algorithms

IMPACT FACTOR 2.3

CITESCORE 4.1

Review

# Performance and Applicability of Post-Quantum Digital Signature Algorithms in Resource-Constrained Environments

Marin Vidaković and Kruno Miličević

Special Issue

Surveys in Algorithm Analysis and Complexity Theory, Part II

Edited by

Dr. Jesper Jansson

MDPI

https://doi.org/10.3390/a16110518

*algorithms*

MDPI

*Review*

# Performance and Applicability of Post-Quantum Digital Signature Algorithms in Resource-Constrained Environments

**Marin Vidaković [1] and Kruno Miličević [2,*]**

[1] Sedmi Odjel Ltd., 10000 Zagreb, Croatia; marin.vidakovic@sedmiodjel.com
[2] Random Red Ltd., 31000 Osijek, Croatia
* Correspondence: kruno@randomred.eu

**Abstract:** The continuous development of quantum computing necessitates the development of quantum-resistant cryptographic algorithms. In response to this demand, the National Institute of Standards and Technology selected standardized algorithms including Crystals-Dilithium, Falcon, and Sphincs+ for digital signatures. This paper provides a comparative evaluation of these algorithms across key metrics. The results indicate varying strengths and weaknesses for each algorithm, underscoring the importance of context-specific deployments. Our findings indicate that Dilithium offers advantages in low-power scenarios, Falcon excels in signature verification speed, and Sphincs+ provides robust security at the cost of computational efficiency. These results underscore the importance of context-specific deployments in specific and resource-constrained technological applications, like IoT, smart cards, blockchain, and vehicle-to-vehicle communication.

**Keywords:** quantum-resistant algorithms; digital signatures; computational efficiency standardization



**Citation:** Vidaković, M.; Miličević, K. Performance and Applicability of Post-Quantum Digital Signature Algorithms in Resource-Constrained Environments. *Algorithms* **2023**, *16*, 518. https://doi.org/10.3390/a16110518

Academic Editor: Jesper Jansson

Received: 24 October 2023
Revised: 6 November 2023
Accepted: 8 November 2023
Published: 13 November 2023



## 1. Introduction

The potential of quantum computing is continuously enhanced by exploring hardware-specific quantum algorithms, advancing the theoretical framework for quantum heuristics [1,2], and by contributing to a broad array of other applications that drive the field's overall development. Thereby, the existing cryptographic methods face unprecedented threats. Conventional public-key cryptographic systems, underpinning much of today's secure digital communication, could be rendered obsolete if sufficiently large quantum computers were to be built [3]. The problem is not speculative; the scientific consensus has increasingly moved toward viewing large-scale quantum computing as an engineering challenge rather than as an unresolvable theoretical obstacle. Predictions indicate that within a few decades, quantum computers capable of breaking the current public-key schemes could be a reality.

While the timeline for the advent of quantum computing capable of breaking the current cryptographic systems is still uncertain, the long lead time required to implement new cryptographic infrastructures warrants immediate action. Specifically, the existing systems must be adapted or replaced to withstand attacks from both classical and quantum computers. In this context, the urgency of developing and implementing post-quantum (or quantum-resistant) cryptographic algorithms is palpable [3].

The National Institute of Standards and Technology (NIST) initiated a process in 2016 to develop and standardize one or more public-key cryptographic algorithms to bolster the existing FIPS 186-4 Digital Signature Standard (DSS) [4]. In response to this call, multiple rounds of standardization processes were carried out [5–7]. The third round, conducted in July 2022, finalized the selection of post-quantum algorithms specifically for public-key encryption and digital signatures [8]. In general, these selections aim to specify additional unclassified, publicly disclosed algorithms capable of securing sensitive information in the foreseeable future, including the post-quantum era.

Of the algorithms chosen, Crystals-Dilithium, Falcon, and Sphincs+ pertain to digital signatures. Crystals-Dilithium and Falcon are lattice-based schemes, with considerations for both depending on the context of application—Dilithium is easier to implement, while Falcon yields shorter signatures [9]. Sphincs+, although slower and larger, is significant because it is based on a different mathematical approach, i.e., hash functions. This diversification offers a fail-safe against unforeseen vulnerabilities in lattice schemes. Furthermore, a "hybrid mode" is permitted, enabling the use of quantum-resistant algorithms alongside already approved NIST algorithms without compromising FIPS validation [10].

While the NIST PQC process provides a roadmap, it is crucial to also consider implementation, integration into existing systems, and potential adversarial strategies. From a research and development perspective, this demands interdisciplinary expertise in quantum computing, cybersecurity, and algorithmic theory.

## 2. About the Chosen Digital Signature Algorithms

Dilithium, Falcon, and Sphincs+, the algorithms that were shortlisted by the NIST for post-quantum cryptographic standardization, matured over several iterations, and their latest (third submission) versions were analyzed in this paper as they offer refined security and performance trade-offs. The standardization of Dilithium and Sphincs+ is currently in draft stage [11,12]. Falcon's first draft standard is anticipated in 2024 [13].

### *2.1. Dilithium*

Dilithium is a digital signature scheme that is highly secure against chosen-message attacks, based on the hardness of lattice problems over module lattices [14].

The design of Dilithium is grounded in Lyubashevsky's "Fiat–Shamir with Aborts" technique [15,16], which employs sample rejection to make lattice-based Fiat–Shamir schemes compact and secure. It most closely resembles schemes proposed in [17,18].

#### 2.1.1. Selected Parameters and Performance Metrics

Table 1 shows the basic parameters of Dilithium, detailed further in [19]. Figure 1 shows the performance of Dilithium in terms of execution speed.

**Table 1.** Basic parameters of Dilithium [19].

| Security Level | 2 | 3 | 5 |
|---|---|---|---|
| Public Key Size (bytes) | 1312 | 1952 | 2592 |
| Signature Size (bytes) | 2420 | 3293 | 4595 |

#### 2.1.2. Advantages and Disadvantages According to the Authors

The authors of [20] listed the following advantages of Dilithium:

- Significantly higher speed and significantly smaller size compared to hash-based schemes (e.g., the Sphincs+ scheme).
- Simple implementation, as Gaussian sampling is not required.

They focused on the comparison between Dilithium and Falcon, as these are the only two lattice-based schemes in the third round of NIST's call, and NIST announced that it will standardize only one such scheme.

The authors emphasized that Falcon's signature and key sizes are about 2.3 times smaller than those of Dilithium (Figure 2) but pointed out its disadvantages:

- It uses high-precision Gaussian sampling (64 bit precision), which makes it difficult to notice subtle implementation errors (the distribution would still look Gaussian even if it is not satisfactory), possibly leading to the leakage of the secret key.
- It is complex to mask, and there have been no serious attempts to improve this, so far. However, masking may not be required for signing a small number of messages (in the order of 100 messages).

- Dilithium has the advantage of using only uniform sampling within a power-of-two range, making it much easier to detect implementation errors.

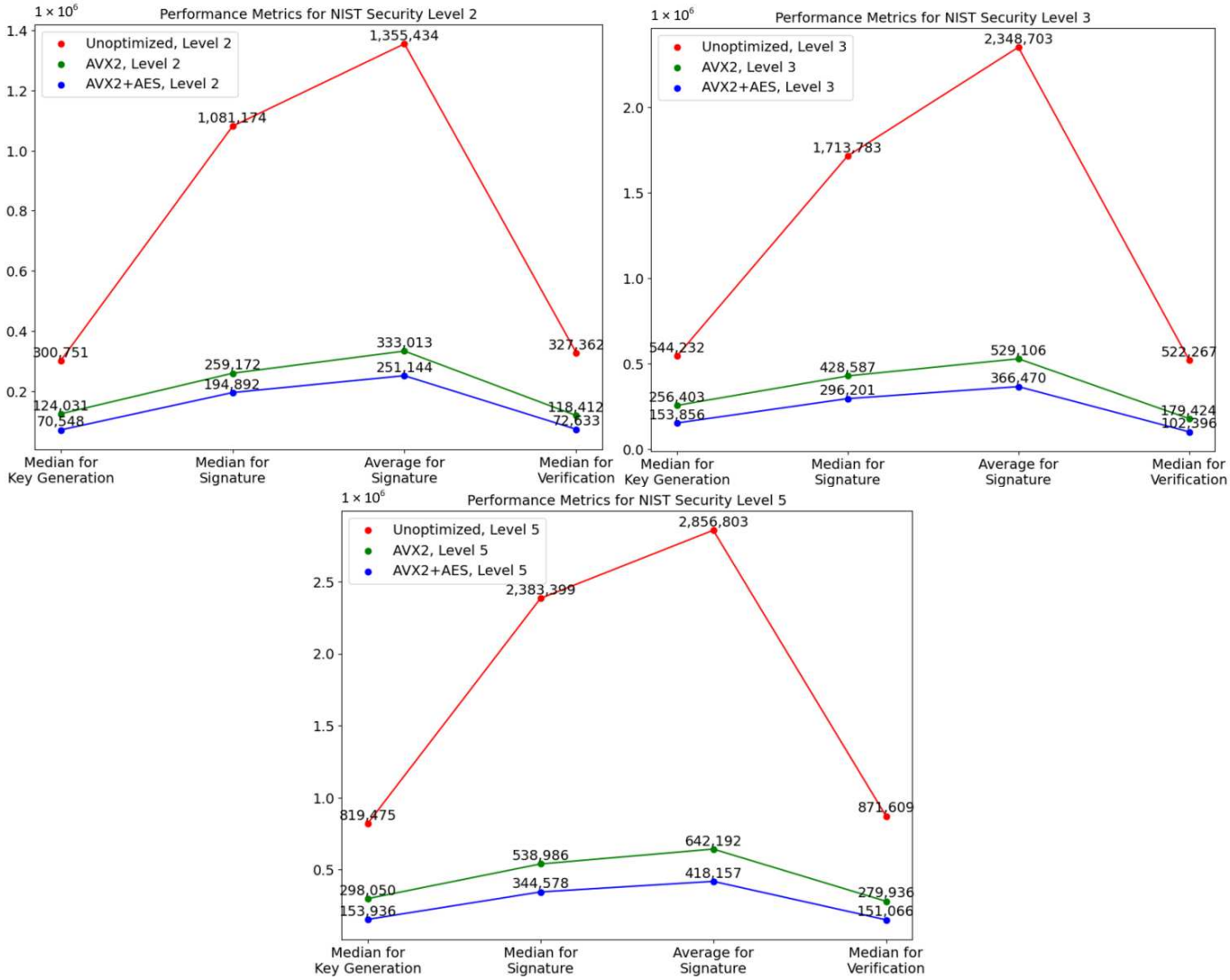


**Figure 1.** Execution speed of Dilithium (cycle counts) [19].

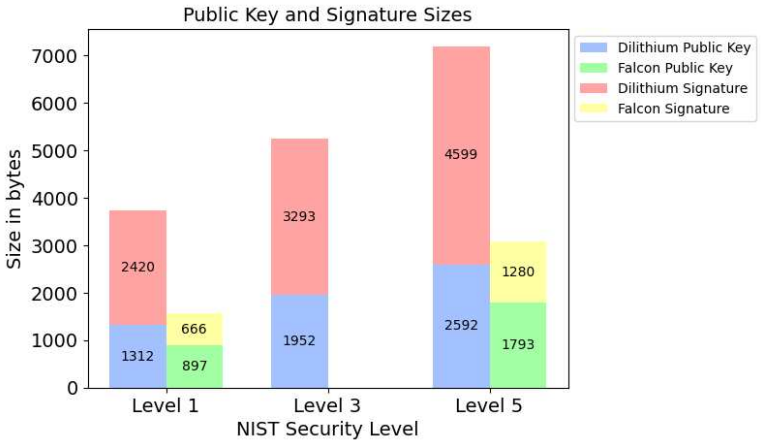


**Figure 2.** Comparison of Dilithium and Falcon by public key and signature lengths (Falcon's values for Level 3 are not available) [20].

Further, the authors compared the two algorithms by their execution speed (according to [9,21]), concluding that Falcon's only advantage is in much faster verification (Table 2).

**Table 2.** Comparison of Dilithium [21] and Falcon [9] by execution speed on Cortex M4 (execution speed (cycles)/memory).

| Parameters | Dilithium—NIST Security Level 3 | Falcon—NIST Security Level 1 |
|---|---|---|
| Key Generation Speed | 6 M/10 KB | 171 M/16 KB |
| Signature Speed | 8 M/70 KB<br>26 M/11 KB<br>6 M/21 KB + 48 KB Flash | 40 M/40 KB<br>21 M/25 KB + 57 KB Flash |
| Verification Speed | 2.7 M/11 KB | 0.5 M/4 KB |

While both algorithms are competitive, the authors of [20] stressed the applicability of Dilithium and Falcon for specific uses. In the end, both schemes were standardized, since Dilithium is easier to implement, and Falcon provides significantly shorter signatures. The authors suggested that Dilithium may serve as a "general-purpose" algorithm, while Falcon could be used in applications requiring multiple signatures (in the order of 100–1000 signatures).

### *2.2. Falcon*

Falcon is a signature scheme grounded in lattice cryptography, specifically designed for efficient and compact execution over NTRU lattices, as indicated by its full name, "Fast-Fourier lattice-based compact signatures over NTRU." The architectural simplicity of Falcon derives from its implementation of the theoretical blueprint set forth by Gentry, Peikert, and Vaikuntanathan in their 2008 paper [22] for creating lattice-based hash-and-sign algorithms. This theoretical model demands two primary components:

- A designated family of cryptographic lattices, for which Falcon selects NTRU lattices.
- A mechanism for trapdoor sampling, with Falcon employing an innovative method known as fast Fourier sampling.

To encapsulate this, Falcon's signature scheme can be concisely described as the combination of the GPV framework, NTRU lattices, and fast Fourier sampling.

#### 2.2.1. Selected Parameters and Performance Metrics

The recommended parameters and performance metrics for Falcon are presented in Table 3. These performance metrics are based on an implementation on an Intel® Core® i5-8259U CPU ("Coffee Lake" core, clocked at 2.3 GHz) [23].

**Table 3.** Parameters for two versions of Falcon [23].

| Parameters | Falcon-512 | Falcon-1024 |
|---|---|---|
| NIST Security Level | 1 | 3 |
| Public Key Length (bytes) | 897 | 1793 |
| Signature Length (bytes) | 666 | 1280 |
| Key Generation Time (ms) | 8.64 | 27.45 |
| Key Generation (RAM) | 14,336 | 28,672 |
| Signatures per Second | 5948.1 | 2913 |
| Verifications per Second | 27,993.0 | 13,650.0 |

#### 2.2.2. Advantages and Disadvantages According to the Authors

According to [24], the advantages include:

- The most bandwidth-efficient algorithm, as shown in Figure 3.

- The modular design; for instance, NTRU lattices could be replaced with other lattice types.
- Extremely fast verification.
- A broad and deep body of research supports lattice security.
- Better-examined security against side-channel attacks.

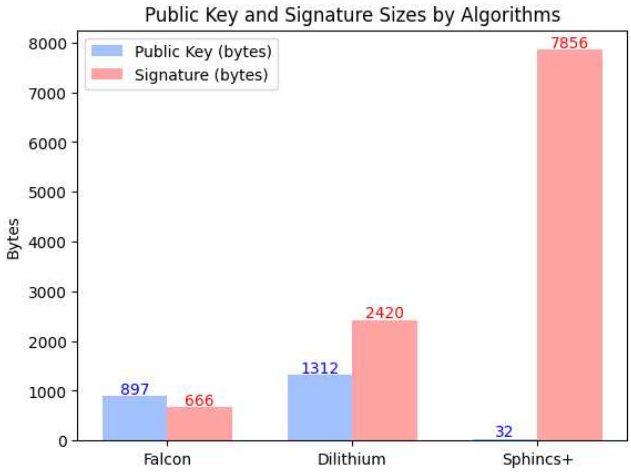


**Figure 3.** Key and signature size for Falcon and some competing algorithms [24].

Disadvantages:

- Complex key and signature generation processes.
- Key and signature generation relies on floating-point arithmetic.
- For Falcon, and generally for all other schemes, the risk of side-channel attacks still needs further research.

### *2.3. Sphincs+*

Sphincs+ serves as a cutting-edge post-quantum digital signature algorithm engineered to withstand quantum-computing attacks. Grounded in hash-based cryptography, the scheme's security is straightforward to evaluate and hinges solely on the characteristics of the hash function in use. Sphincs+ essentially builds on the foundational architecture of Sphincs but enhances it through parameter optimization and the incorporation of novel methods aimed at boosting both the algorithm's speed and its security profile [25].

#### 2.3.1. Selected Parameters and Performance Metrics

The architecture of Sphincs+ allows for a multitude of trade-offs between the speed of the algorithm and the size of the digital signature, even at a given security level. Table 4 presents six distinct sets of parameters that, in conjunction with the cycle counts provided in Figure 4, demonstrate how these trade-offs manifest in practice.

**Table 4.** Examples of parameter sets for Sphincs+ targeting different security levels and different trade-offs between size and speed [25].

| | Bitsec | NIST Security Level | Sig Bytes |
|---|---|---|---|
| Sphincs+-128s | 133 | 1 | 7856 |
| Sphincs+-128f | 128 | 1 | 17,088 |
| Sphincs+-192s | 193 | 3 | 16,224 |
| Sphincs+-192f | 194 | 3 | 35,664 |
| Sphincs+-256s | 255 | 5 | 29,792 |
| Sphincs+-256f | 255 | 5 | 49,856 |

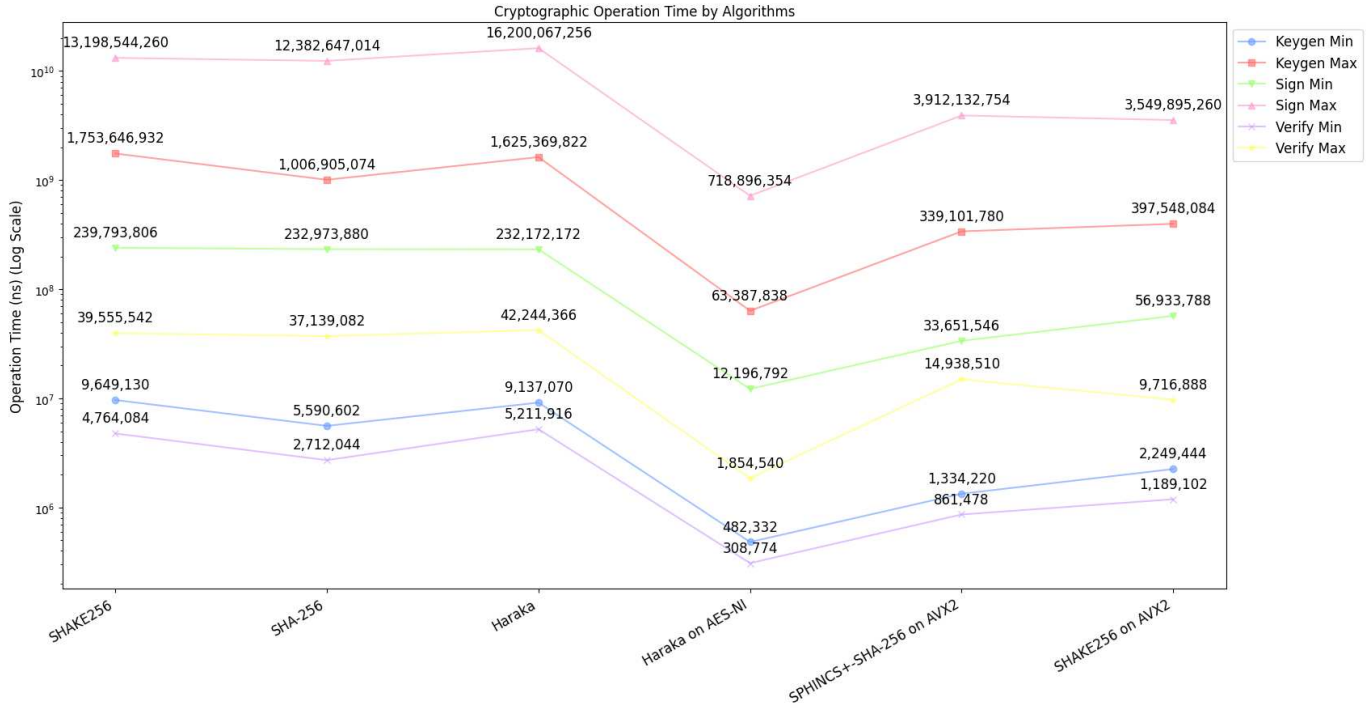


**Figure 4.** Runtime benchmarks for Sphincs+, Sphincs+-Haraka on AES-NI, Sphincs+-Sha-256 on AVX2, Sphincs+-Shake256 on AVX2 [25].

Figure 4 shows the algorithm's performance metrics (minimum and maximum values) when run on a 3.1 GHz Intel Xeon E3-1220 CPU (Haswell architecture), with 32 GB of memory. Optimized implementations for platforms compatible with the AVX2 instruction set are also available. Specifically, for Haraka, the availability of the AES-NI instruction set is particularly noteworthy. Each of six implementations has its version for:

- One, three and five NIST security levels.
- Small and fast parameter sets.
- Simple and robust instantiations of the tweakable hash functions.
- Offering various trade-offs.

Table 5 outlines the sizes of the public keys, secret keys, and digital signatures in bytes. Regarding memory utilization, the reference implementation generally leans toward minimal stack consumption.

**Table 5.** Key and signature sizes in bytes [25].

| | Public Key Size | Secret Key Size | Signature Size |
|---|---|---|---|
| Sphincs+-128s | 32 | 64 | 7856 |
| Sphincs+-128f | 32 | 64 | 17,088 |
| Sphincs+-192s | 48 | 96 | 16,224 |
| Sphincs+-192f | 48 | 96 | 35,664 |
| Sphincs+-256s | 64 | 128 | 29,792 |
| SphincS+-256f | 64 | 128 | 49,856 |

2.3.2. Advantages and Disadvantages According to the Authors

The essence of Sphincs+ lies in its dichotomy: while it arguably represents the most cautious approach to post-quantum digital signature architecture, it pays the price in efficiency, in terms of both signature size and computational speed.

Disadvantages:

- Signature size and speed: Sphincs+ is not designed to be the fastest or the smallest, although it does offer trade-offs between these two metrics.

Advantages:

- Minimal security assumptions: the security is entirely predicated on well-understood hash functions, avoiding new computational hardness assumptions.
- Easily analyzed attacks: the current state-of-the-art attacks, both classical and quantum, can be straightforwardly analyzed, enabling precise security quantification.
- Small key sizes: Sphincs+ boasts compact public keys, a benefit in scenarios where public keys are often transmitted.
- Overlap with XMSS: the scheme harmonizes well with XMSS, facilitating their combined use in specific applications like VPNs.
- Reusable building blocks: speed enhancements in underlying hash functions directly translate into Sphincs+ performance gains.

## 3. Parameter and Performance Comparison

### *3.1. Key and Signature Size*

Two relevant parameters are compared in the analysis of algorithms: key size and signature size, for which smaller values are better. The data on key length and signature length are presented in Figure 5 [19,23,26]. They are similar to the data reported for Dilithium and Falcon in [27].

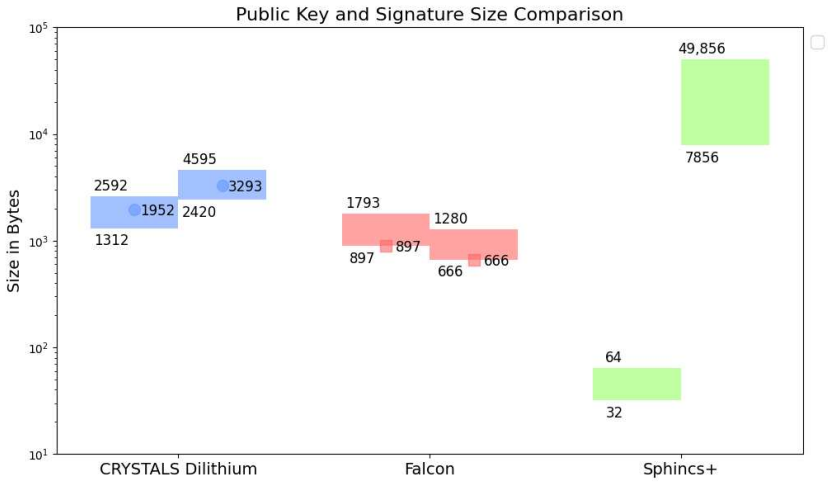


**Figure 5.** Comparison of public keys (left bars/dots) and signatures (right bars/dots) according to [19,23,26] (bars) and [27] (dots).

The two lattice-based algorithms (Crystals-Dilithium, Falcon) have public keys with comparable sizes, while Sphincs+ has a significantly shorter public key length, as seen in Figure 5. On the other hand, Sphincs+ has the largest signature length.

### *3.2. Execution Speed*

The performance speed of an algorithm is evaluated based on the speed of executing three operations: key generation, signing, and signature verification. The execution speed depends on the platform on which a PQC (post-quantum cryptography) algorithm is run, as well as on the implementation of the algorithm. The scientific and professional community continuously proposes new algorithm implementations with the goal of their optimization and achieving better performance.

Performance speed tests are more detailed for the following platforms:

- Arm Cortex M4.
- x86/x64 Processors.

Besides the two mentioned platforms (Arm Cortex M4 and x86/x64 processors), there are implementations of PQ algorithms for other platforms. However, these are individual implementations where a comparative analysis of all three digital signature algorithms is not possible. For example, Dilithium was implemented and tested on various FPGA

platforms [28–31], on the Cortex M3 platform [21], on ARMv8 platforms (Cortex-A72 and Apple M1) [32], as well as on IBM Z computers [33]. Falcon's implementation was also described on ARMv8 platforms (Jetson Xavier CPU with 8 ARMv8.2 cores) [34], and the implementation of Sphincs+ was reported on FPGA [35] and ARM Cortex M3 platforms [36].

#### 3.2.1. Execution Speed on ARM Platforms

Arm Cortex M4 was chosen as the reference platform to test the performance of PQC algorithms on microcontroller systems. Almost all finalists were implemented and tested within a publicly available test platform [37] for various types of cryptographic scheme implementations (clean, reference implementation submitted to NIST, optimal in plain C, optimized for the Cortex-M4), as shown in Figure 6. The presented data include 3 versions of Dilithium (Dilithium2, Dilithium3, Dilithium4), 3 versions of Falcon (Falcon512, Falcon1024, Falcon512-tree), and 36 versions of Sphincs+ (see Section 2.3). A comparison is also shown according to [21] for Dilithium (Dilithium2, Dilithium3, Dilithium4) and Falcon [9], which roughly aligns with the comparison according to [37].

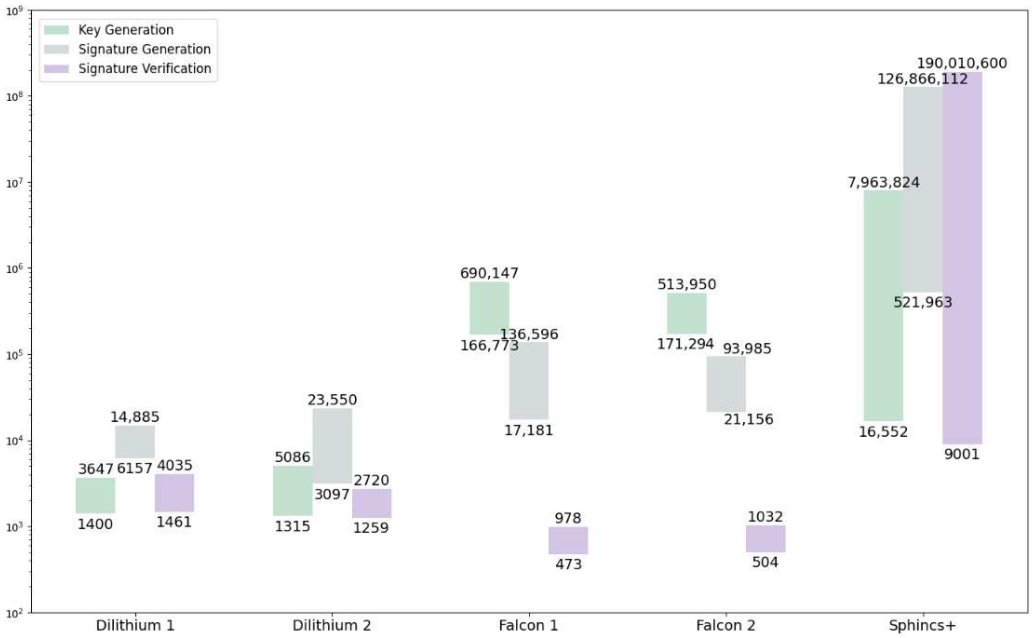


**Figure 6.** Execution speeds (in thousands of clock cycles) of digital signature algorithms on the Arm Cortex M4 platform according to [37] ("Dilithium 1", "Falcon 1" and "Sphincs+" bars), [9] ("Falcon 2" bars) and [21] ("Dilithium 2" bars).

#### 3.2.2. Execution Speed of Digital Signature Algorithms on x86/64 Platforms and Energy Consumption

Data from [38] show the execution speed on the x86/x64 platform with an i7-6700 processor, and [27] provides execution time data for the i7-1165G7 processor for Dilithium and Falcon, as shown in Figure 7. Both [38] and [39] provide data on energy consumption, concluding that energy consumption is roughly proportional to the speed, i.e., the execution time, as power consumption has much smaller deviations (maximum 20% in [38], maximum 50% in [39]) compared to execution speed (which varies by several orders of magnitude).

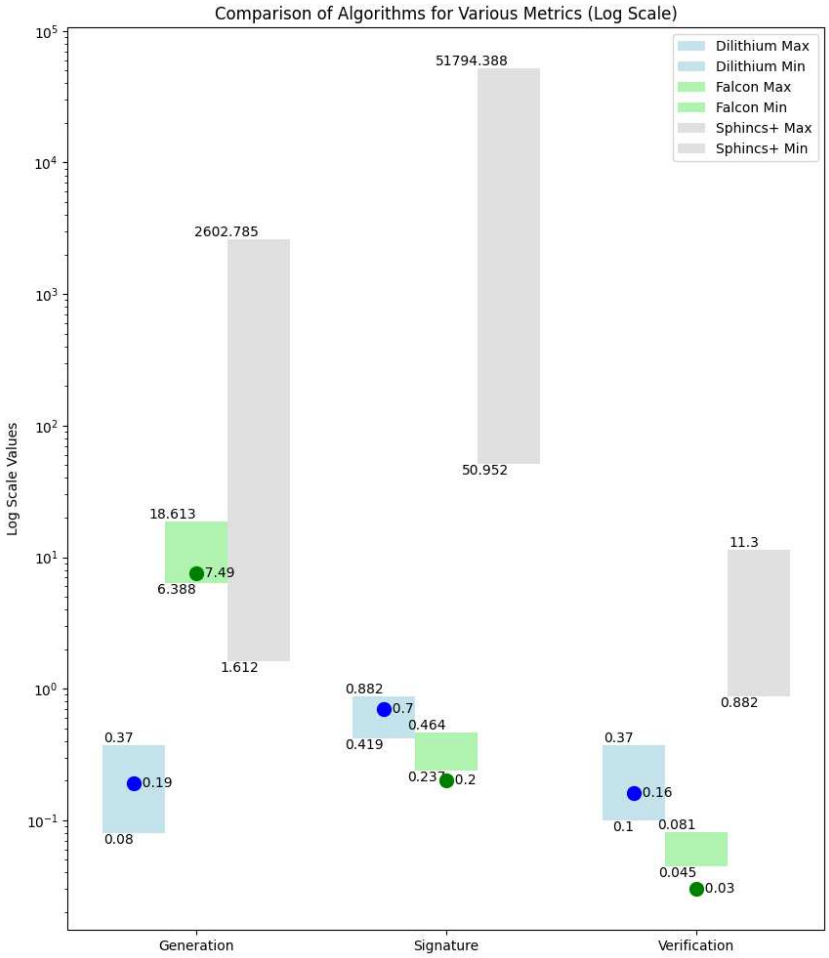


**Figure 7.** Key generation speeds (in milliseconds) for digital signature algorithms for the i7-6700 processor (Dilithium for NIST levels 1–3, other algorithms for levels 1–5; bars) [38] and the i7-1165G7 processor (Dilithium 3 and Falcon 512; blue and green dots) [27].

### *3.3. Recent Algorithm Implementations*

In the current landscape of algorithmic research, there is considerable effort devoted to the enhancement of standardized algorithms through specific implementations. As illustrated in Table 6, a review—conducted based on the authors' estimation for relevance to the topic of this paper—of publications from the years 2022 and 2023 indicated a substantial concentration of studies focused on Dilithium.

**Table 6.** Recent relevant algorithm implementations—list of publications (2022 and 2023).

| | **Publications** |
|---|---|
| Dilithium | [40–48] |
| Falcon | [48] |
| Sphincs+ | [48–51] |

However, active research efforts regarding all three algorithms will warrant their further development, particularly in the context of applicability on performance-constrained platforms, as discussed in next section.

## 4. Applicability on Performance-Constrained Platforms

Real-world systems operate under severe resource constraints, requiring cryptographic algorithms that are not only secure against quantum threats but also computationally efficient. Speed is another common factor; whether it is real-time V2V communication or

swift financial transactions via smart cards, the selected post-quantum algorithms must offer rapid signature verification and message processing. Moreover, the transition to post-quantum secure algorithms involves a complex trade-off between security, computational load, key sizes, and signature verification times, often necessitating hardware-accelerated solutions or innovative approaches like threshold signatures.

Also, the NIST's other recent initiative [52,53] underscores the critical need for lightweight cryptography in constrained environments as analyzed, for example, in [54]. These efforts reflect a future trend in the cryptographic community towards developing solutions that are both quantum-resistant and resource-efficient. Namely, as quantum computing progresses, the expectation is that quantum-resistant cryptographic algorithms will also need to be lightweight to be practical for devices with limited computational capabilities.

Thus, in our selection, we focused on systems that represent a diverse range of constraints and for which relevant data on post-quantum signature performance are available.

### *4.1. IoT Systems*

IoT systems encompass a wide range of applications, from modern smart home sensors to devices embedded in critical infrastructure like power grids, traffic control, and water supply systems [55]. These are often devices that are installed for extended periods.

The NIST uses the ARM Cortex M4 platform as a representative for embedded systems, whereas many IoT systems rely on processors with significantly lower performance capabilities (less RAM, slower processors). According to [55], Falcon is a better choice for implementing post-quantum (PQ) digital signatures on IoT platforms with limited RAM (up to about 64 KB). A similar conclusion is drawn by the authors of [56], where PQ signatures are verified on systems with 8 KB of RAM (ARM Cortex M3), including Sphincs+, thus making all three algorithms applicable for IoT in a broader context. Sphincs+ is also applicable for ARM Cortex M3 [36].

The authors circumvented the problem of oversized keys by streaming the keys into memory in parts and performing operations accordingly [56]. In general, in lattice-based signatures like Dilithium2 and Falcon-512, hashing constitutes a significant portion of the computational load, accounting for 65% and 36% of it, respectively. Similarly, in hash-based signature schemes like Sphincs-Sha256-128s-Simple and Sphincs-Sha256-128f-Simple, hashing operations represent an even larger share, corresponding to 90% and 88%, respectively. Both types of schemes would benefit more from hardware-accelerated hashing.

In a research paper [57], the authors presented a post-quantum secure multi-party collaborative signature scheme that enhances deterability in Industrial Internet of Things (IIoT) settings. This approach addresses both fairness and privacy protection issues that are commonly found in conventional IIoT security mechanisms. To validate the practicality of this Dilithium-based protocol, the authors developed and tested a prototype to assess its performance.

In their research [58], the authors showed the FPGA implementations of all three round-3 parameter sets for Dilithium, targeting the low-end Artix-7 platform. Their design adhered to a universal objective: it achieved low latency in comparison to other post-quantum secure signature algorithms, while maintaining a small area footprint. This dual advantage makes the use of Dilithium viable in a range of low-cost and resource-constrained environments, including IoT.

A paper [59] investigated an area-efficient FPGA implementation of Sphincs+, a post-quantum hash-based signature scheme. By leveraging resource sharing in a compact hardware solution, the authors successfully adapted Sphincs+ for resource-constrained systems like IoT devices and nano-satellites.

Another paper [60] compared the suitability of Falcon and Dilithium signature schemes for IoT applications, focusing on verification performance and transmission cost. Falcon emerged as the more efficient choice in terms of both speed and resource utilization, particularly in FPGA environments. This advantage is likely to extend to ASICs, suggesting a better time–area product for Falcon. However, Dilithium presents an advantage in scenarios

requiring frequent key generation or client-side signing due to its lower complexity in these operations. Thus, while Falcon offers superior verification and transmission efficiencies, Dilithium may be preferable for applications with specific key management requirements.

### *4.2. Smart Cards*

Smart cards, which form the basis for contactless payments, present a challenge for the implementation of post-quantum (PQ) algorithms, especially digital signature algorithms. Compared to the ARM Cortex M4 platform, they have significantly less memory (four times less) and a slower processor (1.7 times slower). When compared with x86-64 platforms, they have 170,000 times less memory and a processor that is 40 times slower. The limited processor speed is further constrained by the slow communication speeds supported (below 100 kB/s) [61].

Currently, smart cards have hardware blocks implemented for random number generation, AES encryption (supporting one AES block), and ECC/RSA signatures (enabling parallel execution and speeds up to 10 times higher than software implementations). As modern smart cards do not have dedicated hardware for executing PQ digital signatures, and the process of implementation and certification is lengthy (6 to 18 months for certification alone), consideration is being given to how PQ algorithms can be executed on the current versions of smart cards.

During algorithm implementation, it is essential to employ masked algorithms to achieve protection against side-channel attacks. Masked algorithms generally require longer execution times and consume more memory (for example, the execution time for Dilithium increases by 5.6 times) [62].

In conclusion, only Dilithium fits within the constraints of smart cards. Falcon requires too much RAM and too much time for key generation. Sphincs+ has an issue with the size of the signature, which can be partly mitigated by hash-based accelerators for embedded devices but not for smart cards [63].

For example, the authors of [64] introduced a variant of Dilithium known as DiLizium, which has the potential to replace distributed RSA and ECDSA signature schemes in authentication contexts, including applications like Smart-ID.

### *4.3. Vehicle-to-Vehicle Communication*

Vehicle-to-vehicle (V2V) communication enables timely warnings about approaching vehicles, which is especially useful in situations without a direct line of sight [65]. Each vehicle broadcasts 10 basic safety messages (BSMs) per second [66]. These BSMs contain information on the direction of travel, current location, time, speed, and the status of the gas and brake pedals. Currently, secure wireless communication between vehicles is facilitated through the IEEE 1609.2 standard, which uses elliptic curve digital signature algorithm (ECDSA) for digital signatures.

Given that the transmission of messages (BSMs) occurs over very short distances, both message transmission and verification must be completed within the order of milliseconds. Therefore, an analysis was conducted to determine whether post-quantum (PQ) digital signature algorithms meet these criteria.

Regarding quick signature verification, Dilithium and Falcon appeared to be suitable replacements for ECDSA. However, when the size of the signature was also considered, Dilithium became inapplicable as its signature size exceeds the message size (2304 bytes) defined in IEEE 802.11p. Falcon was proposed as the most appropriate algorithm for V2V communication [66].

Similar conclusions were reached in [67], where the authors evaluated the performance of Dilithium and Falcon for over-the-air (OTA) updates used in vehicle-to-everything (V2X) communication infrastructures. Namely, an experimental evaluation was conducted on the Xilinx Zynq Ultrascale+ ZCU102 board, tailored for automotive applications, to assess the performance of the Dilithium and Falcon cryptographic algorithms. The findings indicated

that Falcon outperformed Dilithium in both signature verification latency and signature byte footprint.

In a paper [68], the authors achieved significant optimizations of the Dilithium algorithm for Aurix implementation, specifically targeting automotive applications. They managed to reduce the computational time for key generation, signature creation, and verification across varying security levels. Specifically, they reported time reductions in the range from 23% to 29% for key generation, from 12% to 19% for signature creation, and from 18% to 26% for signature verification.

A paper [69] discussed the practical integration of Dilithium into multiprocessor system-on-chip (MPSoC) architectures, commonly employed in automotive environments and managed by a commercial real-time operating system (RTOS). Experimental data revealed that deploying Dilithium on a multi-core framework significantly enhanced the performance metrics—yielding improvements of up to 48% in key generation, 34% in the signature process, and 42% in verification tasks—when contrasted with single-core configurations.

Other researchers [70] investigated the constraints of conforming to IEEE 1609.2 standards in the context of the WAVE protocol for vehicle-to-vehicle (V2V) communications. They identified the inability to incorporate cryptographic schemes with large signature sizes as a fundamental limitation. To circumvent this issue, they introduced a technique termed "split way for Crystals-Dilithium," aimed at ensuring the WAVE protocol's integrity while adhering to maximum packet size restrictions.

### *4.4. Blockchain Technology*

For establishing trust in blockchain technology, it is essential to ensure three pillars: security, authenticity, and integrity. It is crucial that blockchains maintain these attributes even in the post-quantum era, with asymmetric cryptography being the foundational method to secure these properties. The authors of [71] pointed out the shortcomings of Sphincs+ and proposed the use of Dilithium in blockchain technology, not necessarily as a standalone algorithm but also as an enhancement to existing algorithms—essentially, as a hybrid solution. The use of hybrid algorithms was also suggested in [72].

According to [27], Falcon has the most compact key and signature lengths among the lattice-based cryptographic algorithms studied. While alternatives like Dilithium offer rapid performance, they come at the cost of significantly larger keys and signatures. Given this, there is growing consensus that Falcon's integration into blockchain technology could be promising for future advancements.

For key generation in blockchain applications, threshold signatures can also be employed [73]. This method replaces the traditional KeyGen and Sign algorithms of digital signatures with an interactive multi-party protocol. The KeyGen protocol involves multiple participants who interactively generate a key. The Sign algorithm is then replaced with an interactive protocol for generating signatures. A unique feature of this protocol is that a sufficient number of participants must agree to sign a message before a signature can be generated. The verification of the signature is performed in the standard way. This ensures that an attacker cannot generate a signature on any message that has not been signed by an "honest" party. According to [73], Sphincs+ algorithms are hindered by the need to compute a large number of hash function queries on secret data.

Post-quantum digital signature-based blockchain was also proposed as an appropriate method for securing the mobile internet of things (MIoT) ecosystem, increasingly becoming important with the wider availability and increasing speeds of mobile internet (5G and 6G) [74].

In a paper [75], four quantum-resistant algorithms—Sphincs+ + Sha256s, Sphincs+ + Shake256s, Dilithium, and Falcon—were evaluated for their efficiency and practicality. While Dilithium emerged as the most computationally efficient algorithm, it suffered from the disadvantage of having the largest public key size, posing significant challenges for its application in blockchain systems where each transaction embeds the public key.

However, after incorporation of the interplanetary file system (IPFS), Dilithium maintained its efficiency and experienced a significant reduction in both signature and public key sizes to just 32 bytes each. Notably, Sphincs+ also benefits from IPFS, with a remarkable 99% reduction in signature size.

A paper [71] illustrated that Dilithium can be effectively configured to provide varying levels of quantum security in blockchain systems without significantly inflating the block size. Specifically, an 80 bit quantum security level yields a block size of less than 1 MB, closely aligning with the current 1 MB block size in Bitcoin. For moderate and high security levels, the block sizes range from 1.5 to 2.0 MB and approach 3.0 MB, respectively. These are still within acceptable limits for practical blockchain deployment. Moreover, the performance gains are optimized further by efficient AVX2 implementations, which are consistent with the efficiency trends observed in Skylake architectures. This flexibility makes Crystals-Dilithium a versatile choice for enhancing quantum security in blockchain, accommodating both current needs and future upgradations without burdening the system with large block sizes.

A paper [76] evaluated the viability of using the Falcon signature algorithm in the context of blockchain, specifically within the Ethereum mainnet. The analysis revealed a prohibitive cost for Falcon signature verification, consuming an average of 500 million gas units. This is substantially higher than the current block gas limit of 12 million units in Ethereum.

Another paper [77] presented a compelling argument for replacing W-OTS, the current signature scheme in IOTA—a distributed ledger technology (DLT) tailored for internet of things (IoT) and internet of vehicles (IoV) applications—with the post-quantum signature scheme Dilithium. The findings indicated that Dilithium offers significant advantages over W-OTS in multiple dimensions. Firstly, it accelerates transaction processing and reduces signature space, thereby cutting down latency, which is critical in fast-paced IoV networks. Secondly, while W-OTS necessitates the repetitive generation of public and private keys, Dilithium requires only a one-time generation, offering a more efficient operational model. Furthermore, Dilithium decreases the network's bandwidth requirement by 7% due to its smaller cumulative size for cryptographic elements. The execution time for the initial use case scenario, involving the generation of public and private keys, was also dramatically reduced by 87%. Given these benefits, Dilithium emerges as a more efficient and scalable choice for enhancing the security and performance of IOTA-based networks.

## 5. Conclusions

Dilithium and Falcon rely on lattice-based cryptography, a domain with substantial theoretical grounding for resistance against quantum attacks. Sphincs+ takes a different approach, opting for hash-based mechanisms that minimize the algorithm's security assumptions, providing a more easily quantifiable security posture.

Their computational efficiency was benchmarked across ARM and x86-64 platforms, revealing distinct performance characteristics. Dilithium consistently excelled in computational efficiency across both platforms, which makes it a prime candidate for systems where processing power is limited. Falcon, on the other hand, stood out in signature verification speed, despite requiring more computational resources for key generation. Sphincs+ remains the most computationally demanding algorithm, yet it offers the flexibility to trade off between computational efficiency and security parameters.

Energy consumption evaluations further emphasized Dilithium's advantage in low-power scenarios, such as IoT devices. Falcon follows closely, with its energy footprint moderated by its efficiency in signature verification, whereas Sphincs+ is generally less energy-efficient due to its higher computational load. In terms of key and signature sizes, Falcon scored well with its compact design, thereby optimizing bandwidth. Dilithium, while generally efficient, suffers from somewhat larger signature sizes, which affects its bandwidth efficiency. Sphincs+, although secure, presents a significant challenge in bandwidth-constrained applications due to its large signatures.

Dilithium appears as a highly versatile and computationally efficient algorithm, particularly appealing for embedded systems and internet of things (IoT) devices, thanks to its low energy consumption and computational requirements across ARM and x86-64 platforms. It also presents adaptability in smart cards and shows potential for strengthening blockchain architectures through hybrid cryptographic solutions. Falcon, despite its more cumbersome key generation, emerges as an ideal fit for high-throughput, low-latency environments due to its excellence in rapid signature verification. This makes it particularly suited for IoT platforms with RAM constraints up to 64 KB and real-time V2V communication systems, though its higher computational costs render it less favorable for smart card deployments.

Sphincs+, on the other hand, brings to the table a high degree of security customizability at the expense of computational speed and energy efficiency. While this trade-off positions it as a viable option for certain IoT configurations, its less efficient performance and large signature size significantly limit its utility in smart cards and make it a less-than-optimal choice for multi-party threshold signature-based blockchain systems.

Furthermore, the computational burden imposed by hashing operations in these algorithms, especially in Sphincs+, opens avenues for further optimization through hardware-accelerated hashing solutions. These insights underscore the necessity of context-specific evaluations for algorithm selection, given the trade-offs between computational efficiency, energy consumption, and security.

Overall, our study offers insights for the implementation and integration of post-quantum cryptographic algorithms into existing systems and highlights the importance of interdisciplinary expertise in quantum computing, cybersecurity, and algorithmic theory.

Conclusively, a comparative evaluation across key metrics revealed the following features:

- Security level: we compared public key and signature sizes, showing that Sphincs+ offers the smallest public key size, beneficial for storage-limited devices.
- Performance speed: Dilithium leads in key generation and signing speed, while Falcon is optimal for fast signature verification.
- Computational efficiency: this paper outlines the potential for hardware-accelerated optimizations and the significant computational demands of Sphincs+ in comparison to the other two algorithms.
- Applicability on constrained platforms: Dilithium and Falcon are more suitable for devices like IoT sensors, whereas Sphincs+'s larger signature size may hinder its use in the most resource-limited environments.

In summary, each algorithm has distinct advantages, and the choice depends on the specific requirements of the deployment context.

In the future, there will be a significant push to integrate post-quantum cryptographic algorithms into existing communication protocols and security infrastructures. Consequently, resource-constrained devices, such as those analyzed in this paper, will require lightweight cryptographic solutions due to their limited resources. Research into efficient algorithms that are both quantum-resistant and suitable for low-power, low-compute environments is likely to increase. Thereby, considering smart cards, the weakest device type analyzed, the emerging trend of substituting them with more resourceful mobile devices aligns with the need for stronger, quantum-resistant cryptographic solutions.

From the authors' personal viewpoint, a paradigm shift in research and development (for example, changes in hardware and/or materials as an approach to optimizing algorithms) is bound to occur when the development of quantum computers will pose a tangible threat to the usual algorithms. Predicting this shift is a challenging task and falls outside the scope of this paper.

**Author Contributions:** Conceptualization, M.V. and K.M.; methodology, K.M.; validation, M.V.; formal analysis, K.M.; investigation, K.M.; resources, M.V.; data curation, M.V.; writing—original draft preparation, K.M.; writing—review and editing, M.V.; visualization, K.M.; supervision, M.V.; project administration, M.V.; funding acquisition, M.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was supported in part by the European Union, through the European Regional Development Fund, under Grant KK.01.2.1.02.0294. The findings presented in the paper are the sole responsibility of the authors and do not necessarily reflect the official position of the European Union or the European Commission.

**Conflicts of Interest:** K.M. is working at FERIT as a full professor, but K.M. also has a private company (Random Red Ltd.). K.M. (as CEO of Random Red Ltd.) wrote this paper together with M.V. (from company Sedmi Odjel Ltd.). Sedmi Odjel Ltd. has a grant from the European Union. However, the findings presented in the paper are the sole responsibility of the authors and do not necessarily reflect the official position of the European Union or the European Commission. K.M. personally doesn't have any connection to the grant, neither his faculty (FERIT) nor his company (Random Red Ltd.). Sedmi Odjel Ltd. is funded as a legal entity (as reported in "Funding" section). M.V. is not funded personally ("as an author").

## References

1. Cutugno, M.; Giani, A.; Alsing, P.M.; Wessing, L.; Schnore, A. Quantum Computing Approaches for Mission Covering Optimization. *Algorithms* **2022**, *15*, 224. [CrossRef]
2. Hadfield, S.; Wang, Z.; O'gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms* **2019**, *12*, 34. [CrossRef]
3. NIST. Post-Quantum Cryptography. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography (accessed on 17 September 2023).
4. NIST. Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms. Available online: https://csrc.nist.gov/news/2016/public-key-post-quantum-cryptographic-algorithms (accessed on 17 September 2023).
5. NIST. Post-Quantum Cryptography—Call for Proposals. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals (accessed on 17 September 2023).
6. NIST. Post-Quantum Cryptography—Round 1 Submissions. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions (accessed on 17 September 2023).
7. NIST. Post-Quantum Cryptography—Round 2 Submissions. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions (accessed on 17 September 2023).
8. NIST. Post-Quantum Cryptography—Round 3 Submissions. 11 September 2023. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions (accessed on 17 September 2023).
9. Pornin, T. New Efficient, Constant-Time Implementations of Falcon. Cryptology ePrint Archive 2019, Paper 2019/893. Available online: https://eprint.iacr.org/2019/893 (accessed on 6 November 2023).
10. NIST. SP 800-56C Rev. 2—Recommendation for Key-Derivation Methods in Key-Establishment Schemes. Available online: https://csrc.nist.gov/publications/detail/sp/800-56c/rev-2/final (accessed on 8 May 2022).
11. NIST. Module-Lattice-Based Digital Signature Standard. 24 August 2023. Available online: https://csrc.nist.gov/pubs/fips/204/ipd (accessed on 17 September 2023).
12. NIST. Stateless Hash-Based Digital Signature Standard. 24 August 2023. Available online: https://csrc.nist.gov/pubs/fips/205/ipd (accessed on 17 September 2023).
13. NIST. NIST to Standardize Encryption Algorithms That Can Resist Attack by Quantum Computers. 24 August 2023. Available online: https://www.nist.gov/news-events/news/2023/08/nist-standardize-encryption-algorithms-can-resist-attack-quantum-computers (accessed on 17 September 2023).
14. CRYSTALS Team. CRYSTALS-Dilithium—Cryptographic Suite for Algebraic Lattices. Available online: https://pq-crystals.org/dilithium/index.shtml (accessed on 17 September 2023).
15. Lyubashevsky, V. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Advances in Cryptology—ASIACRYPT 2009*; Springer: Berlin, Heidelberg, 2009.
16. Lyubashevsky, V. Lattice Signatures Without Trapdoors. In *Advances in Cryptology—EUROCRYPT 2012*; Springer: Berlin, Heidelberg, 2012; Volume 7237, pp. 738–755.
17. Guneysu, T.; Lyubashevsky, V.; Poppelmann, T. Practical lattice-based cryptography: A signature scheme for embedded systems. In *Cryptographic Hardware and Embedded Systems—CHES 2012*; Springer: Berlin, Heidelberg, 2012; Volume 7428, pp. 530–547.
18. Bai, S.; Galbraith, S. An improved compression technique for signatures based. In *CT-RSA, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8366, pp. 28–47.

19. Bai, S.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Damien, S. CRYSTALS-Dilithium–Algorithm Specifications and Supporting Documentation, 2020. Available online: https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Dilithium-Round3.zip (accessed on 6 November 2023).
20. Lyubashevsky, V. CRYSTALS-Dilithium Presentation at Third PQC Standardization Conference—Session I Welcome/Candidate Updates. NIST, 2021. Available online: https://csrc.nist.gov/presentations/2021/crystals-dilithium-round-3-presentation (accessed on 6 November 2023).
21. Greconici, D.O.C.; Kannwischer, M.J.; Sprenkels, D. Compact Dilithium Implementations on Cortex-M3 and Cortex-M4. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *1*, 1–24.
22. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 17–20 May 2008.
23. Fouque, P.-A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. 2020. Available online: https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Falcon-Round3.zip (accessed on 6 November 2023).
24. Prest, T. Falcon Presentation at Third PQC Standardization Conference—Session I Welcome/Candidate Updates. 2021. Available online: https://www.nist.gov/video/third-pqc-standardization-conference-session-i-welcomecandidate-updates (accessed on 6 November 2023).
25. Bernstein, D.J.; Hopwood, D.; Hülsing, A.; Lange, T.; Niederhagen, R.; Papachristodoulou, L.; Schneider, M.; Schwabe, P.; Wilcox-O'Hearn, Z. SPHINCS: Practical stateless hash-based signatures. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 26–30 April 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 368–397.
26. Ding, J.; Chen, M.-S.; Kannwischer, M.; Patarin, J.; Petzoldt, A.; Schmidt, D.; Yang, B.-Y. Rainbow—Algorithm Specification and Documentation; 2020. Available online: https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Rainbow-Round3.zip (accessed on 6 November 2023).
27. Yokubov, B.; Gan, L. Comprehensive Comparison of Post-Quantum Digital Signature Schemes in Blockchain. In Proceedings of the 2021 IEEE International Conference on Electronic Communications, Internet of Things and Big Data, Yilan County, Taiwan, 10–12 December 2021.
28. Soni, D.; Basu, K.; Nabeel, M.; Karri, R. A Hardware Evaluation Study of NIST Post-Quantum Cryptographic Signature schemes. In Proceedings of the 2nd NIST PQC Standardization Conference, Santa Barbara, CA, USA, 22–24 August 2019.
29. Ortega, K.D.; Perez, L.J.D. Implementing CRYSTAL-Dilithium on FRDM-K64. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, New York, NY, USA, 1–4 December 2021.
30. Beckwith, L.; Nguyen, D.T.; Gaj, K. High-Performance Hardware Implementation of CRYSTALS-Dilithium. In Proceedings of the 2021 International Conference on Field-Programmable Technology (ICFPT), Auckland, New Zealand, 6–10 December 2021.
31. Zhao, C.; Zhang, N.; Wang, H.; Yang, B.; Zhu, W.; Li, Z.; Zhu, M.; Yin, S.; Wei, S.; Liu, L. A Compact and High-Performance Hardware Architecture for CRYSTALS-Dilithium. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**, *2022*, 270–295. [CrossRef]
32. Becker, H.; Hwang, V.; Kannwischer, M.J.; Yang, B.-Y. *Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1*; IACR Transactions on Cryptographic Hardware and Embedded Systems; 2022; Volume 2022, pp. 221–244. [CrossRef]
33. Bradbury, J.; Hess, B. Fast Quantum-Safe Cryptography on IBM Z. In Proceedings of the 3rd NIST PQC Standardization Conference, Virtual, 7–9 June 2021.
34. Kim, Y.; Song, J.; Seo, S.C. Accelerating Falcon on ARMv8. *IEEE Access* **2022**, *10*, 44446–44460. [CrossRef]
35. Amiet, D.; Leuenberger, L.; Curiger, A.; Zbinden, P. FPGA-based SPHINCS+ Implementations: Mind the Glitch. In Proceedings of the 2020 23rd Euromicro Conference on Digital System Design (DSD), Kranj, Slovenia, 26–28 August 2020.
36. Hülsing, A.; Rijneveld, J.; Schwabe, P. ARMed SPHINCS. In *PKC 2016*; Springer: Berlin/Heidelberg, German, 2016; pp. 446–470.
37. Kannwischer, M.J.; Rijneveld, J.; Schwabe, P.; Stoffelen, K. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. In Proceedings of the NIST 3rd PQC Standardization Conference, Virtual, 7–9 June 2021.
38. Roma, C.A.; Tai, C.-E.A.; Hasan, M.A. Energy Efficiency Analysis of Post-Quantum Cryptographic Algorithms. *IEEE Access* **2021**, *9*, 71295–71317. [CrossRef]
39. Dimopoulos, C.; Fournaris, A.P.; Zhao, R.K.; Sakzad, A.; Steinfeld, R. Energy Consumption Evaluation of Post-Quantum TLS 1.3 for Resource-Constrained Embedded Devices. In Proceedings of the 20th ACM International Conference on Computing Frontiers, Bologna, Italy, 9–11 May 2023.
40. Li, X.; Lu, J.; Liu, D.; Li, A.; Yang, S.; Huang, T. A High Speed Post-Quantum Crypto-Processor for Crystals-Dilithium. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *1*, 1. [CrossRef]
41. Aikata, A.; Mert, A.C.; Jacquemin, D.; Das, A.; Matthews, D.; Ghosh, S.; Roy, S.S. A Unified Cryptoprocessor for Lattice-Based Signature and Key-Exchange. *IEEE Trans. Comput.* **2022**, *72*, 1568–1580. [CrossRef]
42. Campbell, D.; Rafferty, C.; Khalid, A.; O'Neill, M. Acceleration of Post Quantum Digital Signature Scheme CRYSTALS-Dilithium on Reconfigurable Hardware. In Proceedings of the 2022 32nd International Conference on Field-Programmable Logic and Applications (FPL), Belfast, UK, 29 August–2 September 2022.
43. Pham, T.X.; Duong-Ngoc, P.; Lee, H. An Efficient Unified Polynomial Arithmetic Unit for CRYSTALS-Dilithium. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *1*, 1. [CrossRef]

44. Wang, T.; Zhang, C.; Cao, P.; Gu, D. Efficient Implementation of Dilithium Signature Scheme on FPGA SoC Platform. *IEEE Trans. Very Large Scale Integr. (vlsi) Syst.* **2022**, *30*, 1158–1171. [CrossRef]
45. Pratiwi, N.; Firmansyah, M.R.; Ezerman, M.F. Implementing CRYSTALS Kyber and Dilithium in Intel SGX Secure Enclaves. In Proceedings of the 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs), Bogor, Indonesia, 22–24 August 2023.
46. Zhao, Y.; Kuang, H.; Sun, Y.; Yang, Z.; Chen, C.; Meng, J.; Han, J. Enhancing RISC-V Vector Extension for Efficient Application of Post-Quantum Cryptography. In Proceedings of the 2023 IEEE 34th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Porto, Portugal, 19–21 July 2023.
47. Aikata, A.; Mert, A.C.; Imran, M.; Pagliarini, S.; Roy, S.S. KaLi: A Crystal for Post-Quantum Security Using Kyber and Dilithium. IEEE Trans. *Circuits Syst. I Regul. Pap.* **2022**, *70*, 747–758. [CrossRef]
48. Mandev, R.; Kavun, E.B. Performance Comparison of Post-Quantum Signature Algorithms Through An Android Email Application Plug-in. In Proceedings of the 2023 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Berlin, Germany, 23–25 July 2023.
49. Dai, Y.; Song, Y.; Tian, J.; Wang, Z. High-Throughput Hardware Implementation for Haraka in SPHINCS+. In Proceedings of the 24th International Symposium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 5–7 April 2023.
50. Sim, M.; Eum, S.; Song, G.; Yang, Y.; Kim, W.; Seo, H. K-XMSS and K-SPHINCS+: Enhancing Security in Next-Generation Mobile Communication and Internet Systems with Hash Based Signatures Using Korean Cryptography Algorithms. *Sensors* **2023**, *23*, 7558. [CrossRef] [PubMed]
51. Hülsing, A.; Kudinov, M.; Ronen, E.; Yogev, E. SPHINCS+C: Compressing SPHINCS+ With (Almost) No Cost. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023.
52. NIST. Lightweight Cryptography. 5 September 2023. Available online: https://csrc.nist.gov/Projects/lightweight-cryptography (accessed on 3 November 2023).
53. NIST. Lightweight Cryptography Standardization Process: NIST Selects Ascon. NIST. 7 February 2023. Available online: https://csrc.nist.gov/news/2023/lightweight-cryptography-nist-selects-ascon (accessed on 3 November 2023).
54. Hernández-Álvarez, L.; Pérez, J.B.; Batista, F.; Queiruga-Dios, A. Security Threats and Cryptographic Protocols for Medical Wearables. *Mathematics* **2022**, *10*, 886. [CrossRef]
55. Atkins, D. Requirements for Post-Quantum Cryptography on Embedded Devices for the IoT. In Proceedings of the 3rd NIST PQC Standardization Conference, Virtual, 7–9 June 2021.
56. Gonzales, R.; Hulsing, A.; Kannwischer, M.J.; Kramer, J.; Lange, T.; Stottinger, M.; Waitz, E.; Wiggers, T.; Yang, B.-Y. Verifying Post-Quantum Signatures in 8 kB of RAM. In Proceedings of the 3rd NIST PQC Standardization Conference, Virtual, 7–9 June 2021.
57. Liu, J.; Wen, J.; Zhang, B.; Dong, S.; Tang, B.; Yu, Y. A post quantum secure multi-party collaborative signature with deterability in the Industrial Internet of Things. *Futur. Gener. Comput. Syst.* **2023**, *141*, 663–676. [CrossRef]
58. Land, G.; Sasdrich, P.; Güneysu, T. A Hard Crystal—Implementing Dilithium on Reconfigurable Hardware. In *International Conference on Smart Card Research and Advanced Applications*; Springer International Publishing: Cham, Switzerland, 2021.
59. Berthet, Q.; Upegui, A.; Gantel, L.; Duc, A.; Traverso, G. An Area-Efficient SPHINCS+ Post-Quantum Signature Coprocessor. In Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 17–21 June 2021.
60. Beckwith, L.; Nguyen, D.T.; Gaj, K. Hardware Accelerators for Digital Signature Algorithms Dilithium and FALCON. *IEEE Des. Test* **2023**, *1*, 1.
61. Greuet, A. Smartcard and Post-Quantum Crypto. In Proceedings of the 3rd NIST PQC Standardization Conference, Virtual, 7–9 June 2021.
62. Migliore, V.; Gerard, B.; Tibouchi, M.; Fouque, P.-A. Masking Dilithium: Efficient Implementation. Cryptology ePrint Archive 2019, Paper 2019/394. Available online: https://eprint.iacr.org/2019/394 (accessed on 6 November 2023).
63. Flaherty, N. NXP, IBM Look to Post Quantum Algorithms for Smart Cards. 8 July 2022. Available online: https://www.eenewseurope.com/en/nxp-ibm-look-to-post-quantum-algorithms-for-smart-cards/ (accessed on 17 September 2023).
64. Vakarjuk, J.; Snetkov, N.; Willemson, J. DiLizium: A Two-Party Lattice-Based Signature Scheme. *Entropy* **2021**, *23*, 989. [CrossRef] [PubMed]
65. Shim, K.-A. A Survey on Post-Quantum Public-Key Signature Schemes for Secure Vehicular Communications. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 14025–14042. [CrossRef]
66. Bindel, N.; McCarthy, S.; Rahbari, H.; Twardokus, G. Suitability of 3rd Round Signature Candidates for Vehicle-to-Vehicle Communication. In Proceedings of the 3rd NIST PQC Standardization Conference, Virtual, 7–9 June 2021.
67. Manna, M.L.; Perazzo, P.; Treccozzi, L.; Dini, G. Assessing the Cost of Quantum Security for Automotive Over—The-Air Updates. In Proceedings of the 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 7–9 June 2021.
68. Winkler, D.; Sepúlveda, D.; Cupelli, M.; Olexa, R.; Sepúlveda, J. Quantum secure high performance automotive systems. In Proceedings of the 19th Escar Europe: The World's Leading Automotive Cyber Security Conference, Frankfurt, Germany, 10–11 November 2021.
69. Sepúlveda, J.; Winkler, D. Super Acceleration of Dilithium in MPSoCs Critical Environments. In Proceedings of the 2022 IEEE European Test Symposium (ETS), Barcelona, Spain, 23–27 May 2022.

70. Kim, Y.; Seo, S.C. Signature Split Method for a PQC-DSA Compliant with V2V Communication Standards. *Appl. Sci.* **2023**, *13*, 5874. [CrossRef]
71. Sharma, L.; Mihra, A. Analysis of Crystals-Dilithium for BlockChain Security. In Proceedings of the Second International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 21–23 May 2021.
72. Raavi, M.; Chandramouli, P.; Wuthier, S.; Zhou, X.; Chang, S.-Y. Performance Characterization of Post-Quantum Digital Certificates. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021.
73. Cozzo, D.; Smart, N.P. Sharing the LUOV: Threshold Post-Quantum Signatures. In Proceedings of the 2nd NIST PQC Standardization Conference, Santa Barbara, CA, USA, 22–24 August 2019.
74. Mirtskhulava, L.; Iavich, M.; Razmadze, M.; Gulua, N. Securing Medical Data in 5G and 6G via Multichain Blockchain Technology using Post-Quantum Signatures. In Proceedings of the 2021 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), Kyiv, Ukraine, 29 November–3 December 2021.
75. Thanalakshmi, P.; Rishikhesh, A.; Marceline, J.M.; Joshi, G.P.; Cho, W. A Quantum-Resistant Blockchain System: A Comparative Analysis. *Mathematics* **2023**, *11*, 3947. [CrossRef]
76. Allende, M.; León, D.L.; Cerón, S.; Pareja, A.; Pacheco, E.; Leal, A.; Da Silva, M.; Pardo, A.; Jones, D.; Worrall, D.J.; et al. Quantum-resistance in blockchain networks. *Sci. Rep.* **2023**, *13*, 1–23. [CrossRef]
77. Verma, N.; Kumari, S.; Jain, P. Post Quantum Digital Signature Change in IOTA to Reduce Latency in Internet of Vehicles (IoV) Environments. In Proceedings of the 2022 International Conference on IoT and Blockchain Technology (ICIBT), Ranchi, India, 6–8 May 2022.