seeds	Tokens accepted only from elected master; rotation prevents capture
Cloning/impersonation	Captured drones physically replicated	HSP unclonability; anomalies detected via statistical trust analysis
Sybil attack	Forged multiple identities	Each identity requires a genuine HSP instance; fakes rejected
Wireless jamming/collision	Interference delays or drops responses	Inconclusive rounds retried; system tolerant to losses
Physical side-channels	Power/EM leakage or probing of HSP	Mitigated by shielding, noise injection, and obfuscation

Robustness to noisy PUFs and environmental variance: The protocol assumes a stable initial seed $S_i(0)$. In practice, raw PUF responses exhibit noise due to temperature, voltage, and aging. During provisioning, we therefore stabilize PUF reliability (e.g., for Ring Oscillator or Arbiter PUFs) using lightweight error-reduction such as odd-sample majority voting, and then apply established Helper Data Algorithms (HDAs)/Fuzzy Extractors [5,33] to derive a stable, reproducible cryptographic key from noisy measurements. This key initializes $S_i(0)$. We target an intra-device bit error rate (BER) below $X\%$ such that the seed can be regenerated with failure probability below 10^{-6} under the chosen voting/ECC parameters. Subsequent state evolution via deterministic hashing is immune to raw PUF noise, ensuring that intra-device variance does not cause false positives in the statistical trust evaluation.

Impact of physical propagation delays: The assumption of negligible wireless latency (Section 6.6) holds for tightly clustered swarms. Quantitatively, with $c \approx 3 \times 10^8$ m/s, a 1 km path corresponds to ≈ 3.3 μ s of time-of-flight (ToF), whereas our scheduled delays and guard intervals are on the order of milliseconds (e.g., ≈ 5 –15 ms); thus, for ≤ 1 km swarm diameters ToF contributes only ~ 0.02 – 0.06% of the delay spread and does not materially bias response ordering. In wide-area deployments, varying signal ToF can bias the observed response order. To mitigate this, the master can perform delay normalization. By estimating the relative distance to each drone (via RSSI or known positions from collaborative localization) and subtracting the estimated ToF from the observed response time, the master can compute a more accurate protocol-intended delay. The guard interval G must be configured to accommodate the maximum ToF differential within the swarm's operational radius.

Resistance to PUF modeling attacks: Strong PUFs, if their CRPs are exposed, are vulnerable to machine learning modeling attacks. Our protocol does not expose raw CRP data. The PUF output is used only once to seed the initial state $S_i(0)$ or is intermittently refreshed in a controlled manner. All observable values (responses $R_i(k)$) are the output of a cryptographic hash function applied to the evolved internal state $S_i(k+1)$. This non-linear transformation, combined with the secrecy of the internal state, prevents an adversary from collecting the direct input-output pairs needed to train an effective model of the underlying PUF.

6.6. Summary of Modeling Assumptions

The following assumptions underpin the above formulations: Hash functions behave as ideal pseudorandom functions; HSPs (PUFs/TRNGs) are tamper-evident and unique per device; Challenge tokens T_k cannot be predicted or influenced by adversaries; Wireless latency is negligible compared to randomized delays or uniformly distributed, and delay collisions are resolved deterministically or via retry, and occurrences are logged. These assumptions align with the operational constraints of low-power embedded flight controllers in swarm scenarios.

The mathematical formulations and security analysis presented in Section 6 establish the theoretical soundness of the proposed protocol, demonstrating its resilience against replay, cloning, and timing-based attacks under the stated assumptions. To validate these properties in practice and assess operational feasibility, we now turn to empirical evaluation. Section 7 details the simulation setup, performance metrics, and experimental results, focusing on scalability, fairness, anomaly detection accuracy, and timing overhead under both benign and adversarial conditions.

7. Simulation Setup and Results

We evaluate the protocol across a range of conditions to assess its scalability, statistical uniformity, tamper detection capability, and timing overhead. All experiments were designed to reflect realistic constraints typical of embedded drone systems equipped with hardware security primitives.

7.1. Simulation Environment and Parameters

The simulation environment was developed in Python 3.9, with NumPy-based numerical evaluation and Matplotlib 3.5.0 for visualization. Hashing operations were simulated using SHA-256, truncated to 64 bits for resource-equivalent modeling of lightweight hash cores. Each drone's HSP module was emulated using a fixed seed random function (for deterministic uniqueness) combined with round-specific entropy to reflect PUF/TRNG

characteristics. The following base configuration was used unless otherwise stated in Table 3:

Table 3. Simulation parameters and values.

Parameter	Value
Number of drones N_N	5–50
Authentication rounds K_K	100–1000
Hash output width	64 bits
Delay constants a, b, c	$a = 10, b = 1000, c = 5$ ms
Noise model	Optional Gaussian jitter on delay ± 1 ms
Adversary model	Static clone or delay-biasing attacker
Evaluation trials	50 independent simulations per scenario

The simulator adopts a simplified wireless channel abstraction with optional timing jitter and loss. To incorporate contention effects without implementing a full MAC protocol, we include a discrete-slot collision model: each drone schedules its transmission at time $t_i(k)$ (derived from the protocol delay $d_i(k)$ plus optional jitter) and we quantize time into slots of width Δ . The slot index is

$$s_i(k) = \left\lfloor \frac{t_i(k)}{\Delta} \right\rfloor \quad (16)$$

where Δ is the slot width.

If two or more drones transmit in the same slot in round k ($|\{i : s_i(k) = s\}| > 1$), a collision occurs and no response from that slot is received. Colliding drones retry in the next round using a fresh delay draw (equivalently, a randomized backoff at the round level). This lightweight abstraction captures the dominant impact of collisions on throughput and latency as N grows, while remaining implementation-light; it does not model carrier sensing, capture effects, or CSMA/CA.

As future work, we will extend the simulator to an explicit slotted ALOHA MAC model with per-slot retransmission behavior and analytically grounded collision dynamics, enabling more detailed quantification of MAC-layer effects and non-linear contention overhead, as well as validation via hardware-in-the-loop experiments.

7.2. Performance Metrics

To assess the effectiveness of the protocol, four metrics were considered: *Uniformity Index (U)*, which applies a Chi-square statistic over winner frequencies to test statistical fairness in delay-based response order; *Tamper Detection Rate (TDR)*, representing the probability of detecting a cloned or manipulated drone across rounds; *Authentication Latency (L)*, the average time required to complete one authentication round, dominated by the delay response spread; and *Scalability Factor (S)*, which measures the relative growth in computational steps and authentication latency as the swarm size increases.

7.3. Baseline Performance (Benign Conditions)

Under benign symmetric conditions, the winners' identities were consistent with the uniform special case of the expected multinomial baseline, validating fairness and unpredictability as shown in Figure 6.

Figure 6 reports the average winning frequency of each drone under benign conditions, where a "win" denotes the drone whose response is received first in a round (i.e., the minimum-delay responder). For each run, we compute $f_i = \frac{1}{K} \sum_{k=1}^K \mathbf{1}(w_k = i)$ and then average f_i across independent trials to obtain the values shown.

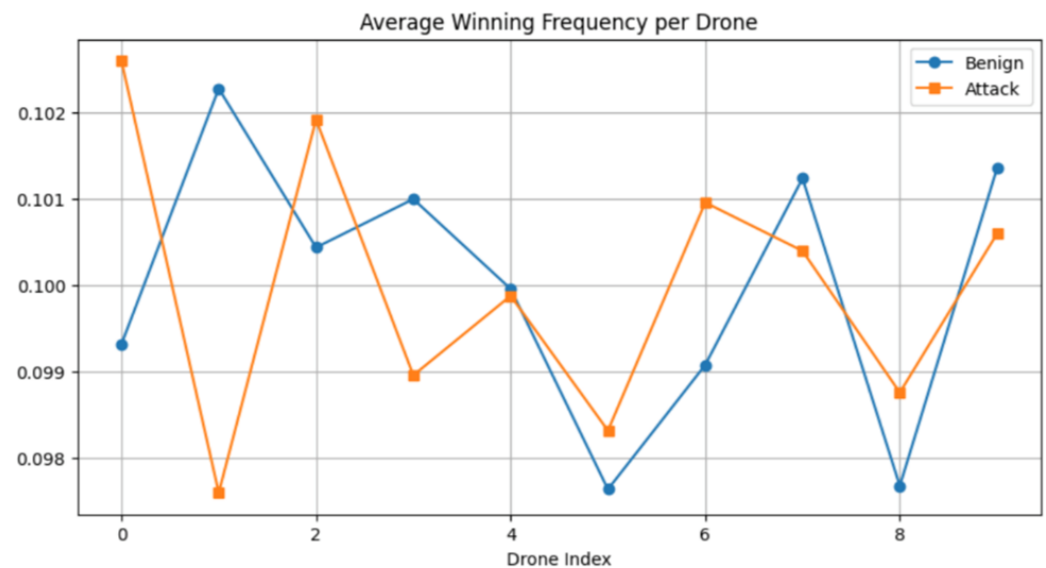


Figure 6. Average winning frequency.

Under benign conditions, the protocol demonstrated statistical fairness. For a swarm of $N = 10$ drones over $K = 1000$ rounds, the expected winning frequency for each drone is $1/N = 0.1$. Figure 6 plots the observed average frequency per drone from the simulation. The results show a mean frequency of 0.1002 with a standard deviation of 8.7×10^{-4} . Using observed win counts $O_i = Kf_i$ and expected counts $E_i = K/N$, a chi-square goodness-of-fit test failed to reject the null hypothesis under the uniform baseline ($\chi^2 = 7.82$, $p = 0.55 > 0.05$), indicating that the winner selection process is consistent with the uniform baseline under symmetric (ideal) conditions. The per-round latency remained in the millisecond range (8–15 ms), supporting feasibility for real-time swarm coordination tasks.

7.4. Adversarial Scenarios and Detection Performance

We consider two attack classes: (i) token injection, where an adversary injects counterfeit tokens to desynchronize seeds; and (ii) delay biasing, where a compromised node manipulates its delay to win disproportionately often. For token injection, we evaluate two acceptance modes: filtered (drones accept tokens only from the elected master) and unfiltered (drones accept any token).

Table 4 compares filtered and unfiltered token acceptance under injection stress. Filtering neutralizes the attack: all 800.000 rounds remain valid, with zero inconclusive cases, and detection rates remain near 0–0.125, consistent with statistical fluctuations at the 5% test level. Unfiltered acceptance leads to desynchronization, with valid rounds dropping to ~ 722.800 (rate 0.1) and ~ 556.600 (rate 0.3), and inconclusive rounds rising to ~ 77.300 and ~ 243.400 , respectively. Detection remains low (~ 0.125), indicating that the chi-square test does not misfire on liveness failures but focuses on manipulation-induced skew.

Table 4. Injection Scenarios (Filtered vs. Unfiltered)—Summary.

Flavour	Trials	Mean Valid Rounds	Mean Inconclusive	Detection Rate
inj_0.1_filtered	8	800	0	0
inj_0.1_unfiltered	8	722.75	77.25	0.125
inj_0.3_filtered	8	800	0	0.125
inj_0.3_unfiltered	8	556.625	243.375	0.125

Detection power versus observation window size K for a single compromised node. The power is negligible for $K \leq 100$, crosses a reliable threshold near $K \approx 300$, and saturates near 1.0 by $K \approx 500$ as shown in Figure 7.

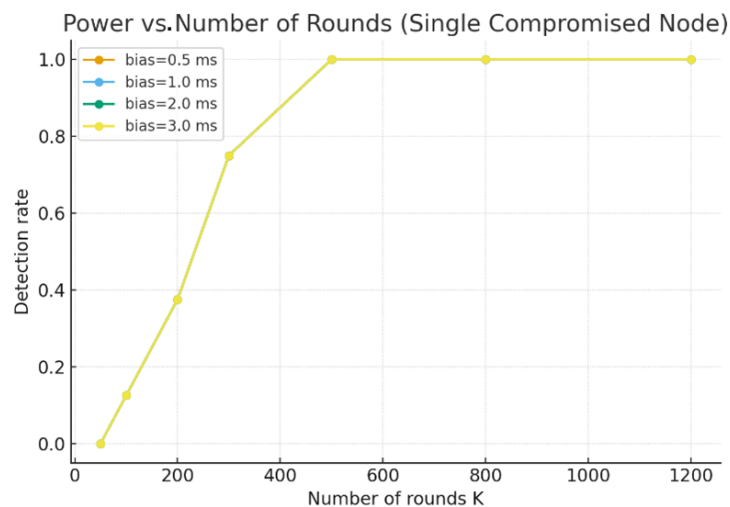


Figure 7. Detection power versus observation window size K . Power rises sharply after $K \approx 300$ and saturates near unity by $K \approx 500$, identifying a practical knee region.

Figure 8 illustrates heatmap detection of power as a function of bias magnitude and observation length K . Dark regions indicate weak detection at small K and low bias; bright regions show near-certain detection for larger K or stronger bias. The dark-to-bright transition band is the actionable design boundary for choosing K .

7.5. Scalability Analysis

The protocol's computational and message complexity scales favorably with swarm size. Per drone per round derived MAC key, the operations are constant: two hash evaluations and one delay calculation. The communication pattern involves one broadcast token from the master and at most one response broadcast from each drone, resulting in $O(N)$ messages per round. Table 5 summarizes the quantitative scalability metrics observed in the simulation. These trends indicate that the protocol can support medium-sized swarms without the overhead of certificate management or asymmetric key operations typically required by PKI-oriented approaches.

Table 5. Scalability Performance Metrics.

Number of Drones (N)	Avg. Comp. Time per Drone (ms)	Avg. Round Latency (ms)	Msgs per Round Derived MAC Key
5	0.8	8.2	6
10	0.8	9.1	11
25	0.9	11.7	26
50	1.0	14.3	51

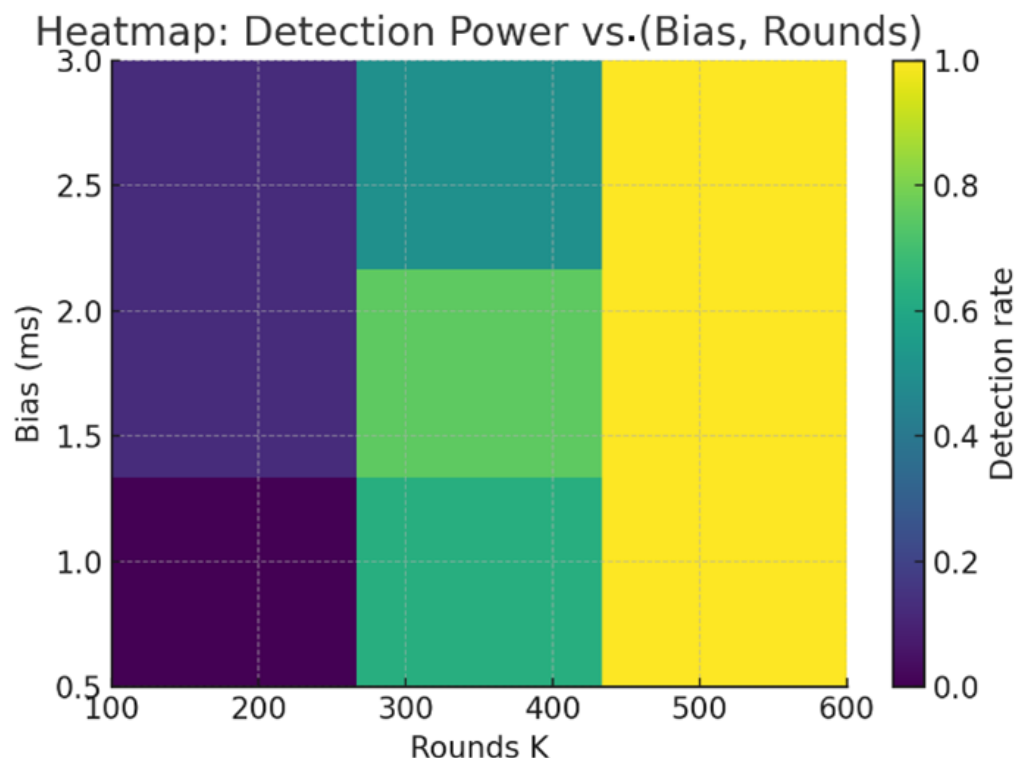


Figure 8. Detection power across bias magnitude and observation window size K . Dark regions indicate weak detection; bright regions show near-certain detection. The transition band marks the design threshold for choosing K .

8. Discussion and Evaluation

The experimental results substantiate the protocol's robustness and the effectiveness of its anomaly detection under realistic conditions. Table 4 demonstrates that enforcing token-source filtering preserves liveness under control-plane disruption: all 800,000 rounds remain valid in the filtered regime, and detection rates (0–0.125) align with occasional threshold crossings expected at a 5% significance level. In contrast, unfiltered acceptance degrades liveness without producing a persistent winner skew, as evidenced by detection rates near 0.125 across injection intensities. This behavior matches the intended separation of concerns: the chi-square statistic targets manipulation and cloning (i.e., skewed winner distributions), whereas desynchronization attacks primarily reduce the number of usable rounds.

The detection-power curves in Figures 7 and 8 reveal a practical knee region in the observation window size. Power increases monotonically with K , remains limited for $K \leq 100$, becomes reliable around $K \approx 300$, and approaches saturation by $K \approx 500$ for moderate biases. The heatmap shows a clear transition band from weak to strong detection as a function of bias magnitude and K ; in practice, designers should choose K beyond this transition for the bias scales expected in their operating environment. These findings directly inform deployment: (i) enforce master-source filtering to maintain liveness under token-injection attempts, and (ii) select K so that the expected manipulation bias lies in the high-power region of Figure 8, balancing detection latency against confidence.

Overall, the findings discussed in this section directly support the primary contributions stated in *Introduction*. First, the results substantiate the feasibility of our lightweight, decentralized authentication protocol for infrastructure-testing drone swarms rooted in HSPs, showing robust operation under realistic wireless conditions and disruption attempts. Second, the observed behavior aligns with the mathematical model of token-driven

HSP seed evolution, delay-based response ordering, and chi-square-based statistical validation: manipulation and cloning manifest as detectable winner-distribution skew, whereas desynchronization primarily affects liveness. Third, the simulation outcomes confirm scalability and effectiveness in terms of tamper detection and computational practicality; in particular, master-source token filtering preserves valid rounds under injection, and detection power exhibits a practical knee around $K \approx 300$, approaching saturation near $K \approx 500$ for moderate bias levels. Finally, the evaluation provides evidence consistent with our benchmarking claims against classical RSA-oriented and static PUF-only approaches, highlighting the advantages of our scheme for real-time, swarm-oriented settings where decentralization, low overhead, and rapid anomaly detection are essential.

Table 6 benchmarks the proposed protocol against representative schemes from the literature across key dimensions relevant to swarm operations.

Comparison with Related Work

The simulation outcomes are consistent with key tendencies reported in the authentication literature reviewed in Section 2. The observed feasibility of millisecond-scale rounds with lightweight per-node operations aligns with prior findings that PUF/HSP-rooted schemes can enable authentication without storing exportable long-term keys (and without network-wide pre-shared secrets), making them suitable for resource-constrained UAV/IoD settings [16,18,37]. In addition, the need for token validity checking under injection attempts is consistent with the known fragility of infrastructure-dependent or centrally managed approaches in intermittently connected swarms, where control-plane disruption can directly impact liveness [4,16]. The statistical winner-frequency monitoring employed in this work also complements trust- and behavior-based frameworks discussed in Section 2: rather than requiring long observation windows of high-dimensional behavior, the proposed detector leverages protocol-native response-order statistics, remaining lightweight while enabling anomaly exposure when an adversary attempts to bias participation [4,16]. Moreover, compared to RSA/PKI-oriented authentication—which incurs higher computational and certificate-management overhead and assumes persistent trusted infrastructure—the protocol’s computational scaling remains constant per node with $O(N)$ message activity per round; however, we also note that MAC-layer contention/collisions can introduce additional non-linear overhead as swarm size increases. The benchmarking comparison in Table 6 further contextualizes these findings against RSA-based and static PUF-only schemes, highlighting the advantages of the proposed approach for real-time swarm-oriented deployments.

In summary, the discussion and evaluation in this section substantiate the four primary contributions stated in the Introduction. (i) The results support the feasibility of the proposed lightweight, decentralized HSP-based swarm authentication protocol, including robust operation under token-injection when token-source filtering is enforced. (ii) The observed response-order behavior and winner-frequency statistics align with the mathematical formulation of HSP seed evolution, delay scheduling, and chi-square validation. (iii) The simulation study demonstrates practical scalability and efficiency under the evaluated settings, with near-constant per-node computation and $O(N)$ per-round messaging, alongside effective anomaly detection as the observation window increases. (iv) The benchmarking results in Table 5 contextualize these findings against RSA-based and static PUF-only schemes, highlighting advantages for real-time swarm deployments.

Table 6. Comparative benchmarking of authentication schemes for drone swarms.

Scheme Type	Decentralization	Cloning Resistance	Replay Resistance	Avg. Auth. Latency	Comm. Overhead per Node	Suitable for Real-Time Swarm
PKI-based [21]	Centralized	Low (CA-dependent)	High (crypto)	High (>100 ms)	High (certs, sigs)	No
Symmetric-key [25]	Partially centralized	Medium (key mgmt.)	Low (key exposure)	Low (~5 ms)	Medium (re-keying)	Limited
Trust-based [30]	Decentralized	High	Low (behavior mimicry)	Very high (observation)	High (trust msgs)	No
PUF-Static [18]	Decentralized	High	High	Very low (~1 ms)	Low	Limited (no replay prot.)
Proposed scheme	Fully decentralized	High	Very high	Low (~10 ms)	Low	Yes

9. Conclusions and Future Work

This study presented a decentralized, external infrastructure-free authentication protocol for swarm drones rooted in hardware security primitives (HSPs). By combining public token broadcasts with HSP-based hidden state evolution, delay-scheduled responses, and chi-square anomaly detection, the scheme achieves membership verification without pre-shared keys, certificates, or centralized authorities. Quantitatively, simulation experiments evaluated swarms of size $N = 5\text{--}50$ over $K = 100\text{--}1000$ authentication rounds (Table 3), demonstrating millisecond-scale operation with per-round latency of 8–15 ms under the configured delay parameters and near-constant per-drone computation dominated by lightweight hash evaluations. Robustness under control-plane disruption was confirmed: with token-source filtering enabled, all 800,000 rounds remained valid under injection scenarios, whereas unfiltered acceptance reduced valid rounds (e.g., to ≈ 722.75 at injection rate 0.1 and ≈ 556.63 at rate 0.3) and increased inconclusive rounds (Table 4). Regarding detection performance, the anomaly detector exhibits a practical observation-window knee: detection power is limited for $K \leq 100$, becomes reliable around $K \approx 300$, and approaches saturation by $K \approx 500$ for moderate bias levels (Figures 7 and 8), corresponding to a tamper detection rate exceeding 95% within 500 rounds in the tested settings. Overall, these results indicate lightweight operation, bounded liveness degradation under adversarial injection with appropriate token acceptance rules, and effective anomaly detection at swarm scale, making the protocol a promising foundation for securing next-generation autonomous swarms in adversarial environments.

Despite these advantages, future work will prioritize hardware-in-the-loop validation using FPGA-based prototypes (Arbiter PUF and ring-oscillator TRNG cores on a Xilinx Zynq platform) to quantify real-world timing, noise, and resource overheads. In parallel, we will extend the evaluation to incorporate large-scale wireless interference and high-mobility effects, investigate adaptive delay scheduling and consensus-driven master election for improved resilience, and study stronger adversaries (including coordinated collusion), as well as secure operation across heterogeneous swarms.

Beyond these technical extensions, the findings have broader implications for secure autonomous systems in defense, disaster response, and industrial IoT ecosystems. Future work will also explore adaptive trust metrics, integration with emerging swarm coordination standards, and side-channel hardening strategies to strengthen end-to-end security against physical compromise.

Author Contributions: Conceptualization, S.M.A. and B.H.S.A.; methodology, S.M.A. and B.H.S.A.; resources, S.M.A. and B.H.S.A.; writing—original draft preparation, B.H.S.A. and S.M.A.; writing—review and editing, S.M.A. and B.H.S.A.; visualization, B.H.S.A. and S.M.A. Supervision, B.H.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Jacobsen, R.H.; Matlekovic, L.; Shi, L.; Malle, N.; Ayoub, N.; Hageman, K.; Hansen, S.; Nyboe, F.F.; Ebeid, E. Design of an autonomous cooperative drone swarm for inspections of safety critical infrastructure. *Appl. Sci.* **2023**, *13*, 1256. [CrossRef]

2. Panigati, T.; Zini, M.; Striccoli, D.; Giordano, P.F.; Tonelli, D.; Limongelli, M.P.; Zonta, D. Drone-based bridge inspections: Current practices and future directions. *Autom. Constr.* **2025**, *173*, 106101. [\[CrossRef\]](#)
3. Aydin, Y.; Kurt, G.K.; Ozdemir, E.; Yanikomeroglu, H. Authentication and handover challenges and methods for drone swarms. *IEEE J. Radio Freq. Identif.* **2022**, *6*, 220–228. [\[CrossRef\]](#)
4. Wang, X.; Zhao, Z.; Yi, L.; Ning, Z.; Guo, L.; Yu, F.R.; Guo, S. A survey on security of UAV swarm networks: Attacks and countermeasures. *ACM Comput. Surv.* **2024**, *57*, 1–37. [\[CrossRef\]](#)
5. Shamsoshoara, A.; Korenda, A.; Afghah, F.; Zeadally, S. A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. *Comput. Netw.* **2020**, *183*, 107593. [\[CrossRef\]](#)
6. Al-Meer, A.; Al-Kuwari, S. Physical unclonable functions (PUF) for IoT devices. *ACM Comput. Surv.* **2023**, *55*, 1–31. [\[CrossRef\]](#)
7. Bellido, J.; Gao, Q. Scalable and Resilient Autonomous Drone Swarm Framework for Secure Operations in Threatened Environments. *Ubiquitous Technol. J.* **2025**, *1*, 1–9. [\[CrossRef\]](#)
8. Ahmad, S.M.; Samie, M.; Honarvar Shakibaei Asli, B. Building Trust in Autonomous Aerial Systems: A Review of Hardware-Rooted Trust Mechanisms. *Future Internet* **2025**, *17*, 466. [\[CrossRef\]](#)
9. Omolara, A.E.; Alawida, M.; Abiodun, O.I. Drone cybersecurity issues, solutions, trend insights and future perspectives: A survey. *Neural Comput. Appl.* **2023**, *35*, 23063–23101. [\[CrossRef\]](#)
10. Javaid, S.; Saeed, N.; Qadir, Z.; Fahim, H.; He, B.; Song, H.; Bilal, M. Communication and control in collaborative UAVs: Recent advances and future trends. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5719–5739. [\[CrossRef\]](#)
11. Sarkar, S.; Shafaei, S.; Jones, T.S.; Totaro, M.W. Secure Communication in Drone Networks: A Comprehensive Survey of Lightweight Encryption and Key Management Techniques. *Drones* **2025**, *9*, 583. [\[CrossRef\]](#)
12. Mohsan, S.A.H.; Othman, N.Q.H.; Li, Y.; Alsharif, M.H.; Khan, M.A. Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends. *Intell. Serv. Robot.* **2023**, *16*, 109–137. [\[CrossRef\]](#)
13. Chen, W.; Liu, J.; Guo, H.; Kato, N. Toward robust and intelligent drone swarm: Challenges and future directions. *IEEE Netw.* **2020**, *34*, 278–283. [\[CrossRef\]](#)
14. Javed, S.; Hassan, A.; Ahmad, R.; Ahmed, W.; Ahmed, R.; Saadat, A.; Guizani, M. State-of-the-art and future research challenges in uav swarms. *IEEE Internet Things J.* **2024**, *11*, 19023–19045. [\[CrossRef\]](#)
15. Rajasoundaran, S.; Kumar, S.S.; Selvi, M.; Kannan, A. Reactive handover coordination system with regenerative blockchain principles for swarm unmanned aerial vehicles. *Peer-Peer Netw. Appl.* **2024**, *17*, 89–114. [\[CrossRef\]](#)
16. Michailidis, E.T.; Vouyioukas, D. A review on software-based and hardware-based authentication mechanisms for the internet of drones. *Drones* **2022**, *6*, 41. [\[CrossRef\]](#)
17. Restuccia, F.; D'oro, S.; Melodia, T. Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet Things J.* **2018**, *5*, 4829–4842. [\[CrossRef\]](#)
18. Lounis, K.; Ding, S.H.; Zulkernine, M. D2D-MAP: A drone to drone authentication protocol using physical unclonable functions. *IEEE Trans. Veh. Technol.* **2022**, *72*, 5079–5093. [\[CrossRef\]](#)
19. Sen, M.A.; Al-Rubaye, S.; Tsourdos, A. Securing uav flying ad hoc wireless networks: Authentication development for robust communications. *Sensors* **2025**, *25*, 1194. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Shafique, A.; Mehmood, A.; Elhadef, M. Survey of security protocols and vulnerabilities in unmanned aerial vehicles. *IEEE Access* **2021**, *9*, 46927–46948. [\[CrossRef\]](#)
21. Jan, S.U.; Abbasi, I.A.; Algarni, F. A key agreement scheme for IoD deployment civilian drone. *IEEE Access* **2021**, *9*, 149311–149321. [\[CrossRef\]](#)
22. Irshad, A.; Alzahrani, B.A.; Albeshri, A.; Alsubhi, K.; Nayyar, A.; Chaudhry, S.A. SPAKE-DC: A secure PUF enabled authenticated key exchange for 5G-based drone communications. *IEEE Trans. Veh. Technol.* **2023**, *73*, 5770–5780. [\[CrossRef\]](#)
23. Khan, M.A.; Ullah, I.; Abdullah, A.M.; Mohsan, S.A.H.; Noor, F. An efficient and conditional privacy-preserving heterogeneous signcryption scheme for the Internet of drones. *Sensors* **2023**, *23*, 1063. [\[CrossRef\]](#)
24. Pandey, G.K.; Gurjar, D.S.; Nguyen, H.H.; Yadav, S. Security threats and mitigation techniques in UAV communications: A comprehensive survey. *IEEE Access* **2022**, *10*, 112858–112897. [\[CrossRef\]](#)
25. Nyangaresi, V.O.; Jasim, H.M.; Mutlaq, K.A.A.; Abduljabbar, Z.A.; Ma, J.; Abduljaleel, I.Q.; Honi, D.G. A symmetric key and elliptic curve cryptography-based protocol for message encryption in unmanned aerial vehicles. *Electronics* **2023**, *12*, 3688. [\[CrossRef\]](#)
26. Ashush, N.; Greenberg, S.; Manor, E.; Ben-Shimol, Y. Unsupervised drones swarm characterization using rf signals analysis and machine learning methods. *Sensors* **2023**, *23*, 1589. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Abduljabbar, Z.A.; Nyangaresi, V.O.; Ma, J.; Al Sibabee, M.A.; Khalefa, M.S.; Honi, D.G. MAC-based symmetric key protocol for secure traffic forwarding in drones. In *International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 16–36.
28. Du, X.; Li, Y.; Zhou, S.; Zhou, Y. ATS-LIA: A lightweight mutual authentication based on adaptive trust strategy in flying ad-hoc networks. *Peer-Peer Netw. Appl.* **2022**, *15*, 1979–1993. [\[CrossRef\]](#) [\[PubMed\]](#)

29. Ibibo, J.T.; Balota, J.E.; Alwada'n, T.; Akinade, O.O. Enhancing IoT-LLN Security with IbiboRPLChain Solution: A Blockchain-Based Authentication Method. *Appl. Sci.* **2025**, *15*, 10557. [\[CrossRef\]](#)
30. Wang, S.; Ding, W.; Li, J.; Yuan, Y.; Ouyang, L.; Wang, F.Y. Decentralized autonomous organizations: Concept, model, and applications. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 870–878. [\[CrossRef\]](#)
31. Alharbi, A.; Petrunin, I.; Panagiotakopoulos, D. Assuring safe and efficient operation of UAV using explainable machine learning. *Drones* **2023**, *7*, 327. [\[CrossRef\]](#)
32. Shirvani, M.H.; Masdari, M. A survey study on trust-based security in Internet of Things: Challenges and issues. *Internet Things* **2023**, *21*, 100640. [\[CrossRef\]](#)
33. Mall, P.; Amin, R.; Das, A.K.; Leung, M.T.; Choo, K.K.R. PUF-based authentication and key agreement protocols for IoT, WSNs, and smart grids: A comprehensive survey. *IEEE Internet Things J.* **2022**, *9*, 8205–8228. [\[CrossRef\]](#)
34. Barbareschi, M.; Casola, V.; De Benedictis, A.; La Montagna, E.; Mazzocca, N. On the adoption of physically unclonable functions to secure IoT devices. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7781–7790. [\[CrossRef\]](#)
35. Wali, A.; Das, S. Hardware and information security primitives based on 2D materials and devices. *Adv. Mater.* **2023**, *35*, 2205365. [\[CrossRef\]](#)
36. Tehranipoor, M.; Pundir, N.; Vashistha, N.; Farahmandi, F. *Hardware Security Primitives*; Springer: Berlin/Heidelberg, Germany, 2023.
37. Choi, J.; Son, S.; Kwon, D.; Park, Y. A puf-based secure authentication and key agreement scheme for the internet of drones. *Sensors* **2025**, *25*, 982. [\[CrossRef\]](#)
38. Tanveer, M.; Aldosary, A.; Khokhar, S.-u.d.; Das, A.K.; Aldossari, S.A.; Chaudhry, S.A. PAF-IoD: PUF-enabled authentication framework for the Internet of Drones. *IEEE Trans. Veh. Technol.* **2024**, *73*, 9560–9574. [\[CrossRef\]](#)
39. Cho, S.M.; Hong, E.; Seo, S.H. Random number generator using sensors for drone. *IEEE Access* **2020**, *8*, 30343–30354. [\[CrossRef\]](#)
40. U.S. Department of Transportation. *Unmanned Aircraft System (UAS) Service Demand 2015–2035: Literature Review & Projections of Future Usage*; Technical Report DOT-VNTSC-DoD-15-01; Volpe National Transportation Systems Center: Cambridge, CA, USA, 2015.
41. Abdelhaleem, S.H.; Abd-El-Hafiz, S.K.; Radwan, A.G. Analysis and guidelines for different designs of pseudo random number generators. *IEEE Access* **2024**, *12*, 115697–115715. [\[CrossRef\]](#)
42. Barakbayeva, T.; Cai, Z.; Goharshady, A.K. SRNG: An efficient decentralized approach for secret random number generation. In Proceedings of the 2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Dublin, Ireland, 27–31 May 2024; IEEE: Piscataway, NY, USA, 2024; pp. 615–619.
43. Ali, A.; Ali, H.; Saeed, A.; Ahmed Khan, A.; Tin, T.T.; Assam, M.; Ghadi, Y.Y.; Mohamed, H.G. Blockchain-powered healthcare systems: Enhancing scalability and security with hybrid deep learning. *Sensors* **2023**, *23*, 7740. [\[CrossRef\]](#)
44. Dang, T.K.; Nguyen, K.D.; Hoang, T.T.; Pham, C.K. A Unified Approach to Strong PUF and TRNG Using Ring Generator for Cryptography. *IEEE Internet Things J.* **2025**, *12*, 36590–36603. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



DSPACE

<https://dspace.org/>

Secure and decentralised swarm authentication using hardware security primitives

Ahmad, Sagir Muhammad; Honarvar Shakibaei Asli, Barmak

2026-01-02

MDPI

<https://doi.org/10.3390/electronics15020423>

Ahmad SM, Honarvar Shakibaei Asli B. (2026) Secure and decentralised swarm authentication using hardware security primitives. Electronics, Volume 15, Issue 2, January 2026, Article number 423