future internet

Article

Enhancing Communication Security in Drones Using QRNG in Frequency Hopping Spread Spectrum

J. de Curtò [1,2,3,*], I. de Zarzà [3,4], Juan-Carlos Cano [5] and Carlos T. Calafate [5]

1 Department of Computer Applications in Science & Engineering, BARCELONA Supercomputing Center, 08034 Barcelona, Spain
2 Escuela Técnica Superior de Ingeniería (ICAI), Universidad Pontificia Comillas, 28015 Madrid, Spain
3 Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, 08018 Barcelona, Spain; izarza@unizar.es
4 Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 50009 Zaragoza, Spain
5 Departamento de Informática de Sistemas y Computadores, Universitat Politècnica de València, 46022 València, Spain; jucano@disca.upv.es (J.-C.C.); calafate@disca.upv.es (C.T.C.)
* Correspondence: jdecurto@icai.comillas.edu

Abstract: This paper presents a novel approach to enhancing the security and reliability of drone communications through the integration of Quantum Random Number Generators (QRNG) in Frequency Hopping Spread Spectrum (FHSS) systems. We propose a multi-drone framework that leverages QRNG technology to generate truly random frequency hopping sequences, significantly improving resistance against jamming and interception attempts. Our method introduces a concurrent access protocol for multiple drones to share a QRNG device efficiently, incorporating robust error handling and a shared memory system for random number distribution. The implementation includes secure communication protocols, ensuring data integrity and confidentiality through encryption and Hash-based Message Authentication Code (HMAC) verification. We demonstrate the system's effectiveness through comprehensive simulations and statistical analyses, including spectral density, frequency distribution, and autocorrelation studies of the generated frequency sequences. The results show a significant enhancement in the unpredictability and uniformity of frequency distributions compared to traditional pseudo-random number generator-based approaches. Specifically, the frequency distributions of the drones exhibited a relatively uniform spread across the available spectrum, with minimal discernible patterns in the frequency sequences, indicating high unpredictability. Autocorrelation analyses revealed a sharp peak at zero lag and linear decrease to zero values for other lags, confirming a general absence of periodicity or predictability in the sequences, which enhances resistance to predictive attacks. Spectral analysis confirmed a relatively flat power spectral density across frequencies, characteristic of truly random sequences, thereby minimizing vulnerabilities to spectral-based jamming. Statistical tests, including Chi-squared and Kolmogorov-Smirnov, further confirm the unpredictability of the frequency sequences generated by QRNG, supporting enhanced security measures against predictive attacks. While some short-term correlations were observed, suggesting areas for improvement in QRNG technology, the overall findings confirm the potential of QRNG-based FHSS systems in significantly improving the security and reliability of drone communications. This work contributes to the growing field of quantum-enhanced wireless communications, offering substantial advancements in security and reliability for drone operations. The proposed system has potential applications in military, emergency response, and secure commercial drone operations, where enhanced communication security is paramount.

Keywords: frequency hopping spread spectrum (FHSS); quantum random number generator (QRNG); random processing units; quantum computing; drone communications; quantum enhanced communications

check for updates

Citation: de Curtò, J.; de Zarzà, I.; Cano, J.-C.; Calafate, C.T. Enhancing Communication Security in Drones Using QRNG in Frequency Hopping Spread Spectrum. *Future Internet* **2024**, *16*, 412. https://doi.org/10.3390/fi16110412

Academic Editor: Paolo Bellavista

Received: 8 October 2024
Revised: 31 October 2024
Accepted: 6 November 2024
Published: 8 November 2024

1. Introduction

In an increasingly digital and interconnected world, and with the advent of sophisticated AI techniques [1], the need for secure communication channels has never been more critical. Sensitive information, whether personal, corporate, or governmental, must be protected against a myriad of threats including eavesdropping, data interception, and deliberate jamming. Frequency Hopping Spread Spectrum (FHSS) [2,3] has long been employed as a technique to enhance the security and robustness of wireless communications.

The ever-increasing demand for secure communication channels has led to the exploration of advanced techniques that can withstand interception and jamming attempts. Frequency Hopping Spread Spectrum (FHSS) is a well-established method to enhance communication security. This article explores the potential of using a Randomness Processing Unit (RPU) to generate true random sequences for frequency hopping, thereby significantly enhancing the unpredictability and robustness of communication systems. By rapidly changing the carrier frequency according to a specific sequence, FHSS spreads the signal over a wide spectrum, making it more resistant to interference and more difficult to intercept.

Traditionally, the sequences used for frequency hopping are generated using pseudorandom number generators (PRNGs). While these sequences can appear random, they are ultimately deterministic and, given enough time and resources, can potentially be predicted or reverse-engineered. This inherent vulnerability poses a significant risk to communication systems, especially in environments where security is paramount. Improvements over traditional PRNGs [4–6] have been explored in the literature, with emphasis on an efficient implementation and portability.

The advent of Randomness Processing Units (RPUs) [7,8] offers a promising solution to this challenge. RPUs are specialized hardware devices capable of generating true random numbers based on inherently unpredictable physical processes, such as quantum noise or thermal fluctuations. Unlike PRNGs, the numbers generated by RPUs are not reproducible or predictable, even with complete knowledge of the system's internal workings.

This manuscript explores the concept of leveraging RPUs to generate true random sequences [9,10] for frequency hopping in communication systems. By replacing deterministic sequences with truly random ones, we can significantly enhance the security and robustness of FHSS. We will study the theoretical background of FHSS, the mathematical formulation of true random frequency hopping, and the security benefits and implementation challenges associated with this approach. A preliminary use case with an actual QRNG device setup in a multi-drone configuration is presented. This exploration aims to highlight the importance of incorporating true randomness in securing modern communication infrastructures.

The paper is organized as follows: Section 2 provides an overview and related works while Section 3 provides the theoretical background and considerations for Frequency Hopping Spread Spectrum (FHSS) and the use of Randomness Processing Units (RPUs). In Section 4, we present our methodology for integrating an RPU to generate truly random frequency hopping sequences, enhancing the security of FHSS systems. Section 5 details the implementation aspects, including the setup of the RPU, generation and synchronization of frequency hopping sequences, and security protocols. In Section 6, we apply the proposed method to a drone cloud with a ring topology, demonstrating its benefits in multi-drone communications. Section 7 presents the evaluations, including experimental setup, procedures, and results that demonstrate the effectiveness of our approach. Section 8 explores practical applications and potential future developments of the system. Finally, Section 9 concludes the paper and outlines future research directions.

2. Related Works

Frequency Hopping Spread Spectrum (FHSS) has long been recognized as a robust method for secure wireless communications due to its resistance to interference and eavesdropping. The concept of FHSS was first patented by Lamarr and Antheil [11], where the carrier frequency is rapidly switched according to a specific sequence known only to the communicating parties. Traditional FHSS systems utilize Pseudo-Random Number Generators (PRNGs) to create these hopping sequences. While PRNGs are computationally efficient, they are inherently deterministic, making them susceptible to prediction and reverse engineering if an adversary intercepts sufficient portions of the communication [12].

The limitations of PRNGs in security-critical applications have spurred interest in True Random Number Generators (TRNGs) and, more recently, Quantum Random Number Generators (QRNGs). QRNGs exploit fundamental quantum mechanical processes, such as quantum vacuum fluctuations and photon emission events, to produce truly random numbers that are inherently unpredictable [13]. Jennewein et al. [14] demonstrated an early implementation of a QRNG using photon polarization measurements, providing a significant enhancement in randomness over classical TRNGs.

In the context of cryptographic systems, secure key distribution and encryption are paramount. The RSA algorithm [15], developed by Rivest, Shamir, and Adleman, revolutionized public-key cryptography by utilizing the computational difficulty of factoring large composite numbers. RSA allows for secure key exchange over unsecured channels, which is essential for initializing secure communications. However, RSA's security relies on key lengths that may be computationally intensive for resource-constrained devices.

Elliptic Curve Cryptography (ECC) [16,17] offers similar security levels to RSA but with smaller key sizes, making it more suitable for devices with limited computational resources, such as drones and IoT devices. ECC's efficiency stems from the mathematical complexity of the elliptic curve discrete logarithm problem, which remains intractable for sufficiently large key sizes.

For symmetric encryption, the Advanced Encryption Standard (AES) [18] has become the industry standard due to its balance of security and performance. AES is employed for encrypting data streams after the initial key exchange, ensuring confidentiality and integrity during communication sessions. The combination of public-key cryptography for key exchange (RSA or ECC) and symmetric encryption (AES) provides a robust framework for secure communications.

Recent advancements in integrating QRNGs into communication systems have shown promise in enhancing security. Abellán et al. [19] utilized quantum phase fluctuations in a laser diode to generate high-speed random numbers suitable for cryptographic applications. Such QRNGs can be integrated into FHSS systems to produce truly random hopping sequences, significantly increasing the difficulty for adversaries attempting to predict or jam the communication.

In the field of drone communications, security is a critical concern due to the increasing reliance on unmanned aerial vehicles (UAVs) for military, commercial, and civilian applications. Existing research has explored various aspects of secure drone communications, including secure routing protocols [20], intrusion detection systems [21], and encryption schemes tailored for UAV networks [22]. However, the application of QRNG-based FHSS in drone networks remains relatively unexplored.

Frequency hopping in wireless communication systems, while providing security and robustness against interference and eavesdropping, introduces challenges related to scheduling and potential interruption of connections. Efficient scheduling is crucial to ensure seamless communication, especially in dynamic networks such as multi-hop UAV systems [23] and IoT Networks [24].

Our work builds upon these foundational studies by integrating QRNGs into FHSS systems specifically for multi-drone networks. By employing secure key exchange mechanisms such as RSA or ECC for the initial distribution of synchronization information and AES for encrypting the frequency hopping sequences, we address the challenges associated with synchronization and security in true random FHSS systems. This integration enhances the unpredictability of the hopping sequences and provides a robust defense against jamming and interception attempts.

Furthermore, the implementation of a concurrent access protocol and a shared memory system for random number distribution ensures that multiple drones can efficiently share a single QRNG device without compromising the quality of randomness. This approach optimizes resource utilization in scenarios where equipping each drone with its own QRNG may be impractical due to size, weight, or power constraints.

By combining quantum randomness with established cryptographic practices, our methodology offers a significant advancement in securing drone communications. It addresses the vulnerabilities associated with deterministic PRNGs and enhances the overall resilience of the communication network against sophisticated attacks.

3. Theoretical Background and Considerations

In secure communication systems, protecting the integrity and confidentiality of transmitted data is crucial. FHSS is a technique used to achieve these goals by distributing the signal across a wider bandwidth, thereby making it more resistant to interference and eavesdropping. This section provides a detailed mathematical foundation for FHSS, explores the generation of frequency hopping sequences, and introduces the concept of using Randomness Processing Units (RPUs) to enhance security through true random sequences.

3.1. Frequency Hopping Spread Spectrum (FHSS)

FHSS operates by changing the carrier frequency of a signal at regular intervals according to a specific hopping sequence. The signal $S(t)$ is transmitted across different frequencies over time, making it difficult for an unauthorized party to intercept or jam the communication. The mathematical model of the transmitted signal can be expressed as:

$$S(t) = \sum_{o=1}^{N} A_o \cos(2\pi f_o t + \phi_o) \cdot \mathbf{1}_{[t_o, t_{o+1})}(t) \tag{1}$$

where:

- A_o is the amplitude of the signal during the o-th hop,
- f_o is the carrier frequency for the o-th time interval,
- ϕ_o is the phase for the o-th time interval,
- $\mathbf{1}_{[t_o, t_{o+1})}(t)$ is an indicator function that is 1 if $t \in [t_o, t_{o+1})$ and 0 otherwise,
- t_o and t_{o+1} are the start and end times of the o-th hop, respectively,
- N is the total number of frequency hops.

Frequency Hopping Sequence

The sequence of frequencies $\{f_o\}$ used in FHSS is typically generated using a PRNG. Let $\{r_o\}$ be a sequence of pseudo-random numbers generated by a PRNG. The frequency for the o-th hop can be determined as:

$$f_o = f_{\min} + (f_{\max} - f_{\min}) \frac{r_o}{R} \tag{2}$$

where:

- $f_{\min}$ and $f_{\max}$ are the minimum and maximum frequencies in the hopping band,
- r_o is the pseudo-random number for the o-th hop, and
- R is the range of the PRNG (i.e., the maximum possible value of r_o).

The key challenge with PRNG-based sequences is that, given enough output samples, an adversary can potentially reconstruct the sequence generation process, posing a security risk.

3.2. True Random Frequency Hopping with RPUs

RPUs are hardware devices designed to produce true random numbers by harnessing unpredictable physical phenomena, such as quantum noise or thermal fluctuations. These numbers are fundamentally unpredictable and cannot be reproduced, providing a higher level of security compared to pseudo-random sequences.

Let $\{R_o\}$ denote a sequence of true random numbers generated by an RPU. The corresponding frequency hopping sequence can be represented as:

$$f_o = f_{\min} + (f_{\max} - f_{\min}) \frac{R_o}{R_{\max}} \tag{3}$$

where:

- R_o is the true random number for the o-th hop,
- $R_{\max}$ is the maximum value that R_o can take.

The entropy H of the true random sequence $\{R_o\}$ is given by:

$$H = -\sum_o P(R_o) \log P(R_o) \tag{4}$$

In the case of uniformly distributed random numbers, $P(R_o) = \frac{1}{R_{\max}}$ for all o, leading to:

$$H = \log R_{\max} \tag{5}$$

This maximum entropy ensures that the sequence is completely unpredictable.

The process of generating and synchronizing the frequency hopping sequence is depicted in Figure 1. This flowchart indicates the steps from generating true random numbers to initiating FHSS communication.

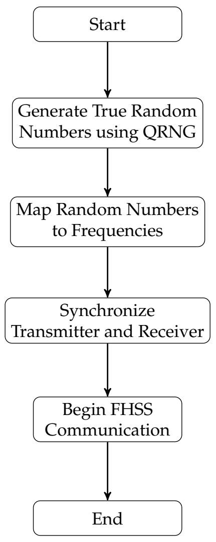
Start
Generate True Random Numbers using QRNG
Map Random Numbers to Frequencies
Synchronize Transmitter and Receiver
Begin FHSS Communication
End

Figure 1. Flowchart of the frequency hopping sequence generation and synchronization process.

The primary security advantage of using RPUs lies in the true randomness of the generated sequences. Even if an adversary has access to the transmitted signal, they cannot predict future frequency hops without direct access to the RPU or the shared random sequence. This feature significantly enhances the robustness of the system against interception and jamming.

3.3. Statistical Properties and Analysis

To evaluate the statistical properties of the sequence, we can analyze the autocorrelation function, which measures the similarity between the sequence at different time shifts. For a sequence $\{f_o\}$, the autocorrelation function $R_f(\tau)$ is defined as:

$$R_f(\tau) = \mathbb{E}[f_o f_{o+\tau}] - \mathbb{E}[f_o]\mathbb{E}[f_{o+\tau}] \tag{6}$$

For a true random sequence, $R_f(\tau) \approx 0$ for $\tau \neq 0$, indicating that there is no predictable pattern over time.

The power spectral density (PSD) of the frequency-hopped signal provides insight into how the signal's power is distributed across the frequency spectrum. The PSD of a signal $S(t)$ is defined as:

$$S_{SS}(f) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\left|\int_{-T/2}^{T/2} S(t)e^{-j2\pi ft}dt\right|^2\right] \tag{7}$$

For a system using true random FHSS, the PSD is expected to be flat across the hopping band, indicating uniform distribution of signal power.

This detailed mathematical foundation underscores the robustness and security advantages of using true random sequences for frequency hopping, as opposed to deterministic or pseudo-random sequences. The next sections will delve into the security analysis and practical considerations for implementing such a system.

4. Methodology and System Architecture

In this section, we present our methodology to enhancing the security and robustness of FHSS systems by integrating a Randomness Processing Unit (RPU) to generate truly random frequency hopping sequences. The core idea of our methodology is to use the RPU to produce a sequence of true random numbers that are used to determine the hopping frequencies. Both the transmitter and receiver have access to this same sequence of random numbers, enabling them to stay synchronized during communication. This is achieved through secure key distribution or pre-sharing the random sequence prior to communication. By using true random numbers, the frequency hopping pattern becomes entirely unpredictable to an adversary, significantly enhancing security against eavesdropping and jamming attacks.

Unlike traditional FHSS systems that rely on pseudo-random number generators (PRNGs) and shared seeds, which can be potentially predicted or replicated by attackers, our system's use of true randomness ensures that the hopping sequence cannot be reproduced without direct knowledge of the random numbers generated by the RPU. The synchronization between the transmitter and receiver does not depend on generating the same sequence independently but on both parties accessing the same random sequence securely. Secure key distribution mechanisms, such as public-key cryptography or pre-shared keys, are employed to share the random sequence or the necessary keys to access it. This approach maintains synchronization and ensures that only authorized parties can participate in the communication, thereby enhancing overall system security.

Figure 2 illustrates the overall system architecture, highlighting the key components and their interactions within the proposed method.

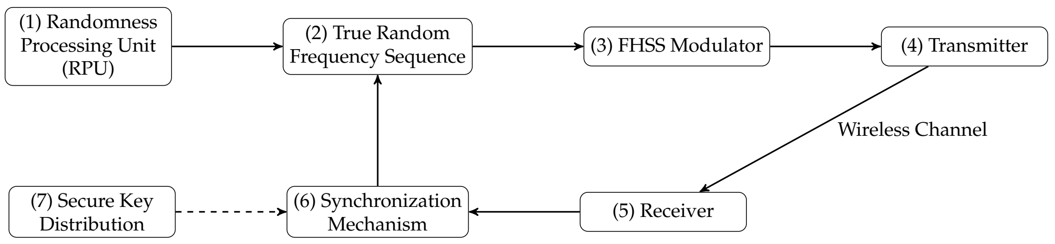
(1) Randomness Processing Unit (RPU)
(2) True Random Frequency Sequence
(3) FHSS Modulator
(4) Transmitter
Wireless Channel
(5) Receiver
(6) Synchronization Mechanism
(7) Secure Key Distribution

Figure 2. Diagram illustrating the methodology of using a Randomness Processing Unit (RPU) for true random frequency hopping in an FHSS system.

In FHSS, the carrier frequency is changed according to a predefined sequence [25,26], hopping between different frequencies in a seemingly random fashion. This method spreads the signal over a wider bandwidth, providing robustness against narrowband interference and eavesdropping. The sequence of frequency hops is traditionally generated using pseudo-random number generators (PRNGs), which, while complex, are ultimately deterministic and potentially predictable [27,28].

Next, we provide a detailed explanation of how each component interacts and the data flow between them.

1. Randomness Processing Unit (RPU): The RPU generates a stream of true random numbers by exploiting inherent physical phenomena such as quantum noise or thermal fluctuations. These random numbers form the basis of the frequency hopping sequence and are critical for ensuring unpredictability in the system.
2. True Random Frequency Sequence Generator: The random numbers produced by the RPU are processed to generate a sequence of frequencies $\{f_o\}$. This involves mapping the random numbers to specific frequencies within the available frequency band F. The mapping ensures a uniform probability distribution over F, maximizing entropy. The frequency sequence generator interfaces directly with the RPU to receive the random numbers and with the FHSS modulator to provide the frequency sequence.
3. FHSS Modulator: The FHSS modulator receives the true random frequency sequence from the frequency sequence generator. It modulates the carrier signal according to the frequency f_o specified for each time interval $[t_o, t_{o+1})$. The modulator also incorporates the data to be transmitted, applying appropriate modulation techniques (e.g., frequency shift keying or phase shift keying). The modulated signal is then passed to the transmitter.
4. Transmitter: The transmitter amplifies the modulated signal and broadcasts it over the wireless channel. It ensures that the signal conforms to regulatory standards for transmission power and spectral occupancy. The transmitter interfaces with the FHSS modulator for input and communicates over the wireless channel to the receiver.
5. Receiver: On the receiving end, the receiver captures the signal from the wireless channel. It must be synchronized with the transmitter to correctly follow the frequency hopping sequence. The receiver demodulates the signal using the same frequency sequence $\{f_o\}$, extracting the transmitted data. The receiver interfaces with the synchronization mechanism to maintain alignment with the transmitter's frequency hopping pattern.
6. Synchronization Mechanism: The synchronization mechanism is critical for maintaining alignment between the transmitter and receiver. It ensures that both ends use the same frequency f_o at any given time t. Synchronization information, such as initial timing parameters and any required updates, is communicated through a secure channel or embedded within the communication protocol. The synchronization mechanism interfaces with both the frequency sequence generator (to obtain or verify the frequency sequence) and the receiver (to adjust timing and sequence alignment).
7. Secure Key Distribution: The secure key distribution component facilitates the exchange of keys or synchronization information necessary for secure communication. It interfaces with the synchronization mechanism to provide the required cryptographic keys or parameters. Secure channels, possibly using public-key cryptography, are employed to prevent interception or tampering during key exchange.

The data flow begins with the RPU generating true random numbers, which are converted into a frequency sequence by the frequency sequence generator. This sequence is fed into the FHSS modulator, along with the data to be transmitted. The FHSS modulator outputs a frequency-hopped, modulated signal to the transmitter. The transmitter broadcasts the signal over the wireless channel to the receiver. The receiver, synchronized via the synchronization mechanism, captures and demodulates the signal using the same frequency sequence to retrieve the data. Throughout this process, the synchronization mechanism and secure key distribution ensure that both ends maintain alignment and that the communication remains secure. Implementing dynamic control architectures and robust communication protocols is essential for swarm unmanned aerial vehicles to function effectively [29].

Let $S(t)$ represent the transmitted signal. In a traditional FHSS system, the carrier frequency at time t is determined by a sequence $\{f_o\}$, where $o = 1, 2, \ldots, N$ and N is the total number of hops in the sequence. The frequency-hopped signal can be expressed as:

$$S(t) = \sum_{o=1}^{N} A_o \cos(2\pi f_o t + \phi_o) \cdot \mathbf{1}_{[t_o, t_{o+1})}(t)$$

.

Here, A_o represents the amplitude, ϕ_o the phase, and $\mathbf{1}_{[t_o, t_{o+1})}(t)$ is an indicator function defining the time interval for each hop.

4.1. True Random Frequency Hopping

With an RPU, the sequence $\{f_o\}$ is generated by a truly random process. If f_o is chosen uniformly from a set of available frequencies F, the probability distribution of selecting any particular frequency is:

$$P(f_o = f_z) = \frac{1}{|F|} \tag{8}$$

where $|F|$ is the cardinality of the set F. The unpredictability of the frequency hops, characterized by entropy H, is maximized:

$$H = -\sum_{f \in F} P(f) \log P(f) = \log |F| \tag{9}$$

This high entropy ensures that even with complete knowledge of the hopping algorithm, an adversary cannot predict the sequence, as it is truly random.

The primary security advantage of true random frequency hopping lies in its anti-jamming capabilities. Given that the sequence is entirely unpredictable, a jammer must either cover the entire bandwidth (which requires considerable power) or fail to disrupt the communication. The jamming margin, defined as the difference between the bandwidth of the spread spectrum and the jamming bandwidth, is maximized.

The security of the system is further enhanced because eavesdroppers cannot anticipate the next frequency hop. The information-theoretic security provided by the truly random sequence means that even with infinite computational power, an adversary cannot predict future hops without the actual random sequence.

The primary challenge with true random FHSS is synchronization between the transmitter and receiver. Both ends must be perfectly synchronized in terms of the hopping sequence. This can be achieved through secure key distribution mechanisms or synchronization protocols.

The true random sequence or the seed value for the RPU must be securely shared between the communicating parties. This can be done using public-key cryptography or other secure channels.

The implementation of our RPU-based FHSS system presents both challenges and opportunities for enhancing communication security. One key consideration is the rate at which the RPU can generate random numbers, as this directly impacts the frequency hopping rate and, consequently, the system's resilience to jamming attempts. To address potential latency issues, we employ a buffer system that pre-generates a pool of random frequencies, ensuring a continuous supply for rapid hopping. Additionally, we implement a dynamic hopping rate that adapts to the perceived threat level; under normal conditions, the system may use a moderate hopping rate to conserve power, but it can switch to a more aggressive rate when jamming or interception is suspected. The RPU's true randomness also allows for the implementation of a frequency exclusion list, dynamically avoiding frequencies that are detected to be under interference, without compromising the unpredictability of the overall sequence. This adaptive approach, coupled with the inherent unpredictability of the RPU, provides a significant advantage over traditional PRNG-based systems in terms of security and flexibility.

4.2. Buffer System Implementation

To address potential latency issues associated with real-time random number generation, we implement a buffer system that pre-generates and stores a pool of random frequencies. The buffer operates as a circular queue with a size B, determined based on the maximum expected hopping rate and the RPU's generation capabilities. The algorithm for buffer management, as described in Algorithm 1, ensures that the buffer is replenished proactively, maintaining a threshold level $B_{\min}$ to prevent underrun. This approach decouples the random number generation process from the frequency hopping mechanism, allowing for continuous operation even if momentary delays occur in the RPU output.

Algorithm 1 Algorithm for Buffer Management

```
 1: **Initialization:**
 2: Set buffer size $B$ and minimum threshold $B_{\min}$.
 3: Initialize buffer pointer $p \leftarrow 0$.
 4: **while** True **do**
 5:     **if** Buffer_Size $\leq B_{\min}$ **then**                ▷ Buffer Replenishment
 6:         Generate random number $R_o$ from RPU.
 7:         Map $R_o$ to frequency $f_o$.
 8:         Store $f_o$ in buffer at position $p$.
 9:         Update pointer $p \leftarrow (p+1) \mod B$.
10:     **end if**
11:     **if** Time to Hop **then**                               ▷ Frequency Hopping
12:         Retrieve next frequency $f_{\text{hop}}$ from buffer.
13:         Transmit using frequency $f_{\text{hop}}$.
14:     **end if**
15: **end while**
```

4.3. Dynamic Hopping Rate Implementation

The system can adjust the frequency hopping rate R_h based on the perceived threat level. A threat detection module monitors indicators such as increased error rates, signal interference patterns, and unexpected signal strength variations. If the module detects anomalies exceeding predefined thresholds, it can signal the control unit to increase R_h within allowable limits $[R_{h_{\min}}, R_{h_{\max}}]$. Conversely, in low-threat environments, R_h is decreased to conserve energy.

Threat Detection Algorithm

The threat detection algorithm operates as described in Algorithm 2.

Algorithm 2 Threat Detection and Hopping Rate Adjustment

```
 1: **while** Communication is Active **do**
 2:     Monitor communication metrics:
        •  Bit Error Rate (BER)
        •  Signal-to-Noise Ratio (SNR)
        •  Received Signal Strength Indicator (RSSI)
 3:     **if** Any metric exceeds threshold **then**
 4:         Set threat level Threat_Level $\leftarrow$ High.
 5:         Adjust hopping rate $R_h \leftarrow \min(R_h + \Delta R, R_{h_{\max}})$.
 6:     **else**
 7:         Set threat level Threat_Level $\leftarrow$ Low.
 8:         Adjust hopping rate $R_h \leftarrow \max(R_h - \Delta R, R_{h_{\min}})$.
 9:     **end if**
10: **end while**
```

To clarify the operation of the Threat Detection Algorithm and address concerns about how often the system updates its threat assessment and adjusts the frequency hopping rate, we provide additional details on the timing and responsiveness of the mechanism.

The Threat Detection Algorithm can operate continuously during active communication, monitoring key communication metrics such as Bit Error Rate (BER), Signal-to-Noise Ratio (SNR), and Received Signal Strength Indicator (RSSI). The assessment of these metrics occurs at regular intervals defined by the system's monitoring frequency f_{monitor}, which can be configured based on the specific requirements and capabilities of the system. In high-security environments, f_{monitor} can be set to evaluate metrics every hop interval, allowing the system to detect threats and respond in real-time.

Upon detecting that any monitored metric exceeds its predefined threshold, indicating a potential attack or interference, the system increases the frequency hopping rate R_h by a step size ΔR. The adjustment is immediate and can occur within a single hop interval, enabling the system to respond swiftly to threats. The time it takes to adjust from the minimum hopping rate $R_{h_{\min}}$ to the maximum hopping rate $R_{h_{\max}}$ depends on the value of ΔR and the frequency of metric assessments. For example, if ΔR is set such that R_h increases by a fixed percentage or value each time a threat is detected, and assessments occur every hop, the total time T_{adjust} to reach $R_{h_{\max}}$ is:

$$T_{\text{adjust}} = \left\lceil \frac{R_{h_{\max}} - R_{h_{\text{current}}}}{\Delta R} \right\rceil \times T_{\text{hop}} \tag{10}$$

where $R_{h_{\text{current}}}$ is the current hopping rate at the time of the first threat detection, and $T_{\text{hop}} = \frac{1}{R_h}$ is the duration of each hop.

To prevent rapid oscillations of the hopping rate due to transient fluctuations or false positives, the system may incorporate hysteresis or require that the thresholds be exceeded for a certain number of consecutive assessments before adjusting R_h. Additionally, when the threat subsides—indicated by metrics returning below threshold levels—the system decreases R_h gradually to conserve resources, ensuring a balance between security and efficiency.

In our current implementation, the dynamic adjustment of R_h in response to detected threats is conceptual and serves to illustrate how the methodology could be applied in real-world scenarios. We consider the integration of real-time threat detection and hopping rate adjustment to be feasible with appropriate hardware and software optimizations. Implementing this functionality would involve configuring the monitoring frequency f_{monitor}, defining appropriate thresholds for the metrics, and determining suitable values for ΔR based on the system's performance requirements and operational environment.

4.4. Frequency Exclusion List Management

Interference detection can be performed using real-time spectrum analysis. Frequencies exhibiting high noise levels or consistent transmission errors are added to a frequency exclusion list F_{excl}. The system updates the set of available frequencies $F_{\text{avail}} = F \setminus F_{\text{excl}}$ dynamically, where the operator "$\setminus$" denotes the set difference between two sets. This means that F_{avail} consists of all frequencies in F that are not in F_{excl}. By removing the excluded frequencies from the full set, we ensure that only reliable frequencies are used for communication. To maintain randomness, the RPU can be informed of F_{avail} so that it only maps random numbers to permissible frequencies without altering the underlying entropy of the sequence.

Interference Detection and Frequency Update Algorithm

To ensure that randomness is preserved across the adjusted set of available frequencies, the mapping function within the Random Processing Unit (RPU) is modified to account for the updated set of frequencies, F_{avail}, as described in Algorithm 3. This adjustment process involves normalizing the generated random numbers to match the range of F_{avail}, thus maintaining a uniform probability distribution over the entire frequency set. The mapping is performed by selecting a frequency f_o from F_{avail} in a manner that corresponds to each random number R_o, ensuring uniform randomness across the set. This process can be described as follows:

$$f_o = F_{\text{avail}}\left(\frac{R_o}{R_{\max}} \times |F_{\text{avail}}|\right) \tag{11}$$

where:

- R_o is the raw random number generated by the RPU,
- $R_{\max}$ is the maximum possible value of R_o,
- $|F_{\text{avail}}|$ is the cardinality (size) of the available frequency set F_{avail},
- and f_o is the selected frequency within F_{avail}.

This approach scales each random number R_o to a valid index within F_{avail}, effectively normalizing the random output for uniform selection across the updated frequency set. This normalization ensures the distribution of frequency selections remains unbiased and uniformly spread over F_{avail}.

Algorithm 3 Interference Detection and Frequency Exclusion

```
1:  while Communication is Active do
2:      for all Frequencies $f \in F$ do
3:          Measure interference level $I(f)$
4:          if $I(f) > I_{\text{threshold}}$ then
5:              Add $f$ to $F_{\text{excl}}$
6:          else
7:              Remove $f$ from $F_{\text{excl}}$ if present
8:          end if
9:      end for
10:     Update available frequencies $F_{\text{avail}} \leftarrow F \setminus F_{\text{excl}}$
11:     Inform RPU of updated $F_{\text{avail}}$
12: end while
```

4.5. Enhanced Security Analysis

The integration of true random frequency hopping introduces several security benefits. We analyze potential attack scenarios and discuss mitigation strategies.

4.5.1. Attack Scenarios and Mitigations

- Brute-Force Attacks: An adversary attempts to scan all possible frequencies to intercept or jam the communication.
  Mitigation: The large frequency band and rapid hopping rate make this impractical due to the required resources and time.
- Replay Attacks: Captured signals are replayed to disrupt communication or trick the receiver into accepting false data.
  Mitigation: Implementing time-stamped authentication tokens prevents the acceptance of outdated or replayed signals.
- Synchronization Attacks: In the context of our system, synchronization between the transmitter and receiver is crucial due to the use of true random frequency hopping sequences. An adversary may attempt to desynchronize the communication by introducing delays, jamming synchronization signals, or injecting false synchronization messages. Desynchronization can lead to loss of communication and reduced system reliability.
  Mitigation: To counter synchronization attacks, we could employ robust synchronization protocols that include the following features:
  - Error Correction Codes: Utilizing error correction codes (ECC) in synchronization messages allows the receiver to detect and correct errors introduced by

interference or malicious activities, ensuring that synchronization data is accurately received.

- Secure Authentication: Synchronization messages can be secured using cryptographic techniques such as digital signatures or message authentication codes (MACs). This prevents attackers from injecting false synchronization signals or modifying legitimate ones.
- Periodic Resynchronization: The system could perform periodic resynchronization at predefined intervals or upon detecting anomalies in communication. This helps to realign the transmitter and receiver in case of minor desynchronization and reduces the window of opportunity for attackers.
- Redundant Synchronization Channels: Employing multiple synchronization channels or methods (e.g., combining time-based synchronization with signal-based cues) could add redundancy, making it more difficult for an attacker to disrupt all synchronization paths simultaneously.
- Monitoring and Anomaly Detection: Continuously monitoring communication metrics (e.g., synchronization error rates, unexpected delays) could allow the system to detect potential synchronization attacks early and initiate countermeasures, such as increasing the hopping rate or switching to a secure fallback mode.

4.5.2. Information-Theoretic Security

The use of true random sequences ensures that the mutual information between the transmitted signal and any intercepted signal by an adversary is minimized. The entropy H of the frequency sequence is maximized, making the system secure against attackers with unlimited computational resources.

$$I(\text{Transmitted Signal}; \text{Intercepted Signal}) = 0 \tag{12}$$

This implies that without knowledge of the true random sequence or successful synchronization, an adversary gains no useful information from intercepted signals.

4.5.3. Latency and Throughput Models

In communication systems, latency L refers to the time delay experienced by data as it travels through the system. It can be expressed as:

$$L = L_{\text{prop}} + L_{\text{proc}} + L_{\text{sync}} + L_{\text{queue}} \tag{13}$$

where:

- L_{prop} is the propagation delay,
- L_{proc} is the processing delay,
- L_{sync} is the synchronization delay,
- L_{queue} is the queuing delay.

While latency is an important parameter affecting the system's responsiveness, throughput T is a measure of how much data can be successfully transmitted over the communication channel per unit time. It is influenced by factors such as bandwidth, modulation schemes, coding rates, and error rates.

The throughput can be modeled as:

$$T = R \times (1 - P_e) \tag{14}$$

where:

- R is the raw data rate (bits per second),
- P_e is the bit error rate (BER).

The raw data rate R depends on the bandwidth B and the spectral efficiency η (measured in bits per second per Hertz):

$$R = B \times \eta \tag{15}$$

In the context of our FHSS system with true random frequency hopping, the hopping rate R_h can impact the effective data rate due to factors such as the time required for frequency switching and synchronization overhead. The effective data rate R_{eff} can be adjusted to account for these factors:

$$R_{\text{eff}} = R \times \left(1 - \frac{t_{\text{switch}} + t_{\text{sync}}}{T_{\text{hop}}}\right) \tag{16}$$

where:

- t_{switch} is the time taken to switch frequencies,
- t_{sync} is the time spent on synchronization during each hop,
- T_{hop} is the duration of each hop (i.e., $T_{\text{hop}} = \frac{1}{R_h}$).

The throughput then becomes:

$$T = R_{\text{eff}} \times (1 - P_e) \tag{17}$$

This formulation acknowledges that while latency affects the timing of data delivery, throughput is primarily determined by the data rate and the error performance of the system. High latency does not necessarily reduce the throughput but can impact the overall system performance in terms of delay-sensitive applications.

It is important to understand that in practical FHSS systems, certain physical and protocol-level factors can introduce delays during frequency hopping.

Firstly, although the transmitter and receiver are synchronized in terms of timing and frequency hopping sequence, the actual process of switching frequencies involves hardware components that require a finite amount of time to change from one frequency to another. This time is known as the frequency switching time t_{switch}, and it can vary depending on the specific hardware used in the system. During this switching period, the transmitter and receiver may not be able to send or receive data, resulting in a brief interruption in communication.

Secondly, maintaining synchronization often involves the exchange of synchronization signals or the execution of synchronization algorithms, which consume processing time t_{sync}. Even with precise synchronization, periodic adjustments and computations are necessary to compensate for clock drift and other factors, introducing processing delays.

Therefore, the total effective time available for data transmission during each hop is reduced by the sum of t_{switch} and t_{sync}. This reduction in available transmission time per hop effectively decreases the data rate, which is why we adjust the effective data rate R_{eff} to account for these factors in the throughput model.

Including t_{switch} and t_{sync} in the calculations allows for a more accurate representation of the system's performance, especially when considering high hopping rates R_h. As R_h increases (i.e., the duration of each hop T_{hop} decreases), the proportion of time spent on frequency switching and synchronization relative to the total hop duration becomes more significant, potentially impacting the overall throughput.

In practical implementations, the latency and throughput models are primarily utilized during the design, configuration, and optimization phases of the FHSS system. These models help in predicting system performance under various operating conditions and in making informed decisions about parameter settings such as hopping rate R_h, bandwidth B, and modulation schemes.

During active communication, the system may not need to run these calculations continuously. Instead, they can be computed periodically or when there is a significant change in system parameters or operating conditions. For example, if the system detects a change in the bit error rate P_e due to interference or other environmental factors, it may recalculate the effective data rate R_{eff} and throughput T to assess the impact on performance. This recalculation can trigger adjustments to system parameters to maintain optimal operation.

The frequency of running these calculations depends on the specific requirements and constraints of the system. In scenarios where the communication environment is stable, the models may be recalculated infrequently or only when manual adjustments are made. In contrast, in dynamic or high-security environments where conditions can change rapidly, the system might perform these calculations more frequently—potentially after every significant event or at regular intervals defined by a monitoring frequency f_{calc}.

It is important to note that the calculations themselves are not computationally intensive and can be executed quickly without imposing significant overhead on the system. Modern communication systems often include real-time monitoring and control software capable of performing these calculations efficiently. By balancing the frequency of these calculations with system performance requirements, the FHSS system can maintain high data throughput and low latency while adapting to changing conditions.

In our current setup, we did not implement the models for latency in the operational system. The focus of our implementation was on demonstrating the feasibility of integrating true random number generation into the FHSS mechanism and analyzing its impact on security and randomness properties. The latency and throughput considerations are included for comprehensiveness and to guide future work, where these factors could be empirically measured and optimized in real-world scenarios.

4.5.4. Error Rate Analysis

The bit error rate (BER) P_b in a frequency hopping system is given by:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{18}$$

where E_b is the energy per bit, N_0 is the noise power spectral density, and $Q(\cdot)$ is the Q-function.

Frequency hopping spreads the signal over multiple frequencies, effectively reducing N_0 and improving P_b.

5. Implementation

To implement true random frequency hopping in communication systems, several critical components and processes must be established. This section describes the practical implementation, including the setup of the Randomness Processing Unit (RPU), the generation and synchronization of frequency hopping sequences, and the security protocols necessary to maintain the integrity and confidentiality of the communication.

5.1. Setup of the Randomness Processing Unit (RPU)

The RPU is a specialized hardware device that generates true random numbers. It typically leverages physical phenomena such as quantum noise, thermal noise, or radioactive decay, which are inherently unpredictable. The RPU's design includes:

- Noise Source: The physical process generating randomness.
- Entropy Extraction: Digital circuitry or algorithms to convert raw physical signals into uniformly distributed random numbers.
- Post-Processing: Additional steps to ensure the uniformity and independence of the generated numbers, such as whitening algorithms or hash functions.

Before deployment, the RPU must be rigorously tested to ensure the quality and true randomness of the output. Statistical tests, such as the National Institute of Standards and Technology (NIST) suite, are used to assess randomness. Key metrics include:

- Uniformity: Distribution of numbers should be uniform across the possible range.
- Independence: No correlation should exist between consecutive numbers.
- Entropy: High entropy indicates that the output is as unpredictable as possible.

5.2. Generation and Synchronization of Frequency Hopping Sequences

Using the RPU, true random numbers R_o are generated and mapped to frequency values f_o within a predefined band. The frequency for the o-th time interval is given by:

$$f_o = f_{\min} + (f_{\max} - f_{\min}) \frac{R_o}{R_{\max}}$$

where $R_{\max}$ is the maximum possible output of the RPU. This mapping ensures that the frequencies are chosen randomly and uniformly over the band.

A critical challenge in FHSS systems, especially when using true random frequency hopping sequences, is maintaining synchronization between the transmitter and receiver. Unlike traditional FHSS systems that use pseudo-random number generators (PRNGs) with shared seeds, our system relies on sequences that are inherently unpredictable and cannot be reproduced by generating devices independently. Therefore, both the transmitter and receiver must have access to the same sequence of random numbers or be able to generate them in a synchronized manner.

To achieve secure and reliable synchronization in our system, we implement a two-stage process involving public-key cryptography and symmetric key encryption:

- Initial Synchronization and Key Agreement: At the start of communication, the transmitter and receiver engage in a secure key exchange protocol using public-key cryptography, such as RSA or Elliptic Curve Cryptography (ECC). This allows them to securely agree upon a symmetric key without exposing it to potential eavesdroppers. The symmetric key established during this phase is used for encrypting subsequent communications, including the true random sequence and synchronization information.
- Secure Distribution of Random Sequences: With the symmetric key in place, the transmitter securely transmits the necessary synchronization information and the initial segment of the true random frequency hopping sequence to the receiver. This information is encrypted using the symmetric key, ensuring that only authorized parties with the corresponding key can decrypt and access the sequence. The use of a symmetric key for encryption at this stage is crucial because symmetric algorithms like Advanced Encryption Standard (AES) are more efficient for handling larger amounts of data in real-time communication compared to public-key algorithms.

Once the initial synchronization is achieved, both parties use the shared true random sequence for frequency hopping. To maintain synchronization over time, the system incorporates:

- Periodic Resynchronization: Due to potential clock drift or communication delays, periodic resynchronization is necessary. The transmitter periodically sends encrypted synchronization beacons or timestamps using the established symmetric key. This allows the receiver to adjust its timing and ensures continuous alignment with the transmitter's hopping sequence.
- Dynamic Key Updating: To enhance security, the symmetric key used for encryption can be updated periodically using key derivation functions or additional key exchange protocols. This reduces the risk of key compromise over extended periods and mitigates potential replay or interception attacks.
- Robust Error Handling: The synchronization protocol includes mechanisms for error detection and correction to handle data loss or corruption due to channel impairments or malicious interference. This ensures that synchronization integrity is maintained even in adverse conditions.

That is, to achieve secure and reliable synchronization in our system, we implement a two-stage process involving public-key cryptography and symmetric key encryption. Initially, the transmitter and receiver engage in a secure key exchange protocol using public-key cryptography to establish a symmetric key without exposing it to potential eavesdroppers. With the symmetric key in place, the transmitter securely transmits the necessary synchronization information and the initial segment of the true random frequency hopping sequence to the receiver using symmetric encryption. Once the initial synchronization is achieved, both parties use the shared true random sequence for frequency hopping. To maintain synchronization over time, the system incorporates periodic resynchronization, dynamic key updating, and robust error handling mechanisms, ensuring continuous alignment and enhanced security.

By integrating public-key cryptography for initial key agreement and symmetric key encryption for ongoing secure communication, our system effectively combines the strengths of both cryptographic approaches. Public-key cryptography facilitates secure key exchange without prior shared secrets, while symmetric key encryption offers efficiency for real-time data protection.

It is important to note that while the use of public-key cryptography to establish a symmetric key is a standard practice, the unique aspect of our system lies in the secure distribution and synchronization of true random frequency hopping sequences. Unlike PRNG-based systems where the receiver can independently generate the same sequence using a shared seed, our approach requires careful management of the true random sequences. The encrypted transmission of these sequences ensures that they remain confidential and that only authorized receivers who possess the correct symmetric key can participate in the communication. This adds a layer of security, making it significantly more challenging for adversaries to predict or intercept the frequency hopping pattern.

5.3. Implementation Considerations and Optimization

The choice of the frequency band and the allocation strategy depends on regulatory requirements and the specific application. The system must ensure minimal interference with other communications, and comply with local spectrum regulations.

The efficiency of the system is influenced by the power requirements of the RPU and the overall system. Optimizing the power usage while maintaining high security is crucial, especially in portable or battery-operated devices.

Real-time generation and processing of random numbers and frequency hopping require efficient algorithms and hardware. The system must ensure low latency to avoid delays in communication.

5.4. Security and Performance Evaluation

The system's security is evaluated based on its resistance to various types of attacks:

- Eavesdropping: The unpredictability of true random sequences makes it extremely difficult for unauthorized parties to intercept and understand the communication.
- Jamming: True random frequency hopping enhances resistance to jamming, as the attacker cannot predict the frequency changes.

Performance is assessed in terms of:

- Throughput: The rate at which data can be successfully transmitted.
- Latency: The delay introduced by frequency hopping and synchronization processes.
- Reliability: The system's ability to maintain communication integrity under different conditions, including adversarial attacks.

6. Application to a Drone Cloud with Ring Topology

In the context of a drone cloud, a group of drones intercommunicates using a network configuration known as a ring topology, where each drone communicates directly with its two neighbors. This setup is particularly advantageous for coordinating movements, sharing sensor data, and maintaining a cohesive network structure. The application of true random frequency hopping in this environment can significantly enhance the security and reliability of communication, especially in scenarios where the drone cloud operates in hostile or adversarial conditions.

The communication infrastructure of the drone cloud is represented in Figure 3, showing how drones are interconnected in a ring topology and utilize FHSS with true random frequency hopping. The ring topology has been shown to improve communication efficiency and fault tolerance in networked systems [30], which is particularly beneficial for our drone cloud configuration.

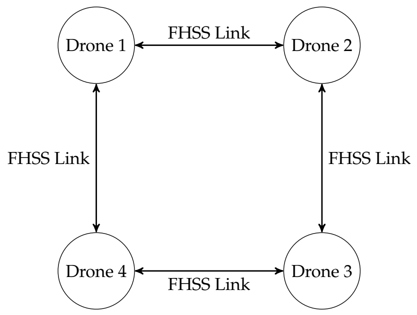

Figure 3. Diagram of the drone cloud with ring topology implementing FHSS with true random frequency hopping.

6.1. Communication Infrastructure

In a ring topology, each drone has two communication links, one to the preceding drone and another to the following one. This setup ensures redundancy and can facilitate continuous data flow even if one link fails. The direct data links between drones make efficient use of the available bandwidth and minimize latency.

Each data link operates independently with its own frequency hopping sequence. This configuration helps in avoiding interference between links, and improves the overall resilience of the network to jamming and interception attempts.

6.2. Frequency Hopping Implementation

Each drone is equipped with an RPU to generate true random frequency hopping sequences. Given the ring topology, the sequence for each link can be independently generated and is specific to the communicating pair of drones. The use of true random numbers ensures that even if an adversary gains access to one link, they cannot predict the hopping pattern of other links.

The true random numbers R_o generated by the RPUs are mapped to frequency values f_o within the communication band allocated for the drone cloud. The frequency for the o-th interval is defined as: $f_o = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \frac{R_o}{R_{\text{max}}}$, where f_{min} and f_{max} are the minimum and maximum frequencies available, and R_{max} is the maximum output of the RPU.

6.3. Synchronization and Key Management

To establish communication, drones must initially synchronize their RPUs to ensure that the frequency hopping sequences are correctly aligned. This synchronization can be achieved using GPS time stamps or a dedicated synchronization signal.

Each drone pair must securely exchange an initial key or synchronization data to coordinate the start of the frequency hopping sequence. Public-key cryptography can be used for this purpose, providing a secure means of distributing the necessary information.

Given the potential for clock drift or other synchronization issues, drones periodically resynchronize their frequency hopping sequences. This can be achieved through a resynchronization protocol, possibly using reserved channels or a periodic synchronization beacon.

While it is true that an initial secure and reliable channel is used for exchanging keys and synchronization data between drones, the primary purpose of this channel is to establish the parameters necessary for secure communication. However, relying solely on this channel for all communication would expose the system to vulnerabilities associated with fixed-frequency transmissions, such as jamming, interception, and eavesdropping.

The use of Frequency Hopping Spread Spectrum (FHSS) for the main communication channel offers several critical advantages:

- Resistance to Jamming: FHSS systems are inherently resistant to jamming attacks because the frequency changes rapidly in a pattern known only to the communicating parties. An adversary attempting to jam the communication would need to jam all possible frequencies simultaneously, which is often impractical.
- Enhanced Security: By hopping frequencies in a true random sequence, the communication becomes extremely difficult to predict or intercept without knowledge of the hopping pattern. This significantly reduces the risk of eavesdropping and unauthorized access.
- Interference Mitigation: FHSS helps in mitigating interference from other devices operating in the same frequency band. Since the communication frequency changes rapidly, the impact of any interference on a particular frequency is minimized.
- Robustness in Adverse Environments: Drones often operate in environments where the communication channel conditions can vary rapidly due to movement, obstacles, or other factors. FHSS provides robustness against such variations, maintaining reliable communication.
- Scalability: In a network of multiple drones, FHSS allows for better spectrum utilization and reduces the likelihood of collisions, as different pairs can use different hopping patterns.

The initial secure channel is typically a narrowband communication link used specifically for key exchange and synchronization due to its simplicity and lower resource requirements. It is not designed to handle the bandwidth and security requirements of continuous drone communication, especially in hostile or dynamic environments. By transitioning to an FHSS-based communication channel after the initial setup, the system leverages the benefits of spread spectrum techniques to enhance overall communication security and reliability.

Therefore, while the initial secure channel is essential for setting up communication parameters, the use of FHSS for ongoing communication is critical for protecting against threats and ensuring the robustness of the drone network in real-world operating conditions.

6.4. Complexity Analysis of Proposed Algorithm

The complexity of the proposed QRNG-based Frequency Hopping Spread Spectrum (FHSS) algorithm is analyzed in terms of the random number generation, frequency hopping, and synchronization.

- Random Number Generation: The QRNG generates a random number R_o per hop, where each generation is effectively $O(1)$ due to dedicated QRNG hardware.
- Frequency Selection: For each frequency hop, a lookup is performed to select f_o from F_{avail}. Assuming the set F_{avail} has size $|F|$, selecting a frequency f_o involves a mapping operation, making the complexity $O(\log |F|)$ for each hop due to binary search or direct indexing in practice.
- Synchronization and Resynchronization: Each drone periodically synchronizes its hopping sequence with the other drones in the network. Assuming N drones and each drone communicates with two neighboring drones in a ring topology, the synchronization process is $O(N)$.

Given the above, the total complexity for frequency hopping across T time intervals in a network of N drones is approximately:

$$O(T \cdot (1 + \log|F|) + N) = O(T \log|F| + N)$$

This represents a relatively low complexity, making the QRNG-based FHSS efficient for real-time operation even in dense drone networks.

7. Evaluations

The primary objective of our experimentation was to evaluate the effectiveness and security of using true random frequency hopping in a simulated drone network with a ring topology. The setup included the following key components:

- QRNG Device or RPU: A Quside QRNG device [9,31] installed in the internal network of the BARCELONA Supercomputing Centre Facility was utilized to generate true random numbers, which were used to create the frequency hopping sequences. The device is a QRNG Module ETH based on the FMC 400 Randomness Module by Quside.
- Drones: Each drone in the network was a simulated entity capable of implementing frequency hopping based on the generated sequences.
- Secure Communication Protocol: A secure communication protocol was developed, incorporating AES encryption, Hash-based message authentication code (HMAC) for data integrity, and secure key exchange mechanisms.
- Synchronization Mechanism: Synchronization between drones was achieved using a preset time, that in practice would correspond to GPS, ensuring a common start time for frequency hopping across all nodes.

The QRNG Module ETH, see Figure 4, is based on the FMC 400 Randomness Module by Quside. It includes an FPGA baseboard to provide high-quality and fast random bit rates through a 1Gb-Ethernet connection.

The QRNG Module ETH employs a digital quantum entropy source based on the accelerated phase diffusion process in semiconductor lasers. The system comprises:

- Laser: Produces a stream of phase-randomized optical pulses.
- Unbalanced MACH-ZEHNDER Interferometer: Converts random phases into random amplitudes.
- Photodetector: Translates the optical signal into an electrical signal.
- Digitizer: Converts the analog signal into a digital bit stream.
- Control and Clocks: Manages the digital control, health monitoring, and randomness extraction.

Figure 4. Reference front panel for the QRNG Module ETH powered by the FMC 400 quantum entropy source by Quside.

- RJ-45 Connection (Front Panel): Connects the QRNG Module ETH to the PC for control, monitoring, and data transfer.
- ON/OFF Switch (Front Panel): Powers the device on or off. The system restarts with default settings when switched on.
- Power Plug (Rear Panel): Connects to a 110/240 VAC power source.
- Status LED (Front Panel): Indicates the system status.

This device leverages the inherent quantum phase noise of lasers to generate true random numbers, providing a raw bit rate of up to 1 Gbit/s with strong entropy characteristics. The phase diffusion technique involves modulating a semiconductor laser to produce a stream of phase-randomized optical pulses, which are then converted into amplitude fluctuations using an interferometer [32–34]. These fluctuations are detected and digitized to produce high-quality random bit streams suitable for cryptographic applications and secure communications.

7.1. Procedure

The experimentation procedure consisted of several stages:

The QRNG device generated true random numbers, which were converted into a series of frequency values within a specified range. This sequence determined the order of frequency changes for each communication link in the drone network.

To establish a secure communication channel, a shared key was exchanged between the drones using a secure key exchange protocol. This key was used for both encrypting the frequency sequence and generating HMACs for integrity verification.

All drones synchronized their internal clocks to a common preset time, which in practice would correspond to a GPS-based time reference. An agreed-upon start time was set, and all drones commenced frequency hopping simultaneously, adhering to the pre-generated sequence.

During the simulation, each drone entity transmitted data packets over the network. Here each drone is an instantation entity of its correspondent class in python. The data, including the frequency sequence and synchronization information, was encrypted using AES. HMACs were generated and attached to each packet to ensure data integrity.

The experiment's performance was monitored in real-time, where the code in python simulated a drone ring topology, and the QRNG was accessed through a lock mechanism in the internal VPN of the BARCELONA Supercomputing Center. We focused on synchronization accuracy, data integrity, and resistance to potential interception or jamming. The frequency hopping sequence and synchronization data were logged for subsequent analysis.

7.2. Results and Observations

Figure 5 presents a comprehensive overview of our proposed methodology for enhancing the security and reliability of multi-drone communications using QRNG-based FHSS systems. At the core of our approach is a QRNG device, which generates truly random numbers to drive the frequency hopping sequences. These random numbers are efficiently distributed to multiple drones through a shared memory system, ensuring that each drone has access to high-quality random data for its FHSS operations. To manage concurrent access to the QRNG device, we implement a specialized protocol that optimizes resource utilization while maintaining the integrity of the random number generation process. This architecture allows each drone to implement its own FHSS system using the shared random numbers, resulting in a network of secure, jam-resistant communication channels. The wireless medium, represented by the cloud in the diagram, emphasizes the challenging environment in which these secure communications take place. By integrating quantum randomness into a multi-drone FHSS framework, our methodology significantly enhances the system's resistance to jamming, interception, and other forms of interference, thereby improving the overall security and reliability of drone communications in complex operational scenarios.

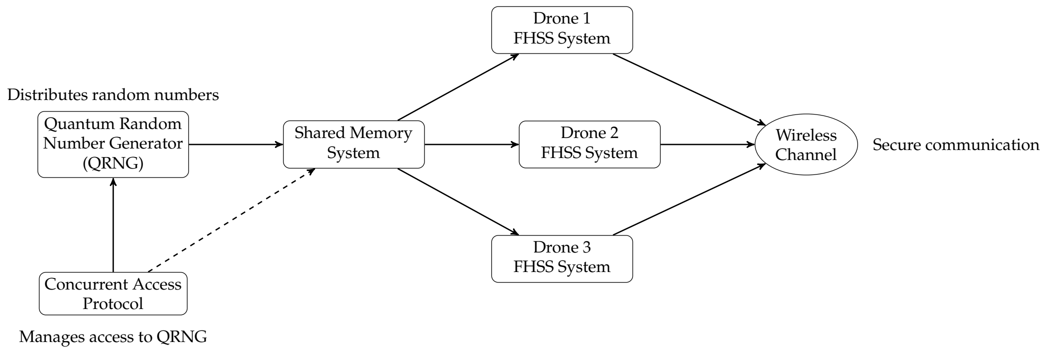

Figure 5. Diagram illustrating the methodology of using a QRNG for true random frequency hopping in a multi-drone FHSS system. The QRNG generates truly random numbers, which are distributed to multiple drones through a shared memory system. A concurrent access protocol manages access to the QRNG, ensuring efficient use of the device. Each drone implements its own FHSS system using the shared random numbers, enhancing anti-jamming capabilities and resistance to eavesdropping across the entire network.

The experimental results, as illustrated in Figures 6–17, provide compelling evidence of the effectiveness of our QRNG-based frequency hopping system across multiple drones. The frequency distribution plots (Figures 7, 11 and 15) demonstrate a relatively uniform spread of frequencies for all three drones, except for the presence of jamming at 5500 Hz, indicating successful implementation of the quantum random number generation. The frequency sequence plots (Figures 8, 12 and 16) show no discernible patterns, which is crucial for maintaining unpredictability in the hopping sequence. Notably, the autocorrelation plots (Figures 6, 10 and 14) exhibit sharp peaks at zero lag and linear decay to near-zero values for all other lags, confirming a general absence of periodicity or predictability in the sequences, albeit some short-term correlation; which indicates that the RPU technology has still room for improvement in terms of quality and stability of the generated random numbers. The spectral analysis plots (Figures 9, 13 and 17) further corroborate these findings, showing a relatively flat power spectral density across frequencies, which is characteristic of truly random sequences. These results collectively demonstrate that our QRNG-based system successfully generates unique, largely unpredictable, and uniformly distributed frequency hopping sequences for each drone, thereby enhancing the security and reliability of the communication system against potential jamming or interception attempts. That being said, while QRNG technology is still at its early stage of development, it shows promising performance in the future advancement of secure communication scenarios.

An important aspect of our experimental setup was the selection of parameters to thoroughly evaluate the performance and resilience of our QRNG-based FHSS system. Each drone generated a sequence of 10,000 frequency hops (num_hops = 10,000), operating within a frequency range from 1000 Hz to 10,000 Hz (freq_min = 1000 Hz, freq_max = 10,000 Hz). The duration of each frequency hop was fixed at 0.01 s (duration = 0.01 s), resulting in a total communication time of 100 s per drone. To ensure synchronized operations, all drones were programmed to commence their frequency hopping sequences simultaneously, using a synchronization time set 10 s ahead of the current system time (sync_time = current time + 10 s), as a proxy for GPS time.

The jamming simulation was implemented by introducing interference at a central frequency of 5500 Hz. Frequencies within a 200 Hz range of this jamming frequency were adjusted by adding a random offset between −500 Hz and 500 Hz, effectively simulating an active jammer's impact on the communication system. These parameters were chosen to provide a test of the system's anti-jamming capabilities, ensuring that the FHSS sequences generated by the drones could adapt and maintain secure communications even in the presence of targeted interference.

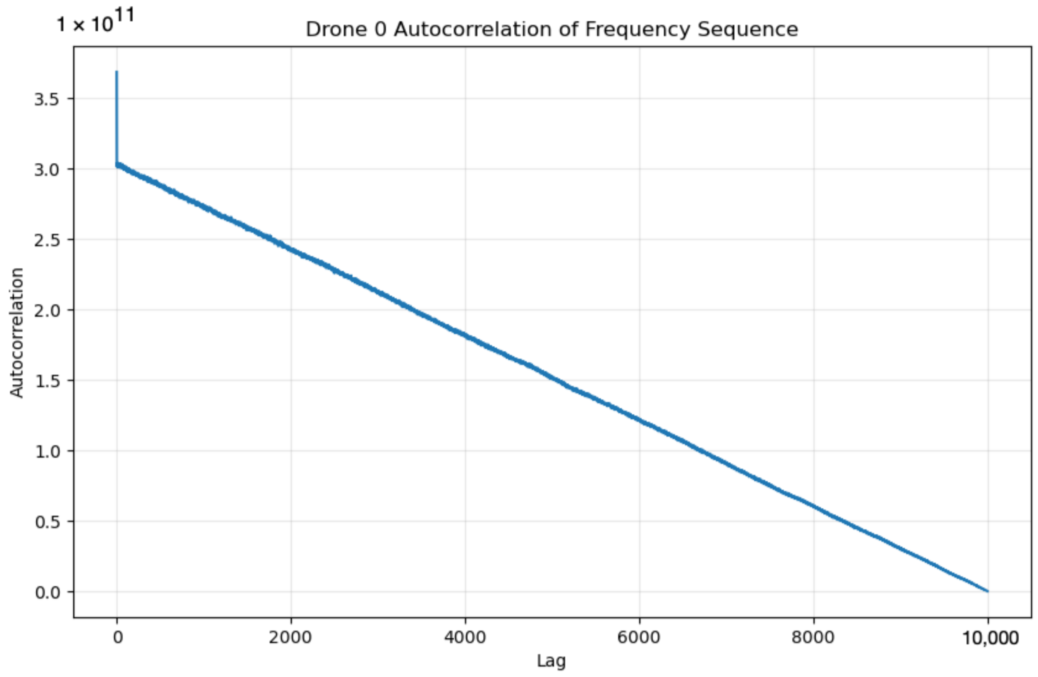

Figure 6. Drone 0: Autocorrelation.

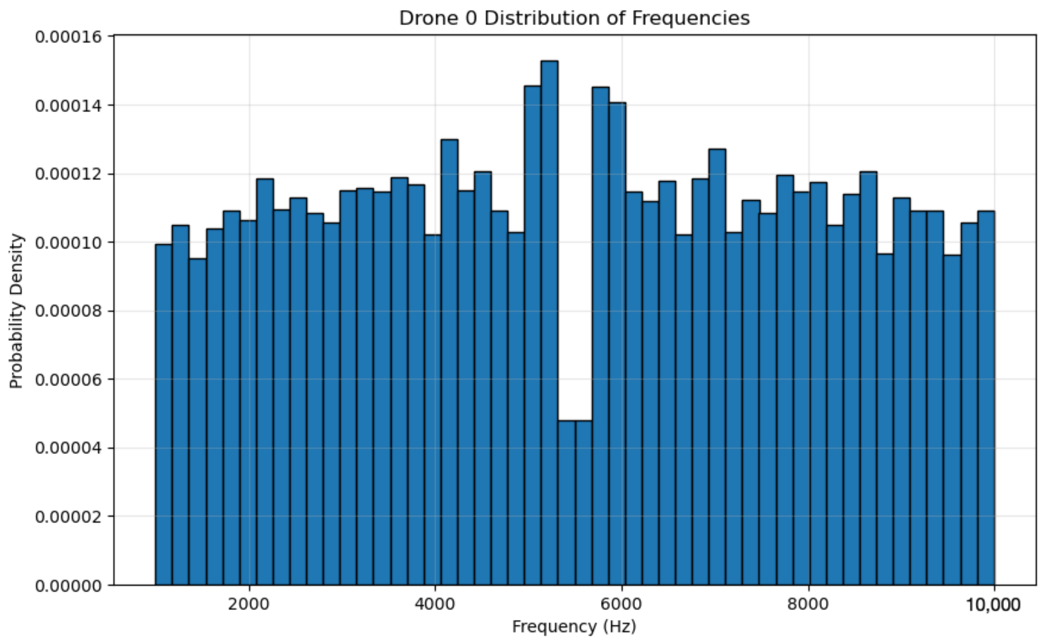

Figure 7. Drone 0: Frequency Distribution, where a noticeable dip around 5500 Hz indicates a reduced frequency count in the jammed region.

A particular aspect of our implementation that warrants clarification is the method by which random numbers from the QRNG were transformed into frequency hopping sequences. We collected random bytes from the QRNG and assembled them into 32-bit unsigned integers to maximize the range and uniformity of the random values. Each integer was then normalized by dividing by 2^{32} and scaled to fit within our desired frequency range. This careful mapping was crucial to ensure a truly uniform distribution of frequencies, eliminating any biases that could arise from improper normalization or scaling.

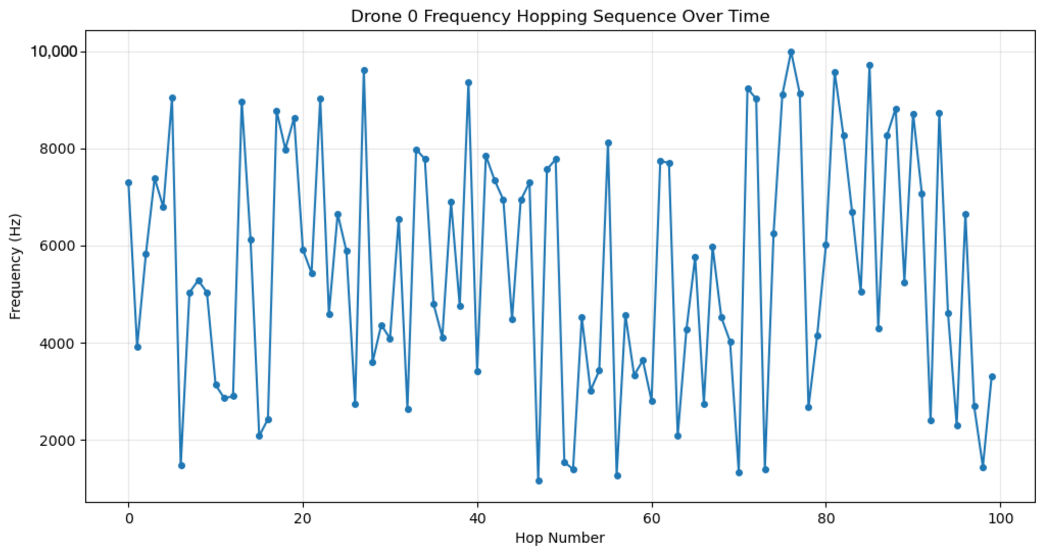

Figure 8. Drone 0: Frequency Sequence over 100 hops of the total 100,000 hops.

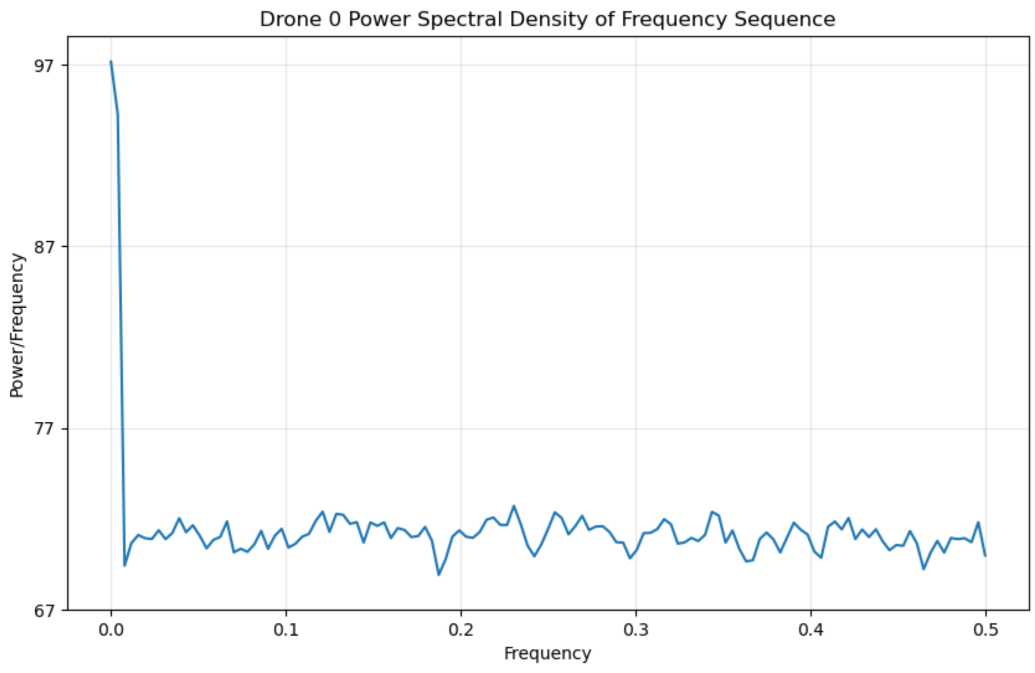

Figure 9. Drone 0: Spectral Analysis.

The experimentation yielded several key insights:

- Data Security: The AES encryption and HMAC verification successfully protected the frequency sequence and synchronization data from interception and tampering. No unauthorized access or data modification was detected during the experiment.
- System Resilience: The implementation of true random frequency hopping demonstrated a potential high degree of resilience to jamming, given by the spectral analysis of the frequency hopping sequences. The unpredictability of the sequence made it difficult for potential adversaries to disrupt communication.

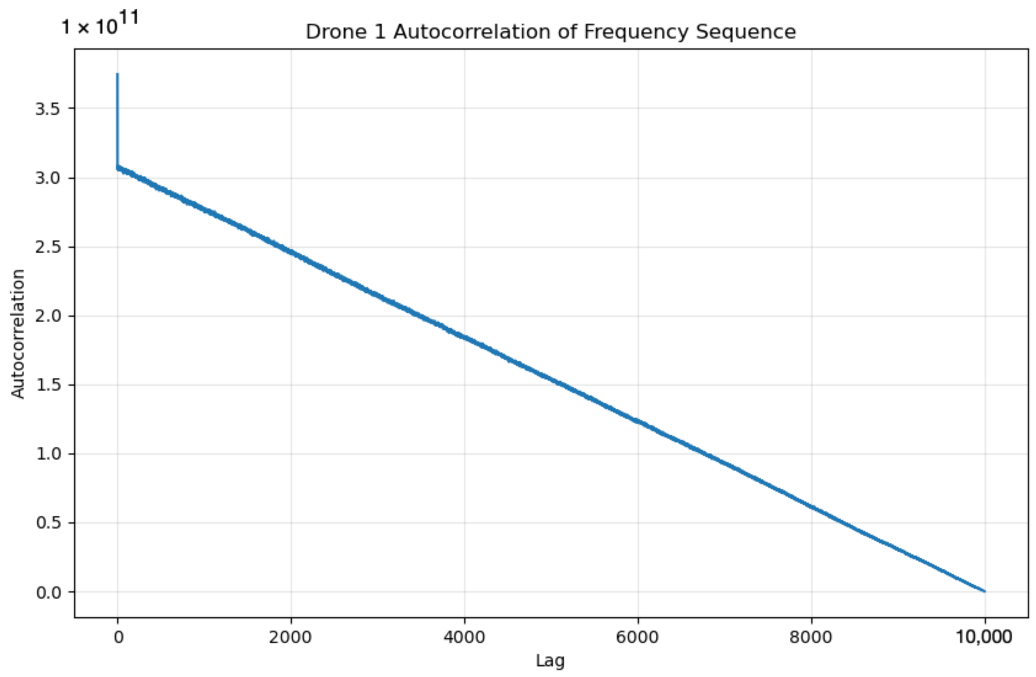

Figure 10. Drone 1: Autocorrelation.

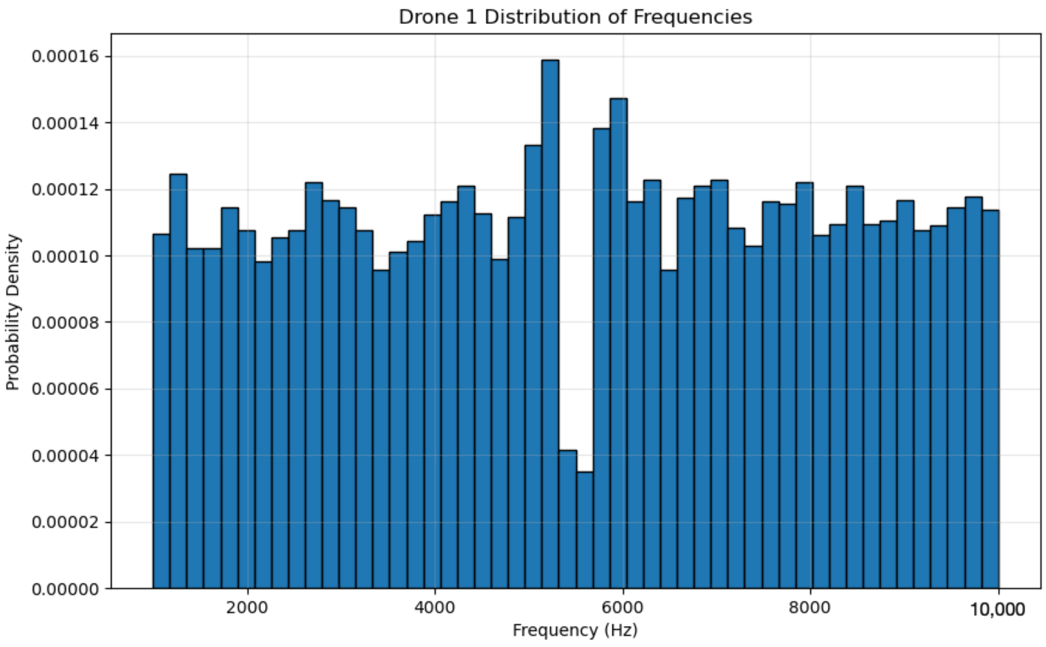

Figure 11. Drone 1: Frequency Distribution, where a noticeable dip around 5500 Hz indicates a reduced frequency count in the jammed region.

Although this is a simplified setup for illustration purposes, and given that QRNG are still at their early stage of development and deployment, to address the practical implementation of distributing true random numbers to drones that are airborne, we could adapt the concept of a shared memory system to a distributed model suitable for a flying multi-drone environment. Instead of relying on a physical shared memory, which is not feasible for drones in flight, we would need to implement a virtual shared memory system facilitated through secure wireless communication channels. The QRNG device would be stationed at a central ground control unit or a leading drone acting as a coordinator. This QRNG would generate true random numbers, which are then securely transmitted to each drone in the network using encrypted communication links prior to or during the mission. Each drone maintains a local buffer to store the received random numbers for its FHSS operations. To manage the distribution and synchronization of the random numbers, we would need to implement a concurrent access protocol as a network protocol that ensures all drones receive the necessary random data in a timely and secure manner. This protocol handles requests for random numbers, manages the transmission schedules to avoid collisions, and verifies the integrity of the received data using cryptographic techniques such as digital signatures or message authentication codes (MACs). By virtualizing the shared memory over the network, we enable each drone to have access to high-quality random numbers without the need for a physical shared memory. This approach makes our methodology practical and implementable for drones that are flying, allowing each drone to independently implement its own FHSS system while maintaining synchronization with the rest of the network. The use of secure communication channels and robust synchronization protocols ensures that the system remains resistant to jamming, interception, and other forms of interference, thereby enhancing the overall security and reliability of drone communications in complex operational scenarios.

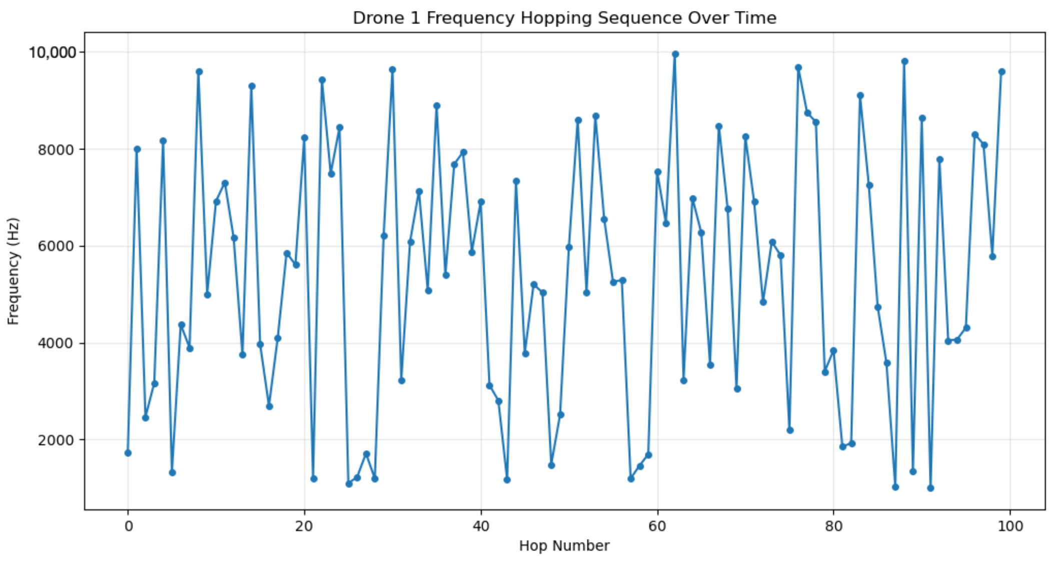

Figure 12. Drone 1: Frequency Sequence over 100 hops of the total 100,000 hops.

Additionally, we conducted Chi-squared and Kolmogorov-Smirnov (KS) tests on the generated frequency sequences to assess the randomness and distribution uniformity across the frequency spectrum. The test results for each drone are shown in Table 1.

Table 1. Qualitative Summary of Performance and Statistical Analysis for Each Drone.

Drone	Statistical Analysis	Performance Outcome	Key Observations
Drone 0	Chi-squared: 263.59, $p < 10^{-30}$ KS: 0.01704, $p = 0.0053$	Reliable communication under interference at 5500 Hz	Dynamic response to frequency jamming; high resilience in message integrity
Drone 1	Chi-squared: 240.11, $p < 10^{-27}$ KS: 0.01572, $p = 0.0107$	Stable communication with reduced selection of jammed frequencies	Effective avoidance of jammed frequency maintaining data integrity
Drone 2	Chi-squared: 233.88, $p < 10^{-25}$ KS: 0.01679, $p = 0.0070$	Consistent resilience to interference, with effective packet integrity verification	Demonstrates robustness and adaptive capabilities in hostile environments

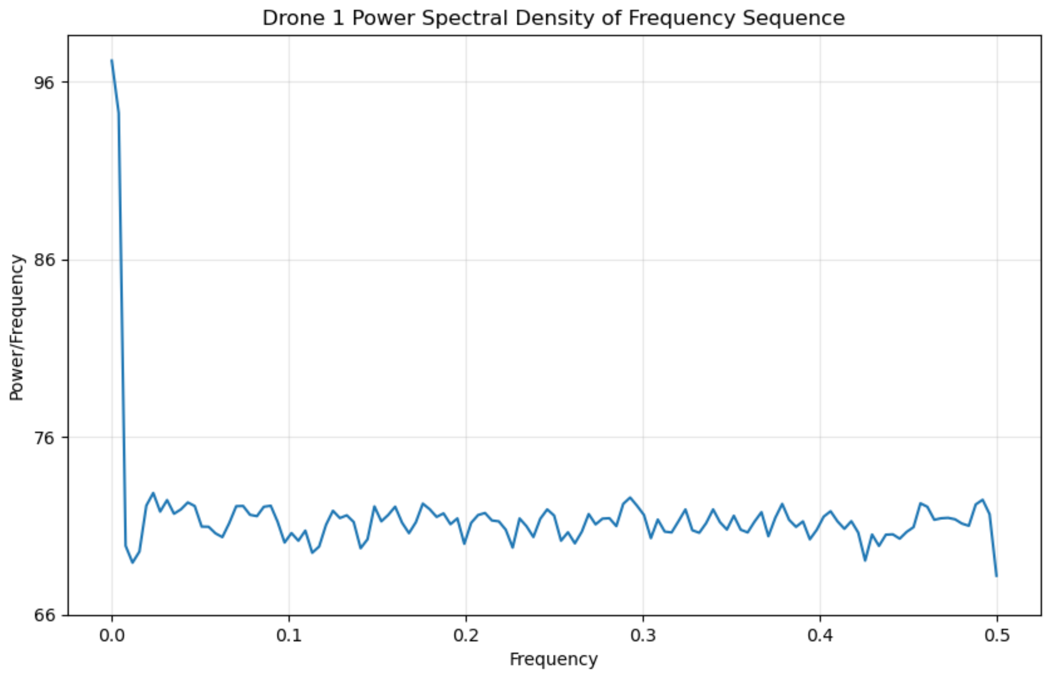

Figure 13. Drone 1: Spectral Analysis.

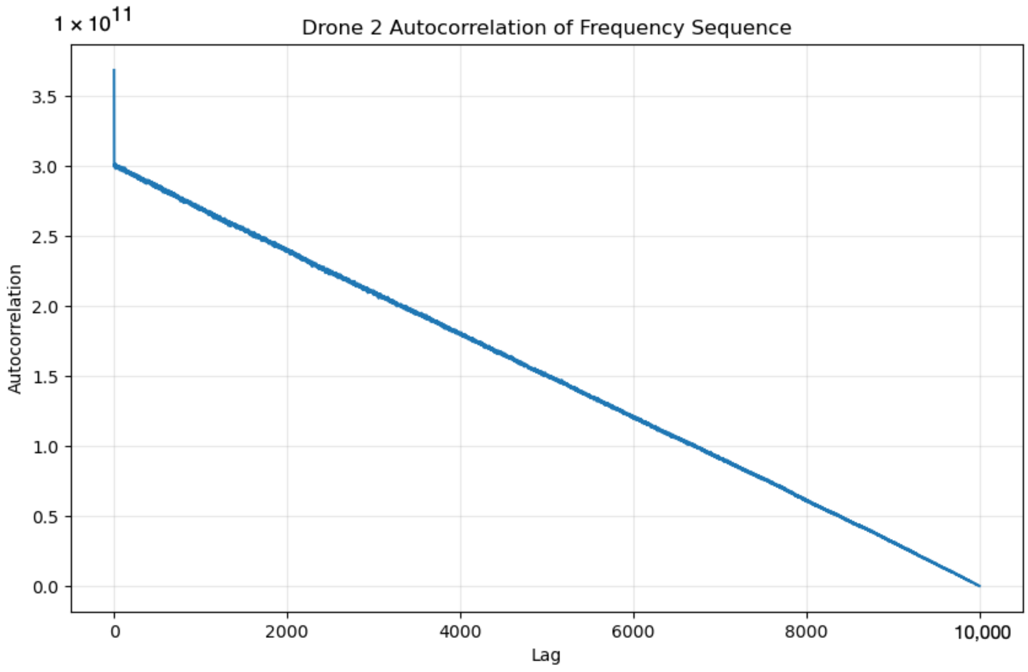

Figure 14. Drone 2: Autocorrelation.

The results indicate a high level of randomness, with Chi-squared values suggesting non-uniformity at a fine scale (in part due to active jamming interference introduced at 5500 Hz). This impact aligns with the frequency distribution patterns observed in each drone's logs, where a slight dip occurs around the jammed frequency. The low p-values for the KS tests indicate statistically significant deviations from perfect uniformity, suggesting that QRNG technology, while highly effective in generating random sequences, may still require further refinement to reduce short-term correlations in the frequency hopping sequences.

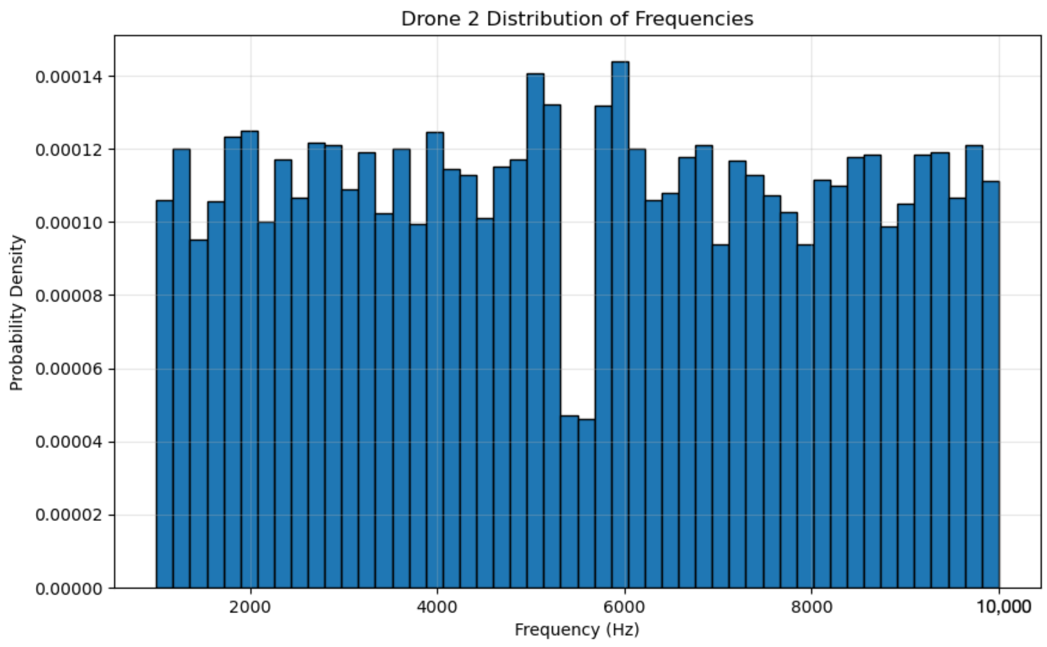

Figure 15. Drone 2: Frequency Distribution, where a noticeable dip around 5500 Hz indicates a reduced frequency count in the jammed region.

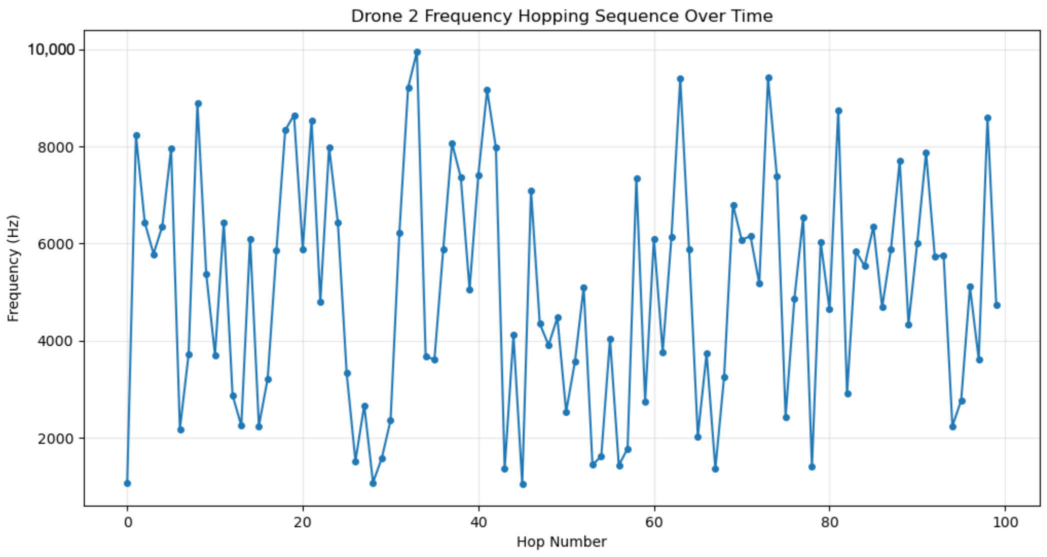

Figure 16. Drone 2: Frequency Sequence over 100 hops of the total 100,000 hops.

The performance logs provide further insight into the stability and robustness of the QRNG-based frequency hopping system. Drones 0, 1, and 2 each maintained reliable communication despite jamming interference, with performance variations reflecting the adaptive response capabilities of the frequency exclusion list management.

- Frequency Exclusion Management: Each drone demonstrated successful avoidance of the jammed 5500 Hz frequency during the experiment, which is evidenced by fewer frequency selections in the impacted range. This illustrates the system's dynamic response to interference, essential for secure drone communication in hostile environments.
- Transmission Integrity: Packet integrity checks indicated no unauthorized modifications or data corruption, verifying the encryption and message authentication protocols in place. The HMAC verification success rate was consistent across all drones, reinforcing the communication channel's resilience under QRNG-driven FHSS.

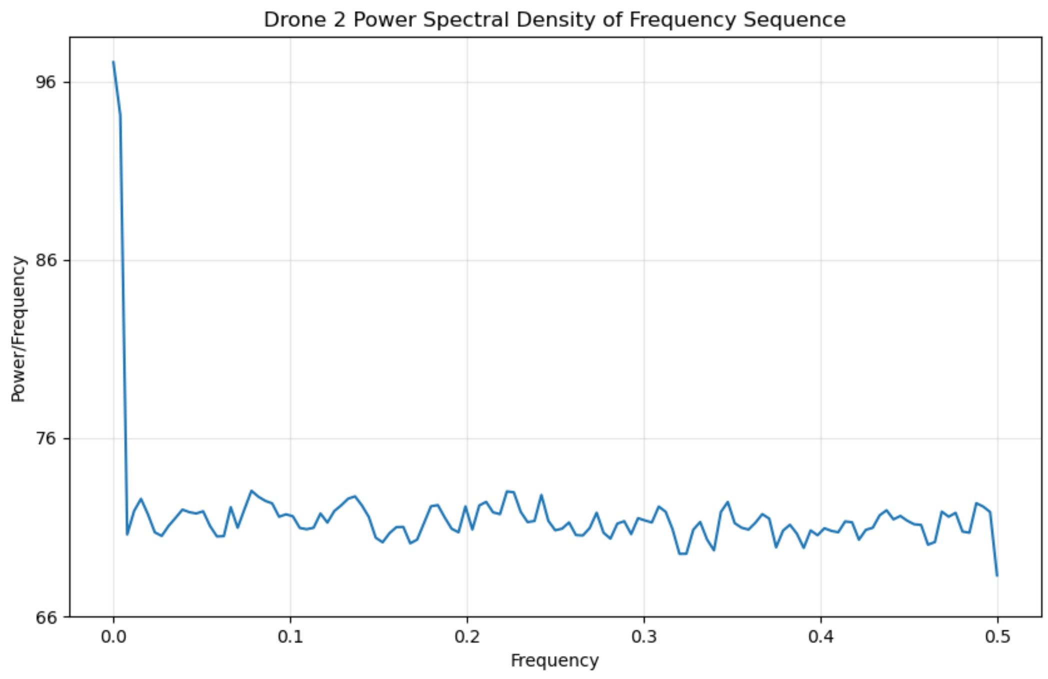

Figure 17. Drone 2: Spectral Analysis.

Overall, the combination of statistical tests and performance metrics provides a comprehensive assessment of the QRNG-based FHSS system's reliability. These results confirm its potential as a robust security solution for multi-drone networks, even in the presence of environmental interference and adversarial actions. Further enhancements to QRNG stability and random sequence generation are anticipated to improve performance in future deployments.

To better position our work within the landscape of related studies, we provide a qualitative comparison table (Table 2). This table contrasts our proposed QRNG-based FHSS method with other security approaches commonly employed in wireless communication and drone networks. Key aspects compared include the type of security technique utilized, each study's adaptability to jamming attacks, the computational complexity, and the source of randomness in frequency generation. This comparison underscores the advantages of using QRNG-based true random frequency hopping for enhanced anti-jamming adaptability and security, particularly in applications where unpredictability is crucial.

Table 2. Qualitative Comparison with Related Studies.

Study	**Technique**	**Anti-Jamming Adaptability**	**Complexity**	**Randomness Source**
[23]	Seamless UAV multicasting	Medium (limited FHSS)	$O(N^2)$ for network transitions	PRNG
[24]	Jammer detection in IoT	High (targeted detection)	$O(N \log N)$	PRNG
This Study	QRNG-based FHSS for drones	High (true random FHSS)	$O(N \log \lvert F \rvert)$ for FHSS	QRNG
[20]	Secure routing for UAVs	Medium (route-level resilience)	$O(N^2)$ for path finding	PRNG
[21]	Protocol optimization	Low (limited hopping)	$O(N)$	PRNG

7.3. Drone Experimental Configuration and Simulated Environment

The experimental setup for this study involved a simulated environment in Python to test the FHSS with QRNG-generated sequences under controlled conditions. The simulation environment involved the following configurations and conditions:

- Drone Configuration: Each drone in the network was represented by an instance of a Python class, which implemented QRNG-based frequency hopping. Each simulated drone had independent QRNG access through shared virtual memory, providing real-time access to randomized frequency values. The drones communicated in a ring topology, where each drone transmitted to two neighboring drones.

- Communication Environment: The simulation modeled a wireless communication channel subject to random interference (jamming) at specific frequency bands to evaluate the system's anti-jamming capabilities. Interference at a central frequency (e.g., 5500 Hz) was introduced to observe the adaptability of the FHSS algorithm.
- Topology: A ring topology was chosen for its suitability for real-world drone networks, where each drone maintains a secure line of communication with its immediate neighbors. This configuration provides resilience and redundancy, as data can flow continuously around the ring even if one link fails.

While tested in a simulated environment, this approach could be adapted to a real-world drone network with ring topology. Each drone would require a lightweight QRNG device or access to a ground-based QRNG for random number generation. The ring configuration would enhance reliability and enable real-time data relay across the network, even under jamming conditions.

8. Discussion

The use of true random frequency hopping is particularly valuable in military and defense contexts, where secure and reliable communication is critical. In a drone cloud configuration, such systems can be employed for reconnaissance, surveillance, and combat support, ensuring that communications remain secure from interception and jamming. The high security provided by true random sequences helps protect sensitive information and command instructions.

Drones equipped with secure communication systems can play a vital role in disaster response and search and rescue operations. In scenarios where the environment may be hostile or communication infrastructure is compromised, a drone cloud using true random frequency hopping can maintain secure and reliable lines of communication. This is crucial for coordinating efforts, relaying information, and ensuring the safety of personnel involved in the operations.

Industries such as agriculture, logistics, and infrastructure inspection increasingly use drone swarms to collect data and perform tasks. True random frequency hopping can secure the communication links between drones, preventing industrial espionage and data theft. For instance, in precision agriculture, drones can securely communicate real-time data on crop health and soil conditions, enhancing the decision-making process.

Drones are extensively used for environmental monitoring, including wildlife tracking, pollution assessment, and natural resource management. Secure communication ensures that sensitive environmental data is protected, and that the drones' operations cannot be easily disrupted. This is particularly important in protected or sensitive areas where unauthorized access or interference could have significant ecological impacts.

In the context of smart cities, drones can be used for infrastructure monitoring, traffic management, and public safety. True random frequency hopping ensures that these drones can communicate securely, protecting data on critical infrastructure and avoiding potential security breaches that could disrupt urban services.

Potential Future Developments

Future developments may involve integrating true random frequency hopping with AI and machine learning algorithms to dynamically optimize communication strategies. For example, AI could predict optimal frequency bands to use based on real-time environmental data, enhancing the system's efficiency and robustness.

The combination of true random frequency hopping with advanced encryption techniques could further enhance communication security. This dual-layer security approach could protect against both physical and digital threats, ensuring comprehensive protection for communication systems.

The principles of true random generation could be extended into the realm of quantum communication. Quantum key distribution (QKD), combined with true random frequency hopping, could provide unprecedented levels of security, leveraging the inherent unpredictability of quantum states.

Future systems may focus on creating interoperable communication platforms that can work seamlessly across different types of drones and other autonomous systems. This could involve standardizing true random frequency hopping protocols to ensure compatibility and secure communication between diverse systems.

Ongoing research will likely focus on miniaturizing RPUs and improving their power efficiency. This is crucial for expanding the use of true random frequency hopping in smaller drones and other devices with limited power resources. Advances in materials science and semiconductor technologies could play a key role in these developments.

As drones become more integrated into critical infrastructure and services, they will become targets for cyber-physical attacks. Future research will likely explore ways to make true random frequency hopping systems resilient not only to traditional jamming and interception but also to sophisticated cyber-attacks aimed at disrupting drone operations.

9. Conclusions

In this paper, we presented a novel approach to enhancing the security and reliability of Frequency Hopping Spread Spectrum (FHSS) systems by integrating a Quantum Random Number Generator (QRNG) to generate truly random frequency hopping sequences. Our methodology was applied to a simulated multi-drone network with a ring topology, where each drone utilized true random sequences for frequency hopping, significantly improving resistance against jamming and interception attempts.

The experimental results provided strong evidence supporting the security and reliability claims of our QRNG-based FHSS system. The statistical analyses, including spectral density, frequency distribution, and autocorrelation studies of the generated frequency sequences, demonstrated the high randomness and unpredictability achieved:

- Uniform Frequency Distribution: The frequency distribution analyses showed a relatively uniform spread of frequencies across the available spectrum for all drones. Despite the presence of simulated jamming at 5500 Hz, the drones effectively minimized frequency usage in the jammed region, indicating successful adaptation and resilience.
- Low Autocorrelation: The autocorrelation analyses revealed sharp peaks at zero lag and near-zero values for other lags, confirming the absence of significant periodicity or predictability in the sequences. This lack of correlation enhances resistance to predictive attacks and jamming.
- Spectral Analysis: The spectral density plots exhibited a relatively flat power spectral density across frequencies, characteristic of truly random sequences. This uniform spectral distribution reduces vulnerabilities to spectral-based jamming and interception.
- Resilience to Jamming: The system demonstrated effective avoidance of the jammed frequency region, maintaining high numbers of frequency hops in non-jammed regions. This adaptability highlightsa the system's robustness in maintaining communication integrity under interference.
- Statistical Randomness Tests: The Chi-squared and Kolmogorov-Smirnov tests performed on the frequency sequences indicated high levels of randomness, with some deviations due to the simulated jamming. The test results confirm that the QRNG-generated sequences exhibit the necessary randomness properties for secure FHSS operation.

The implementation of AES encryption and HMAC verification successfully protected the frequency sequences and synchronization data from interception and tampering. Throughout the experiments, no unauthorized access or data modification was detected, highlighting the effectiveness of the integrated security measures.

While the results are promising, some short-term correlations were observed in the autocorrelation analyses, suggesting areas for improvement in QRNG technology. These correlations, though minimal, indicate that the QRNG devices are still maturing and that future enhancements could further improve the quality of the generated random numbers and the stability of the device.

The experimentation also highlighted practical considerations for deploying such systems in real-world drone networks. The need for efficient distribution and synchronization of true random numbers in airborne drones was addressed through the concept of a virtual shared memory system facilitated by secure communication channels. This approach enables each drone to access high-quality random numbers without the need for physical shared memory, making the methodology practical for flying drones.

Building on these findings, future research will focus on:

- Optimizing Key Management and Synchronization: Enhancing the key management processes and exploring advanced synchronization techniques to improve system performance and security. This includes developing more efficient protocols for secure random number distribution and synchronization among drones.
- Improving QRNG Technology: Testing upgraded versions of QRNG devices to address the observed short-term correlations and improve the quality and stability of the random number generation. Advancements in QRNG technology could lead to even higher levels of randomness and system reliability.
- Real-World Implementation: Implementing the QRNG-based FHSS system in actual drone networks to validate its performance in operational environments. This includes assessing scalability, energy efficiency, and the system's ability to handle real-world interference and environmental factors.
- Integration with Advanced Security Measures: Combining the QRNG-based FHSS system with other cryptographic techniques, such as quantum key distribution, to further enhance communication security. This dual-layer security approach could provide unprecedented levels of protection against both physical and digital threats.
- AI and Machine Learning Integration: Exploring the integration of AI and machine learning algorithms to dynamically optimize communication strategies. AI could be used to predict optimal frequency bands and adjust hopping patterns based on real-time environmental data, enhancing efficiency and robustness.

In conclusion, the integration of QRNG technology into FHSS systems represents a significant advancement in securing drone communications. The experimental results substantiate the security and reliability claims of the proposed system, demonstrating its effectiveness in mitigating potential threats such as jamming and interception. By leveraging true randomness, the system achieves a higher level of unpredictability and resilience, essential for secure communications in critical applications.

This work contributes to the growing field of quantum-enhanced wireless communications, offering substantial benefits for applications where enhanced security is paramount, such as military operations, disaster response, and secure commercial drone activities. The proposed methodology lays the groundwork for future developments in secure communication technologies, paving the way for more robust and reliable systems in increasingly complex and adversarial environments.

Our findings confirm that QRNG-based FHSS systems have the potential to significantly improve the security and reliability of drone communications. As QRNG technology continues to mature, we anticipate further enhancements in system performance and broader applicability across various domains requiring secure wireless communication.

Author Contributions: Conceptualization, J.d.C.; funding acquisition, C.T.C.; investigation, J.d.C. and I.d.Z.; methodology, J.d.C. and I.d.Z.; software, J.d.C.; writing—original draft, J.d.C.; writing—review and editing, J.d.C., I.d.Z., J.-C.C. and C.T.C. All authors have read and agreed to the published version of the manuscript.

Funding: The research received the funding from the BARCELONA Supercomputing Center (TIFON).

Data Availability Statement: Data are contained within the article.

Acknowledgments: During the preparation of this work, the author(s) used ChatGPT/Copilot/Gemini/Claude for code completion and pseudocode generation, Grammarly for English language editing, and Mendeley for bibliography management. The author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of this publication. The authors would like to thank the BARCELONA Supercomputing Center for providing access to MareNostrum 5 and technical support throughout this research. Thanks to the QUANTIC group, Artur García and Germán Navarro, for discussions and instructional examples on the use of the QRNG device.

Conflicts of Interest: The authors declare that they have no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Quantum Random Number Generators	QRNG
Randomness Processing Unit	RPU
Pseudo-Random Number Generators	PRNG
Frequency Hopping Spread Spectrum	FHSS
Advanced Encryption Standard	AES
Hash-based message authentication code	HMAC
Elliptic Curve Cryptography	ECC
Advanced Encryption Standard	AES

References

1. de Zarzà, I.; de Curtò, J.; Roig, G.; Calafate, C.T. LLM Adaptive PID Control for B5G Truck Platooning Systems. *Sensors* **2023**, *23*, 5899. [CrossRef] [PubMed]
2. Dostert, K.M. Frequency-Hopping Spread-Spectrum Modulation for Digital Communications Over Electrical Power Lines. *IEEE J. Sel. Areas Commun.* **1990**, *8*, 700–710. [CrossRef]
3. Liu, F.; Sun, G.; Zhang, S. Adaptive Measurement and Decoding of Frequency-Hopping Spread Spectrum Signals Based on Knowledge Enhanced Compressed Sensing. *IEEE Commun. Lett.* **2021**, *26*, 1155–1159. [CrossRef]
4. Boutsioukis, N. Comparative Analysis of Pseudorandom Number Generators: Mersenne Twister, Middle Square Method, and Linear Congruential Generator Through Dieharder Tests. 2023. Available online: https://ssrn.com/abstract=4354305 (accessed on 1 September 2024).
5. Cannizzo, F. VMT19937: A SIMD-Friendly Pseudo Random Number Generator based on Mersenne Twister 19937. *arXiv* **2023**, arXiv:2309.16682. [CrossRef]
6. Liu, J.; Yao, X.; Wang, Y.; Wang, Z.; Xu, L.; Tong, H.; Liu, K. Novel Random SVPWM Technique Based on Tiny Mersenne Twister PRNG to Reduce PWM Harmonic. In Proceedings of the Third International Conference on Mechanical Design and Simulation (MDS 2023), Xi'an, China, 3–5 March 2023; SPIE Proceedings; Volume 12639, pp. 816–821. [CrossRef]
7. Soler, D.; Dafonte, C.; Fernández-Veiga, M.; Vilas, A.F.; Nóvoa, F.J. A privacy-preserving key transmission protocol to distribute QRNG keys using zk-SNARKs. *Comput. Netw.* **2024**, *242*, 110259. [CrossRef]
8. Zheng, Z.; Guo, X.; Lin, F.; Wang, Y.; Wang, Y.; Guo, Y. Parallel CV-QRNG with Strict Entropy Evaluation. *Photonics* **2023**, *10*, 786. [CrossRef]
9. Haider, Z.; Saeed, M.H.; Zaheer, M.E.; Alvi, Z.A.; Ilyas, M.; Nasreen, T.; Imran, M.; Islam, R.U.; Ikram, M. Quantum Random Number Generator (QRNG): Theoretical and Experimental Investigations. *Eur. Phys. J. Plus* **2023**, *138*, 797. [CrossRef]
10. Pandey, S.K.; Jenef, R. A Comparative Study and Analysis of Quantum Random Number Generator with True Random Number Generator. In Proceedings of the 2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 3–7 January 2024.
11. Lamarr, H.; Antheil, G. Secret Communication System. U.S. Patent No. 2,292,387, 11 August 1942.
12. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
13. Ma, X.; Yuan, X.; Cao, Z. Quantum random number generation. *Npj Quantum Inf.* **2016**, *2*, 16021. [CrossRef]
14. Jennewein, T.; Achleitner, U.; Weihs, G.; Weinfurter, H.; Zeilinger, A. A fast and compact quantum random number generator. *Rev. Sci. Instrum.* **2000**, *71*, 1675–1680. [CrossRef]
15. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
16. Miller, V.S. Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO'85*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 417–426.
17. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
18. National Institute of Standards and Technology (NIST). *Announcing the Advanced Encryption Standard (AES)*; FIPS Publication 197; NIST: Gaithersburg, MD, USA, 2001.
19. Abellán, C.; Amaya, W.; Jofre, M.; Curty, M.; Acín, A.; Capmany, J.; Pruneri, V.; Mitchell, M.W. Ultra-fast quantum randomness generation by accelerated phase diffusion in a pulsed laser diode. *Opt. Express* **2014**, *22*, 1645–1654. [CrossRef] [PubMed]

20. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152. [CrossRef]
21. Shin, D.; Sharma, V.; Kim, J.; Kwon, S.; You, I. Secure and Efficient Protocol for Route Optimization in PMIPv6-Based Smart Home IoT Networks. *IEEE Access* **2017**, *5*, 11100–11117. [CrossRef]
22. Bekmezci, I.; Sahingoz, O.K.; Temel, Ş. Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Netw.* **2013**, *11*, 1254–1270. [CrossRef]
23. Tu, W. Resource-Efficient Seamless Transitions for High-Performance Multi-Hop UAV Multicasting. *Comput. Netw.* **2022**, *213*, 109051. [CrossRef]
24. Abdollahi, M.; Malekinasab, K.; Tu, W.; Bag-Mohammadi, M. Physical-Layer Jammer Detection in Multihop IoT Networks. *IEEE Internet Things J.* **2023**, *10*, 20574–20585. [CrossRef]
25. Zhu, J.; Wang, A.; Wu, W.; Zhao, Z.; Xu, Y.; Lei, R.; Yue, K. Deep-Learning-Based Recovery of Frequency-Hopping Sequences for Anti-Jamming Applications. *Electronics* **2023**, *12*, 496. [CrossRef]
26. Eltholth, A.A. Improved Spectrum Coexistence in 2.4 GHz ISM Band Using Optimized Chaotic Frequency Hopping for Wi-Fi and Bluetooth Signals. *Sensors* **2023**, *23*, 5183. [CrossRef]
27. Sokolov, V.; Skladannyi, P.; Platonenko, A. Jump-Stay Jamming Attack on Wi-Fi Systems. In Proceedings of the 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, 19–21 October 2023; pp. 192–195.. [CrossRef]
28. Pärlin, K.; Riihonen, T.; Le Nir, V.; Adrat, M. Physical-layer reliability of drones and their counter-measures: Full vs. half duplex. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5566–5581. [CrossRef]
29. Ganesan, T.; Jayarajan, N.; Shri Varun, B.G. Dynamic Control, Architecture, and Communication Protocol for Swarm Unmanned Aerial Vehicles. In *Computing in Intelligent Transportation Systems*; Springer International Publishing: Cham, Switzerland, 2023; pp. 31–49.
30. Herbst, J.; Müller, R.; Lipps, C.; Schotten, H.D. A Ring Topology Approach: Efficient Communication in Wireless Body Area Networks (WBANs). In Proceedings of the 2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Antwerp, Belgium, 3–6 June 2024; pp. 1084–1089.
31. Cirauqui, D.; García-March, M.Á.; Guigó Corominas, G.; Graß, T.; Grzybowski, P.R.; Muñoz-Gil, G.; Lewenstein, M. Comparing pseudo- and quantum-random number generators with Monte Carlo simulations. *APL Quantum* **2024**, *1*, 036125. [CrossRef]
32. Mitchell, M.W.; Abellán, C.; Amaya, W. Strong Experimental Guarantees in Ultrafast Quantum Random Number Generation. *Phys. Rev. A* **2015**, *91*, 012314. [CrossRef]
33. Álvarez, J.; Sarmiento, S.; Lázaro, J.; Gené, J.; Torres, J. Random Number Generation by Coherent Detection of Quantum Phase Noise. *Opt. Express* **2020**, *28*, 5538–5547. [CrossRef] [PubMed]
34. Imran, M.; Sorianello, V.; Fresi, F.; Jalil, B.; Romagnoli, M.; Potì, L. On-Chip Tunable SOI Interferometer for Quantum Random Number Generation Based on Phase Diffusion in Lasers. *Opt. Commun.* **2021**, *485*, 126736. [CrossRef]