symmetry

MDPI

Article

An Analysis of Existing Hash-Based Post-Quantum Signature Schemes

Cristina Maria Pacurar [1,*], Razvan Bocu [1] and Maksim Iavich [2]

[1] Department of Mathematics and Computer Science, Transilvania University of Brasov, 500036 Brasov, Romania; razvan.bocu@unitbv.ro
[2] Department of Computer Science, Caucasus University, 0102 Tbilisi, Georgia; miavich@cu.edu.ge
* Correspondence: cristina.pacurar@unitbv.ro

Abstract: The rapid development of quantum computing poses challenges to the foundations of traditional cryptography. The threats are significant in terms of both asymmetric cryptography (which exposes schemes like RSA and ECC to efficient attacks) and symmetric cryptography, where key sizes must be increased to mitigate these threats. In this paper, we review the evolution of hash-based digital signatures, from early one-time signatures to modern stateless schemes, with an emphasis on their security properties, efficiency, and practical constraints. Moreover, we propose a simple comparative metric that reflects structural symmetry across key parameters such as key size, signature size, and computational cost, enabling a visual clustering of the schemes. We give particular attention to recent developments such as Verkle trees, which preserve symmetric design principles while improving scalability and proof compactness. The study highlights ongoing tradeoffs between stateful and stateless designs and argues for the continued relevance of symmetric cryptographic constructions in building secure, efficient post-quantum systems.

Keywords: post-quantum cryptography; hash functions; hash-based signature schemes; Verkle trees; vector commitments; quantum threats; asymmetric cryptography

Academic Editors: Jie Yang and Sergei Odintsov

Received: 25 April 2025
Revised: 23 May 2025
Accepted: 6 June 2025
Published: 10 June 2025

Citation: Pacurar, C.M.; Bocu, R.; Iavich, M. An Analysis of Existing Hash-Based Post-Quantum Signature Schemes. *Symmetry* **2025**, *17*, 919. https://doi.org/10.3390/sym17060919

1. Introduction

With the rapid development of quantum computing, traditional factorization-based cryptographic systems, such as RSA (see [1]) and discrete logarithm-based systems like ECC (see [2]), have become increasingly vulnerable to attacks. Quantum algorithms threaten these classical systems by drastically reducing the effort needed to break them. Among the main factors influencing the increase in future threats are algorithms such as Shor's algorithm (see [3]), which can efficiently factor large integers and solve discrete logarithmic problems, or Grover's quantum search algorithm (see [4]) that provides a quadratic speed-up over classical search methods.

Thus, the cryptographic community has increasingly turned its attention to post-quantum cryptography (PQC), which aims to secure systems against adversaries equipped with quantum computers. PQC schemes are designed under the assumption that attackers may possess significant quantum computational resources, and thus, it is of utmost importance to have cryptographic primitives that remain secure even when quantum algorithms are applied. For comprehensive overviews of the subject of PQC, see, for example, ref. [5] and the references cited herein.

Among the various PQC schemes, hash-based signature schemes are one of the most promising candidates. For detailed overviews of the topic of hash-based PQC, see [6–8] and the references cited in these works.

In cryptographic systems, there is a key distinction between two types of cryptography: symmetric and asymmetric. Symmetric cryptography is based on the use of the same key for encryption and decryption. Although it is efficient, one challenge is key distribution. On the other hand, asymmetric cryptography uses mathematically linked public and private keys. Hash-based signature schemes belong to the asymmetric category. The security of hash-based signature schemes relies on non-reversible one-way functions to obtain digital signatures from private keys, which can then be verified using public information.

The interest in hash-based signature schemes is tightly linked to their strong security guarantees provided by the one-way nature of cryptographic hash functions. As a result, these schemes are considered resistant to both classical and quantum attacks. Unlike number-theoretic cryptosystems, hash functions are not affected by Shor's algorithm and only experience a quadratic degradation in security due to Grover's algorithm.

Early contributions, such as Lamport's one-time signatures [9], established the basic principles of hash-based signatures, which were later refined with the introduction of the Merkle signature scheme [10] and the Winternitz one-time signature scheme [11].

Although hash-based signature schemes have robust security properties, they are not without limitations. Among the main drawbacks are large key sizes, statefulness issues, and computational inefficiencies. These challenges become especially evident when multiple signatures are required. Thus, enhanced variants were developed, such as XMSS [12] and SPHINCS [13] (with its improved variant SPHINCS+ [14]) to balance efficiency and security while aligning with emerging NIST PQC standardization efforts [15].

Moreover, very recent research has explored alternative approaches that use advanced data structures to overcome these limitations. In particular, Verkle trees and vector commitments have attracted attention as viable alternatives to conventional Merkle trees. Verkle trees offer a compact authenticated data structure capable of significantly reducing proof sizes [16], while vector commitments [17] enable efficient, succinct commitments to large datasets, facilitating secure and scalable verification processes [16,18,19]. These emerging techniques not only address the issues of storage and transmission overhead but also enhance the scalability and overall performance of post-quantum signature schemes.

On the one hand, this paper presents a comprehensive overview of the existing hash-based post-quantum signature schemes. We focus on the security properties, the efficiency, and the applicability in real-world scenarios. Moreover, we consider alternative approaches, such as Verkle trees and vector commitments. We propose a new means of comparing the existing hash-based post-quantum signature schemes and introduce a metric evaluating distances between pairs of schemes. Thus, not only do we provide an overview of the current landscape in PQC, but we also identify promising directions for future research.

Our paper is structured as follows: Section 2 describes the methodology used and defines the research questions; Section 3 contains an overview of the existing hash-based post-quantum signature schemes; Section 4 is one of the main parts of the paper, as it contains a detailed list of the identified hash-based post-quantum signature schemes along with key properties and significant details; Section 5 emphasizes the need for hash-based post-quantum signature schemes to overcome quantum threats; Section 6 discusses the susceptibility to attacks of stateful and stateless hash-based post-quantum signature schemes and some open questions related to these issues; Section 7 proposes a quantitative means of comparison between hash-based post-quantum signature schemes with a distance function based on key size, signature size, and computational cost; Section 8 presents the conclusions of the current paper.

2. Research Methodology

The current survey follows a systematic approach guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology [20] to ensure a transparent and reproducible review process. Specifically, the review process consists of several key phases: defining the research questions, identifying and selecting the relevant literature, and establishing inclusion and exclusion criteria to ensure the quality and relevance of the analyzed studies. This structured methodology ensures a comprehensive and unbiased analysis of existing hash-based post-quantum signature schemes, providing a clear foundation for evaluating their security, efficiency, and practical applicability.

2.1. Research Questions

To ensure a systematic and structured review that aligns with NIST PQC standardization (see [15,21]) and real-world cryptographic needs, as well as being a useful means to identify open problems and future research directions, we define our main research questions as follows:

RQ1: What are the fundamental principles and cryptographic properties of hash-based post-quantum signature schemes, and how do they compare in terms of security and efficiency?

RQ2: What are the advantages and limitations of stateful and stateless hash-based signature schemes?

RQ3: How do hash-based signature schemes compare in terms of key size, signature size, and computational cost?

RQ4: What recent optimizations and improvements have been proposed for hash-based PQC signatures, and what limitations remain despite these advancements?

RQ5: What are the real-world challenges in implementing hash-based PQC signatures?

RQ6: Given the challenges of existing schemes, what alternative approaches, such as Verkle trees and vector commitments, offer improved scalability, security, and efficiency for post-quantum signatures?

The rest of our paper is structured to answer the specific research questions identified in a dedicated section. Thus, in Section 3 we intend to answer RQ1 by reviewing the existing hash-based post-quantum signature schemes and their significance, motivated by the need to establish a theoretical foundation for hash-based signatures. Section 4 is concerned mainly with RQ2 and RQ3, motivated by the need to differentiate between stateful vs. stateless schemes in terms of security, efficiency, and practicality, and it provides an overview of the key features and limitations of the identified schemes. Section 5 concerns RQ4, as it ensures the significance of the current literature review. Section 6 is dedicated to RQ5 and focuses on real-world implementation challenges and the peril of attacks on hash-based post-quantum signature schemes. Section 7 identifies open research problems and future research directions and is related to RQ6.

2.2. Research Process

In this study, we surveyed a broad range of references across different databases, academic search engines and open-access repositories (see Table 1), to ensure that our review would be comprehensive and aligned with the study's objectives. These sources ensured that the collected literature was comprehensive and aligned with the study's research objectives.

Table 1. Preferred scientific databases, academic search engines, and open-access repositories.

Type	Database/Search Engine	Motivation and Link
Digital Libraries (DL)	IEEE Xplore	High-impact cryptographic research, including PQC. https://ieeexplore.ieee.org
	ACM Digital Library	Comprehensive computing and cryptography papers. https://dl.acm.org
	SpringerLink	Covers cryptographic security, mathematical foundations, and PQC research. https://link.springer.com
	ScienceDirect (Elsevier)	Publishes cryptographic algorithm research and theoretical advancements. https://www.sciencedirect.com
	Wiley Online Library	Contains highly cited cryptography and security research. https://onlinelibrary.wiley.com
Academic Search Engines (SE)	Web of Science	Helps identify highly cited cryptography and PQC papers. https://www.webofscience.com
	Google Scholar	Has the broadest academic coverage for recent and relevant research. https://scholar.google.com
	Scopus	High-quality indexing of cryptographic research with citation analysis. https://www.scopus.com
	DBLP	Specialized in computer science papers. https://dblp.uni-trier.de
	Semantic Scholar	AI-powered academic search engine with citation recommendations. https://www.semanticscholar.org
Open-Access Repositories (OAR)	arXiv.org	Key source for cryptography preprints and emerging PQC research. https://arxiv.org
	Cryptology ePrint Archive	Open-access preprints in cryptography, hosted by IACR. https://eprint.iacr.org
	HAL	French open-access repository, including cryptography and mathematics research. https://hal.archives-ouvertes.fr

2.3. Exclusion and Inclusion Criteria

The relevance of the articles reviewed and, by extension, the scientific rigor of this survey are based on a set of inclusion criteria (IC) and exclusion criteria (EC). The filtering process based on IC and EC follows a structured approach, which is outlined as follows:

- **Step 1: Abstract-Based Filtering**—Articles that are irrelevant, based on their abstracts and keywords, are excluded. More precisely, only those that meet at least 50% of the relevance threshold are considered.
- **Step 2: Full-Text Screening**—Papers that address only a marginal aspect of the scope of this research, as determined by abstract and keywords, are excluded.
- **Step 3: Quality Assessment**—The remaining articles undergo an additional filtering step, where the exclusion applies if at least one of the following conditions is not met:
  - The paper provides a formal security analysis that includes resistance to quantum attacks.
  - The study includes experimental validation, benchmark comparisons, or practical implementation details.
  - The research aligns with NIST PQC standardization efforts or widely accepted cryptographic frameworks.

To ensure comprehensive coverage of high-quality research, we searched for papers from major cryptographic conferences such as IACR's EUROCRYPT, ASIACRYPT, IN-

DOCRYPT, and LATINCRYPT. These conferences are widely regarded as a ground for cutting-edge research in cryptography, thus being leading platforms for cryptographic research. These conferences have significantly contributed to advancements in post-quantum signature schemes, so their inclusion ensures that our analysis covers the most significant work in the area.

A detailed breakdown of the inclusion and exclusion criteria is provided in Tables 2 and 3, respectively.

Table 2. Inclusion criteria for selected papers.

ID	Inclusion Criterion
IC1	Papers that focus on hash-based post-quantum signature schemes.
IC2	Studies published in peer-reviewed journals, conferences, or high-impact cryptographic repositories (e.g., IACR, IEEE, ACM) and indexed in recognized academic databases such as Scopus or Web of Science.
IC3	Research evaluating the security, efficiency, and feasibility of hash-based signatures in post-quantum cryptography.
IC4	Papers discussing practical implementations, optimizations, or hybrid approaches for hash-based signatures.
IC5	Studies comparing hash-based signatures with other post-quantum signature schemes (e.g., lattice-based, code-based).
IC6	Papers proposing novel improvements to hash-based signature schemes or addressing their limitations.
IC7	Papers that provide experimental or theoretical security analysis of hash-based signatures against quantum and classical attacks.
IC8	Papers that have been cited frequently or have significant relevance in post-quantum cryptographic standardization efforts (e.g., NIST PQC).

Table 3. Exclusion criteria for selected papers.

ID	Exclusion Criterion
EC1	Papers that do not focus on hash-based post-quantum signature schemes (e.g., RSA, ECC, or generic cryptography papers).
EC2	Studies that lack technical or theoretical depth, such as opinion articles, editorials, or blog posts.
EC3	Research that does not provide security or performance evaluations of hash-based signatures.
EC4	Papers that discuss outdated or deprecated hash-based signature schemes with no relevance to modern post-quantum cryptography.
EC5	Publications with insufficient experimental results, unverifiable claims, or a lack of mathematical proof for their proposed schemes.
EC6	Non-English articles, unless they contain significant contributions and can be reliably translated.
EC7	Preprints that lack peer review and show insufficient methodological rigor, clarity, or completeness based on objective assessment criteria.
EC8	Duplicates of already included studies, unless they provide new experimental results or improvements.

The subsequent sections offer an in-depth analysis of the selected articles, following this systematic review framework.

3. Overview of Existing Hash-Based Post-Quantum Signature Schemes and Their Importance in Cryptography

The first hash-based digital signatures are attributed to Lamport's one-time signatures (see [9]). Lamport's one-time signature scheme (LOTS) introduced a secure hash-based approach to digital signatures that relies on the one-way nature of cryptographic hash functions for security. Given a message of n bits, the signer generates two random secret values $(s_{i,0}, s_{i,1})$ for each bit m_i of the following message:

$$\text{Secret key:} \quad \{(s_{i,0}, s_{i,1}) \mid i = 1, \ldots, n\}. \tag{1}$$

The corresponding public key consists of hashes of the following values:

$$\text{Public key:} \quad \{(H(s_{i,0}), H(s_{i,1})) \mid i = 1, \ldots, n\}.$$

To sign a message $m = (m_1, m_2, \ldots, m_n)$, the signer reveals the corresponding secret values:

$$\text{Signature:} \quad \{s_{i,m_i} \mid i = 1, \ldots, n\},$$

and verification is carried out by checking whether

$$H(s_{i,m_i}) = \text{Public key entry for } m_i, \quad \forall i.$$

Although this approach is secure under the assumption that the hash function H is preimage-resistant, it has a major limitation: each key pair can only be used to sign a single message. Reusing a key would expose parts of the secret key, and an attacker can learn both $s_{i,0}$ and $s_{i,1}$ for some indices, allowing forgery.

Another OTS scheme was independently developed by Merkle (see [11]) and Winternitz (see [10]), which is called Winternitz-OTS (WOTS). The main advantage of WOTS over LOTS is efficiency, as it reduces the size of the signature while maintaining security. Instead of signing each bit individually, WOTS groups multiple bits together, reducing the number of key pairs required. The scheme introduces a parameter w, which determines the number of bits considered at a time. Given a message represented as a base w integer vector $m = (m_1, m_2, \ldots, m_l)$, the signer generates a sequence of secret values defined as follows:

$$\text{Secret key:} \quad s_i \mid i = 1, \ldots, l. \tag{2}$$

The public key consists of iterated hash values:

$$\text{Public key:} \quad H^w(s_i) \mid i = 1, \ldots, l.$$

To sign a message, the signer computes intermediate hash values corresponding to each digit m_i:

$$\text{Signature:} \quad H^{m_i}(s_i) \mid i = 1, \ldots, l.$$

Verification is performed by checking whether

$$H^{w-m_i}(\text{Signature entry for } i) = \text{Public key entry for } i, \quad \forall i.$$

The WOTS scheme reduces the number of key pairs required while maintaining the security properties of LOTS. However, it remains an OTS scheme, which means that reusing the key pair for multiple messages compromises security.

To overcome this limitation and enable multiple signatures, Merkle proposed a tree-based approach in the Merkle Signature Scheme (MSS) [10], which allows a single public

key to authenticate many one-time signatures while maintaining security. In the MSS, the signer first generates 2^h one-time key pairs and arranges their public keys at the leaves of a binary tree. Each non-leaf node is computed as

$$N_j = H(N_{2j} \parallel N_{2j+1}), \tag{3}$$

where N_j is the hash of the concatenated values of its two child nodes. The root node N_0 serves as the public key for the scheme. A Merkle tree is represented in Figure 1. The Merkle tree structure exhibits a topological symmetry, where each non-leaf node derives from the symmetric combination (hashing) of its two child nodes. This recursive structure ensures that the verification paths are balanced and that the integrity checks are efficient.

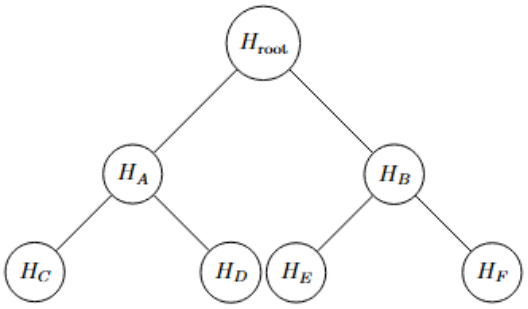

Figure 1. Merkle tree where each non-leaf node is computed as the hash of its two children. The leaves represent hash values of one-time public keys, with the root serving as the overall public key.

To sign a message, the signer first selects an unused OTS key pair, signs the message using LOTS or WOTS, and provides an authentication path proving that the selected key belongs to the Merkle tree. This authentication path consists of h additional hashes from the sibling nodes along the path from the leaf to the root. The verifier reconstructs the tree using these hashes and checks whether the computed root matches the public key.

However, the major drawback of the MSS is the statefulness of its implementation, which requires careful tracking of used keys to prevent reuse and potential security vulnerabilities.

To further enhance usability and efficiency, schemes such as the eXtended Merkle Signature Scheme (XMSS) [12] and WOTS+ [22] have been developed. The XMSS reduces signature sizes by optimizing the tree structure and reusing certain computational results while also addressing some state management issues. However, its drawbacks are computational cost and statefulness. WOTS+ offers a refined balance between security and efficiency by reducing the total number of hash operations required during signature generation.

To address the statefulness limitation, stateless schemes were developed. SPHINCS [13] and its successor, SPHINCS+ [14,23], represent a major leap in hash-based signature design by providing stateless alternatives. Stateless schemes eliminate the need to track the keys that are used. Thus, these schemes offer a simpler implementation in distributed or resource-constrained environments. However, this advantage comes at the cost of larger signature sizes compared to their stateful counterparts, although ongoing research continues to optimize these parameters. SPHINCS+ sacrifices some efficiency: its signatures are tens of kilobytes, significantly larger than XMSS, but it removes the risk of state misuse. Recent research (e.g., [24]) has begun to compress SPHINCS+ signatures without reintroducing state.

Traditionally, hash-based schemes rely on Merkle trees for their simplicity and security. However, as systems scale, the size of the proofs grows logarithmically, which is a significant limitation, especially in environments where storage and communication efficiency are paramount.

To address these challenges, very recent research has focused on alternative data structures that offer more compact proofs and enhanced scalability. Thus, Verkle trees (see [25]) and vector commitments were proposed in [16]. Verkle trees are an improvement of Merkle trees that offer more compact proofs and lower communication overhead. Instead of relying on cryptographic hash algorithms, the Verkle tree technique uses vector commitments to construct a Merkle tree. The process begins by selecting k pieces, followed by computing a Verkle tree using files $f_0, f_1, \ldots, f_n$. Then, it verifies whether the membership of each file in a subset provides proof of a specific binding position P_i relative to the vector commitment VC by performing calculations for each subset. This process continues iteratively until the root commitment is established, computing vector commitments throughout the tree.

A Verkle tree replaces hash-based commitments with vector commitments, allowing for more efficient proofs. Verkle trees maintain a symmetric commitment structure through the consistent application of vector commitments across branches, which makes the proofs more compact and uniform. This design provides more compact proofs compared to traditional Merkle trees. A Verkle tree is depicted in Figure 2. The commitment at each node aggregates all child commitments as follows:

$$V_{\text{parent}} = \text{Commit}(V_1, V_2, \ldots, V_n)$$

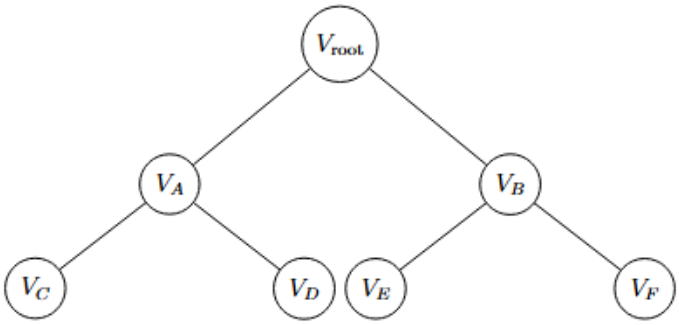

Figure 2. Verkle tree where internal nodes hold vector commitments (e.g., V = Commit ({child commitments})).

By using polynomial commitments, Verkle trees can significantly reduce the size of authentication paths, making them particularly attractive for systems where storage and transmission costs are critical. Vector commitments allow for efficient, succinct commitments to large datasets while supporting dynamic updates and verifications. Their integration into hash-based signature schemes can improve scalability and reduce verification times, thus improving overall system performance.

Unlike many alternative PQC approaches (e.g., lattice-based or multivariate schemes), hash-based schemes do not depend on complex mathematical structures that may be vulnerable to unforeseen attacks. Their proven resilience against quantum adversaries, coupled with ongoing optimizations and the incorporation of novel data structures, underscores their continued relevance and importance in the cryptographic landscape.

4. Classification of Identified Models

In Table 4, we introduce the main identified schemes related to hash-based PQC. This table serves as an efficient reference tool, enabling researchers to quickly compare various schemes and evaluate their performance, strengths, and weaknesses. The schemes are listed in chronological order of their appearance in the literature to better illustrate the development timeline of hash-based signatures. The table offers a comprehensive overview that simplifies decision making when selecting the most suitable scheme for specific PQC applications. Furthermore, the table highlights the efficiency, statefulness, and applicability

of each scheme. This helps practitioners assess which scheme best aligns with the security and performance requirements of a given application.

Table 4. Chronological comparison of hash-based signature schemes.

Scheme	Year	Key Features	Limitations	Efficiency	Statefulness	Applicability in PQC
Lamport OTS (LOTS) [9]	1979	Simple, quantum-secure	Large key and signature size	Low (single-use key, size)	Stateless	Impractical due to size
Merkle Signature Scheme (MSS) [10]	1979	Tree-based OTS	Large signature sizes and expensive key generation	Improved compared to LOTS (reuses OTS keys)	Stateful	Basis for modern PQC schemes
Winternitz OTS (WOTS) [11]	1989	Smaller than Lamport	Slower signing process	More efficient verification (fewer hashes)	Stateless	Single-use limitation
Leighton–Micali Signatures (LMSs) [26]	1995	Variant of Merkle's scheme supporting a multilevel structure for many signatures	Signature size and verification time increase with each additional level	Efficient (tree-based)	Stateful	NIST-approved alternative to XMSS; standardized for PQC (RFC 8554)
BiBa [27]	2001	Few-time scheme (FTS) with fast verification; efficient for broadcasts	Expensive signing, large public keys	Inefficient	Stateful	Not well suited for PQC
Hash to Obtain Random Subset (HORS) [28]	2002	Few-time signature scheme (FTS) with many public keys	Vulnerable to subset resilience attacks	Moderate	Stateless	Efficient but impractical when used alone
Compressed MSS (CMSS) [29]	2006	Hierarchical structure of multiple smaller Merkle trees; OTS	Limited signatures	Better than MSS but still computationally heavy	Stateful	More practical, not widely used
Generalized MSS (GMSS) [30]	2006	Variant of the Merkle OTS which allows signing cryptographically unlimited number of messages by using hyper-tree; generalization of CMSS	As secure as the collision resistance of the underlying hash function	Better than CMSS	Stateful	Secure for PQc
eXtended Merkle Signature Scheme (XMSS) [12]	2011	25% smaller than previous best hash-based schemes; OTS	Computational cost and limited number of signatures per key pair	Efficient if hash functions are efficient	Stateful	Yes, post-quantum secure; standardized for PQC (RFC 8391)
Hash to Obtain Random Subset and Integer Composition (HORSIC) [31]	2012	One-time signature scheme with smaller signature and key size	Higher overhead in key generation and signature verification	Efficient	Stateless	Used for for broadcast authentication in wireless sensor networks
WOTS+ [22]	2013	Improves WOTS and reduces signature size	One-time use; requires additional authentication (e.g., Merkle trees) for key management	More efficient than WOTS	Stateless	Used in modern PQC schemes
Multitree XMSS (XMSS-MT) [32]	2013	Supports more signatures; better scalability than XMSS	Increases signature size	Higher than XMSS	Stateful	Improves performance for long-term keys; strong PQC candidate; NIST-recommended
SPHINCS [13]	2015	First stateless hash-based signature (hybrid approach using multiple layers of one-time signatures and Merkle trees, with HORST as a component)	Large signature sizes	Low efficiency	Stateless	Baseline stateless scheme; not standardized, but concept proven

Table 4. *Cont.*

Scheme	Year	Key Features	Limitations	Efficiency	Statefulness	Applicability in PQC
Haraka [33]	2016	Short-Input Hashing	Not collision-resistant	Highly efficient in hardware	Stateless	Not a general-purpose PQC hash function, as it lacks collision resistance; used in SPHINCS+
Forest of Random Subsets (FORS) [14,23]	2017	Improvement of HORS; FTS	Increased signature size	More efficient than HORS	Stateless	Used in SPHINCS+
SPHINCS+ [14,23]	2017	Stateless, improved efficiency, flexible	Large (but improved)	Better than SPHINCS; uses tweakable hash (Haraka) and WOTS+	Stateless	NIST PQC Round 3 winner
Sphincs-simpira [34,35]	2017	Security-similar original SPHINCS algorithm with faster key pair generation	Larger signature sizes compared to classical schemes	Improved performance	Stateless	Suitable for post-quantum applications requiring efficient hash-based signatures
PORS [36,37]	2017	Improved HORS; avoids weak messages	Requires a pseudorandom generator (PRG) for subset selection, adding computational overhead	Similar to HORS; slightly better	Stateless	Used for Gravity SPHINCS
Gravity SPHINCS [36,37]	2017	Improvement of SPHINCS based on PORS	Larger signature sizes and higher computational demands	Offers a balance between security and performance; optimized implementations available	Stateless	Submitted to NIST's Post-Quantum Cryptography Project; relevant for post-quantum secure applications
Sphincs Streebog [38]	2019	Uses Streeborg hash-function	Depends on Streebog's security assumptions	Dependent on Streebog's performance characteristics	Stateless	Applicable in contexts where Streebog is a standard or preferred hash function
HORSIC+ (Hash to Obtain Random Subset and Integer Composition) [39]	2021	Improved HORSIC	Higher key storage requirements	More efficient than HORSIC	Stateless	Better candidate for PQC than HORSIC
SPHINCS-α [40]	2022	Improvement performance using improved WOTS and improved FORS	Increased key size due to constant-sum encoding, impacting storage requirements	Improved signing and verification times over SPHINCS+	Stateless	Enhances SPHINCS+; suitable for PQC applications
K-SPHINCS+ [41]	2022	Uses Korean hash functions such as LSH, CHAM, and LEA	Performance depends on the efficiency of the integrated hash functions	Comparable efficiency with potential for optimization using advanced techniques	Stateless	Relevant for regions or applications where Korean cryptographic standards are preferred
SPHINCS+C [24]	2023	Compresses SPHINCS+ signatures with minimal computational overhead	Potential tradeoffs between compression ratio and computational cost	Achieves smaller signature sizes with negligible impact on performance	Stateless	Suitable for applications requiring reduced signature sizes without sacrificing efficiency
Verkle tree and vector commitments [16,42]	2023	Use vector commitments to build the Merkle tree	New approach; use vector commitments	Efficient	Stateless	Not PQS-secure if based on RSA assumption [16]; PQS-secure if Verkle trees are used in signature procedures and lattices are used to create vector commitments [42]

Table 4. *Cont.*

Scheme	Year	Key Features	Limitations	Efficiency	Statefulness	Applicability in PQC
Verkle tree with lattice-based vector commitments [18]	2023	Uses vector commitments to build the Merkle tree	New approach; requires further analysis	Aims to provide post-quantum security leveraging Verkle trees and lattice-based techniques	Stateless	Strong potential
GRASP (GPU-based paRallel Accelerated SPHINCS+) [43]	2024	Accelerates SPHINCS+ using GPU parallel architecture; significant throughput improvements	Requires specialized hardware (GPUs); implementation complexity	Surpasses NIST reference implementation by more then three orders of magnitude	Stateless	Enhances performance of SPHINCS+; applicable in PQC where high throughput is required
Syrga2 [44]	2024	Multiuse signatures with state retention	Larger key sizes; requires careful parameter selection	Efficient signing and verification	Stateful	Suitable for post-quantum cryptography
Maximum Utilization Multiple HORS (MUM-HORS) [45]	2024	Multiple-time usage, PQ security, compact key management	Requires careful implementation to avoid weak-message attacks	Fast signing; efficient for IoT devices	Stateless	Suitable for heterogeneous IoT systems
Verkle tree with lattice-based vector commitments and quantum seed-based pseudorandom generators [19]	2025	Verkle tree-based scheme improving efficiency, reducing memory requirements, enhancing security against quantum attacks	Complexity in implementation; requires quantum randomness sources	Faster verification, reduced memory use, efficient proof sizes	Stateless	Suitable for blockchain, IoT, and mobile security applications

In the context of post-quantum cryptography (PQC), it is essential to distinguish between stateful and stateless schemes, as this can significantly impact the performance and usability of a given hash-based signature scheme. Stateful schemes (such as LMS and XMSS) require tracking of the used schemes to ensure that signatures are not reused or duplicated so that the private key evolves correctly after each signature operation to avoid catastrophic security failures [46]. Although stateful schemes can offer more compact signatures and greater efficiency, they require careful state management to avoid errors that could compromise security.

However, stateless schemes (like SPHINCS and SPHINCS+) eliminate the need for state management, meaning that each signature is independent and does not rely on prior computations. While this makes them resilient against synchronization failures, the cost includes increased signature sizes and slower signing times to achieve the same level of security as stateful schemes [46].

We explicitly indicate whether each scheme is stateful or stateless to facilitate comparison. This helps to choose whether the benefits of smaller, more efficient signatures are worth the stateful nature of the scheme or if the simplicity and lower risk of errors in stateless schemes would be a better fit despite the computational cost. Comparison is crucial for selecting the optimal scheme based on security, efficiency, and implementation complexity.

Among the identified models, we selected 10 widely used and appreciated schemes: LOTS, MSS, WOTS, LMS, HORS, XMSS, WOTS+, SPHINCS, SPHINCS+, and the newly introduced Verkle trees, for which we provide an additional discussion of the security assumptions and give some detailed benchmark details in tabular form in Table 5.

Table 5. Security assumptions and performance characteristics of key hash-based signature schemes.

Scheme	Security Assumptions	Performance Characteristics
LOTS	Collision resistance; second-preimage resistance of the hash function	Sign/Verify: n hash operations (one per message bit); Signature size: $n \cdot \|H\|$; Public key size: $2n \cdot \|H\|$
MSS	Collision resistance; secure Merkle tree authentication paths	Sign/Verify: $O(h)$ hashes for authentication path; Signature size: $h \cdot \|H\|$; KeyGen: 2^h LOTS key pairs
WOTS	One-wayness and collision resistance of hash functions	Sign/Verify: l chains of length w (total $l \cdot w$ hashes); Signature size: $l \cdot \|H\|$, where $l = \lceil n/\log_2 w \rceil + \lceil \log_2 n/\log_2 w \rceil$
LMS	Collision and second-preimage resistance; Merkle tree assumptions	Sign/Verify: $(p+h) \cdot \|H\|$ hash operations: Signature size: $(p+h+1) \cdot \|H\|$ bits; KeyGen: $\|H\|$ bits
HORS	Subset resilience; preimage resistance; few-time use only	Sign/Verify: t hash operations per signature; Signature size: $t \cdot \|H\|$; Public key size: $n \cdot \|H\|$
XMSS	Collision resistance; PRF security; second-preimage resistance	KeyGen: ≈0.5–2 s for $h = 10$; Sign/Verify: ≈10 ms each (BDS-optimized)
WOTS+	Collision and second-preimage resistance with chaining tweaks	Same as WOTS with chaining function enhancements
SPHINCS	Collision resistance; PRF security; multitarget second-preimage resistance	Sign ≈ 15 ms; Verify ≈ 1 ms; Sig size ≈ 41 KB
SPHINCS+	Tweakable hashes; strong collision and second-preimage resistance; PRF security	Sign ≈ 10–100 ms; Verify $<$ 10 ms; Sig size ≈ 49 KB
Verkle Trees	Collision resistance of hash or lattice SIS hardness for vector commitment	Prove/Verify ≈ 5–10 ms; Proof size ≈ 300–600 B

5. Motivation for Post-Quantum Security Due to Quantum Threats

With the rapid development of quantum computing, traditional cryptosystems based on integer factorization or discrete logarithms have become increasingly vulnerable. The main threats come from quantum algorithms such as Shor's algorithm [3], which can efficiently solve these problems, making RSA, ECC, and other number-theoretic schemes insecure. Moreover, Grover's algorithm [4] provides a quadratic speed-up in brute-force attacks against symmetric primitives, which implies a reduction in the effective security margin that cannot be ignored. The threat of quantum attacks affects the security assumptions of both symmetric and asymmetric cryptography. Asymmetric cryptographic systems such as RSA and ECC are under serious threat from Shor's algorithm, which can efficiently solve the underlying mathematical problems they rely on. On the other hand, Grover's algorithm threatens symmetric cryptographic schemes. However, the quadratic speed-up provided by Grover's algorithm can be avoided by doubling key sizes. Thus, in order to remain secure in the quantum context, many existing symmetric schemes will require longer key lengths or more robust constructions. The tradeoffs in performance and resource consumption emphasize why it is imperative to transition to PQC solutions, like hash-based signature schemes, that maintain security even in the presence of quantum adversaries. Even though hash-based signature schemes are asymmetric, they inherit some of the durability of symmetric systems due to their reliance on cryptographic hash functions. This hybrid of asymmetric usage that is built using symmetric primitives contributes to their resilience for post-quantum threats.

Hash-based signature schemes, like those based on Merkle trees and their improved variants, are particularly promising for post-quantum security. One key advantage of hash functions is that they are not susceptible to Shor's algorithm, even though they suffer a moderate impact from Grover's algorithm. Thus, they provide a relatively stable foundation for constructing quantum-resistant signatures. Moreover, the simplicity of their underlying operations, primarily relying on hash computations, provides a clear and robust path to security, even as quantum hardware continues to evolve. This straightforward design minimizes potential vulnerabilities by focusing on well-understood cryptographic primitives, making it easier for researchers and practitioners to analyze, implement, and maintain.

Interestingly, hash-based signature schemes, though asymmetric in nature, inherit some of the durability of symmetric systems due to their reliance on cryptographic hash functions. This hybrid characteristic—asymmetric usage built on top of symmetric primitives—contributes to their resilience in the face of quantum threats.

A promising step forward in the design of hash-based signature schemes is the integration of a modified type of Merkle tree called a Verkle tree [25]. Verkle trees improve on traditional Merkle trees by taking advantage of vector commitments to provide more compact proofs and reduce communication overhead [16]. This advancement offers several critical benefits, such as reduced proof sizes, improved scalability, and verification efficiency. Thus, interest in Verkle trees is increasing.

Iavich et al., in [16], were the first to incorporate Verkle trees into a hash-based signature, initially using RSA-based vector commitments in place of hash functions for node computation. However, RSA-based commitments are inefficient and not post-quantum (since RSA is vulnerable to quantum attacks). Later, in [18], Iavich et al. improved this by using lattice-based vector commitments, which are quantum-safe and more efficient. Compared to traditional stateless renewable systems, which depend on private key configurations and centralized authority to manage public parameters, lattice-based structures offer improved security along with increased compactness and efficiency.

Therefore, Verkle trees seem to be a better approach and a significant improvement over Merkle trees, with the main advantages observed as follows:

1. Reduced proof sizes—Verkle trees can achieve shorter authentication paths (significantly shorter than Merkle trees), thus *decreasing the overall signature size.*
2. Enhanced scalability—Verkle trees are a *more efficient data structure*, making them better for handling larger datasets with lower storage and transmission requirements and particularly advantageous in resource-constrained environments.
3. Improved verification efficiency—Verkle trees enable much faster signature verification, *reducing the computational complexity* associated with verifying long Merkle authentication paths, which is a crucial aspect for high-throughput applications and IoT devices.

As the threat of quantum computing looms over classical cryptographic systems, transitioning to post-quantum secure signature schemes has become mandatory. Among the proposed solutions, stateful hash-based signature schemes have emerged as a viable option due to their simplicity and security, which are well-established cryptographic hash functions rather than number-theoretic problems vulnerable to quantum attacks.

6. Gaps and Open Questions in Identified Research

However, these schemes are not foolproof. The inherent risk of state mismanagement is crucial for stateful schemes, such as XMSS and LMS. Each signature requires precise tracking and updating of one-time keys, and any oversight can lead to key reuse, which can mean a catastrophic security breach.

To mitigate the issues introduced by statefulness, two promising strategies have emerged in the recent literature and implementations therein: distributed state management and hardware-secure enclaves. Distributed state management involves coordinated tracking and synchronization of the cryptographic signing state across multiple nodes or devices to ensure that one-time keys are not reused. This approach is particularly useful in environments such as cloud-based key management services, where fault tolerance and high availability are critical. This is to ensure the signing state's integrity and monotonic progression can be used in techniques like consensus protocols (e.g., Paxos or Raft), cryptographic versioning, and append-only logs. However, the cost of these methods comes with additional complexity, latency, and trust assumptions, especially in adversarial network conditions.

Another alternative, which is an increasingly adopted method, is the use of hardware-secure enclaves, which are a protected region of memory and processing within a device's CPU that enables secure execution of code and data storage that is isolated from the rest of the system, even if the operating system is compromised. Technologies such as Intel SGX or ARM TrustZone provide such enclaves, allowing encapsulation of the key and state within a physically and logically isolated execution environment. These secure enclaves protect against a wide range of software-level attacks and can ensure that the state is neither lost nor tampered with, even in the presence of compromised operating systems. However, relying on specific hardware introduces vendor lock-in and may be unsuitable in many situations.

The NIST Special Publication [47] recommends two standardized stateful hash-based schemes: Leighton–Micali Signatures (LMSs) and the eXtended Merkle Signature Scheme (XMSS), along with their multitree variants. Although these schemes are suitable for applications that require long-term security, a key challenge with stateful schemes is the need to precisely manage the state to prevent the reusage of OTS keys, which can lead to catastrophic security breaches.

In [48], several LOTSs and WOTSs were analyzed under different kinds of two-message attacks. The study shows that LOTS experiences only a gradual decline in security within these attack scenarios, with typical parameters remaining somewhat secure even when subjected to a two-message attack. However, for optimized Lamport and its generalization, WOTS, the security deteriorates at an increasingly rapid rate, making standard parameter choices insufficient to ensure a reasonable level of security under two-message attacks.

However, even though stateless schemes like SPHINCS+ eliminate the burden of state maintenance, thereby reducing the risk associated with improper key handling, the tradeoff includes larger signature sizes and potentially lower performance. Moreover, even with the security risks due to state management being reduced, stateless hash-based post-quantum signature schemes are not immune to attacks. Even for SPHINCS+, which was approved by NIST [49], there have been feasible attacks. For example, the research in [50] analyzed the security of SPHINCS+ and proposed quantum attacksl the paper [51] proposed a generic attack (which does not depend on the underlying hash function used) on SPHINCS, gravity-SPHINCS, and SPHINCS+ to forge any message signature at the cost of a single fault message. More research on attacks on SPHINCS+ can be found, for example, in [52–54].

An open question in the field of hash-based post-quantum signature schemes is finding the best balance between efficiency and security. Although these schemes are promising because they resist quantum attacks using well-understood hash functions, integrating advanced techniques like Verkle trees and vector commitments to reduce signature sizes and computational overhead is a viable alternative and an ongoing challenge.

Another direction of research is exploring the tradeoffs between the two different approaches: stateful schemes, which can produce smaller signatures and offer better performance but require rigorous key management to avoid security risks, and stateless schemes, which simplify key handling at the expense of larger signatures and slower processing. Additionally, a crucial yet unsolved aspect is finding effective ways to mitigate side-channel and fault injection attacks without compromising overall efficiency. Addressing these issues is key to making hash-based post-quantum signature schemes both secure and practical for real-world applications.

In summary, while hash-based signature schemes provide a promising foundation for post-quantum security, several open questions remain, ranging from fault resistance and state management to the optimization of emerging structures such as Verkle trees. Addressing these challenges is crucial for developing robust, scalable, and user-friendly cryptographic solutions that can meet the evolving demands of a quantum-enabled future.

7. Comparative Analysis and Remarks

To systematically and quantitatively compare the various hash-based post-quantum signature schemes, we introduce a quantitative metric that encapsulates three critical performance dimensions: key size (KS), signature size (SS), and computational cost (CC). These parameters are first normalized on a scale from 0 to 1, ensuring that differences in their absolute values can be meaningfully compared across different schemes. Table 6 presents the normalized values for 12 selected schemes, such as LOTS, MSS, WOTS, LMS, HORS, XMSS, WOTS+, XMSS-MT, SPHINCS, SPHINCS+, Gravity SPHINCS, and the newly introduced Verkle trees, highlighting the tradeoffs inherent in each design.

Table 6. Normalized key size, signature size, and computational cost.

Scheme	KS (0–1)	SS (0–1)	CC (0–1)
LOTS	1.0	1.0	0.2
MSS	0.55	0.6	0.65
WOTS	0.6	0.6	0.5
LMS	0.5	0.55	0.6
HORS	0.75	0.85	0.4
XMSS	0.4	0.45	0.7
WOTS+	0.5	0.5	0.5
XMSS-MT	0.35	0.4	0.75
SPHINCS	0.8	0.7	0.75
SPHINCS+	0.75	0.65	0.75
Gravity SPHINCS	0.7	0.8	0.8
Verkle Trees	0.75	0.4	0.65

Let $\mathcal{S}$ denote the set of all hash-based PQC schemes.

Definition 1. *We define the distance function* $d : \mathcal{S} \times \mathcal{S}$ *to* $[0, 1]$ *between two signature schemes* $S_1, S_2 \in \mathcal{S}$ *as*

$$d(S_i, S_j) = w_1|KS_i - KS_j| + w_2|SS_i - SS_j| + w_3|CC_i - CC_j| \quad (4)$$

where

- KS_i *= key size (normalized between 0 and 1);*
- SS_i *= signature size (normalized between 0 and 1);*
- CC_i *= computational cost (normalized between 0 and 1);*
- w_1, w_2, w_3 *= weights assigned to each factor.*

For our comparison, we set the weights based on practical importance, as identified in the previous section:

- **Key size (KS):** $w_1 = 0.4$ (important, since larger key sizes require more storage and transmission bandwidth but are manageable with modern storage capabilities);
- **Signature size (SS):** $w_2 = 0.4$ (crucial for efficiency, as large signatures can lead to increased verification times and network overhead);
- **Computational cost (CC):** $w_3 = 0.2$ (matters but depends on hardware).

Since KS and SS impact storage, transmission, and real-world feasibility, the two were given the same weight of 0.4, while CC, which can be optimized to minimize its impact, was given a 0.2 weight.

We selected 12 widely recognized schemes from the 29 schemes analyzed in Section 4 in Table 4 and normalized their values on a hl0–1 scale based on relative differences.

For better visualization of the similarities in efficiency Figure 3, of the 12 hash-based signature schemes that were selected based on their key size, signature size, and computational cost. This illustrates which schemes are more similar in efficiency and security tradeoffs.

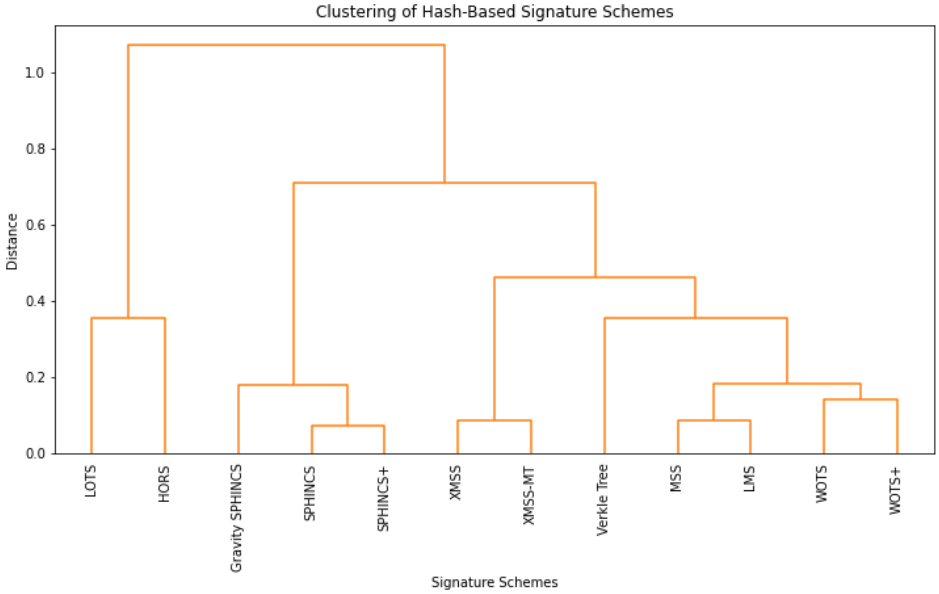

Figure 3. Clustering dendrogram.

Beyond the basic parameters of key size, signature size, and computational cost, our analysis suggests several avenues for further refinement and extension of this framework.

Firstly, our metric could benefit from incorporating additional performance metrics. Although our primary focus has been on KS, SS, and CC, additional factors such as memory consumption, energy efficiency, and even latency under various operational loads might be included. Moreover, the current fixed weights ($w_1 = 0.4$, $w_2 = 0.4$, $w_3 = 0.2$) reflect our assessment of general importance. However, in practice, the relative importance of these parameters may vary depending on the application; thus, a dynamic weighting scheme, possibly determined through multicriteria decision-making methods or even machine learning models, could tailor the metric to specific contexts.

To demonstrate how the choice of the weights (w_1, w_2, w_3) can be adapted to reflect different real-world priorities, we briefly present two illustrative case studies.

Firstly, let us consider the case of IoT sensor networks. One characteristic is that IoT deployments typically operate under severe storage, energy, and bandwidth constraints. Thus, by choosing

$$(w_1, w_2, w_3) = (0.5,\ 0.4,\ 0.1),$$

we place a greater emphasis on minimizing public key sizes (w_1) while keeping signature sizes significant (w_2) and only account 10% for code complexity considerations (w_3). Under

this weighting, the preferred schemes would be XMSS and Gravity-SPHINCS because of their compact key–signature footprints and moderate CPU requirements.

In contrast, we look at blockchain nodes. On-chain storage costs dominate in blockchain systems: each additional byte of signature imposes a permanent ledger bloat. We therefore set

$$(w_1, w_2, w_3) = (0.3,\ 0.6,\ 0.1),$$

allocating 30% to size, 60% to verification/generation speed (critical for high-throughput block production), and 10% to simplicity of implementation. Using these weights, stateless but CPU-heavy schemes like SPHINCS+ are expected to drop, whereas K-SPHINCS+ and GRASP rise because of their favorable speed–size tradeoffs.

As a generalization, we propose another metric and leave open the practical usage of it as an open question.

The linear combination in Equation (4) corresponds to an L_1 norm weighted by w_i. One can generalize this approach by considering the following p-norm:

$$d_p(S_i, S_j) = \left(w_1|KS_i - KS_j|^p + w_2|SS_i - SS_j|^p + w_3|CC_i - CC_j|^p\right)^{\frac{1}{p}}, \quad p \geq 1. \tag{5}$$

For $p = 1$, Equation (5) reduces to our original linear metric, while $p = 2$ gives a weighted Euclidean distance. The choice of p can be optimized depending on the desired sensitivity to outlier differences.

The clustering analysis that we performed based on our distance metric provides a quantitative tool to compare schemes and also offers practical insights for selection. For example, schemes with lower normalized KS and SS values, despite the higher computational cost, may be better suited for environments with strict bandwidth or storage constraints. On the other hand, when computation is a limiting factor, schemes with higher KS or SS values might be acceptable if they offer significantly lower computational overhead. This understanding can guide system architects in selecting the most appropriate scheme for their specific post-quantum security requirements.

Moreover, our proposed distance emphasizes a step forward in the research direction of hash-based PQC, which is provided by Verkle trees and vector commitments. Our metric captures asymmetries and similarities in the tradeoffs among key size, signature size, and computational cost. In some cases, closely clustered schemes show symmetric parameter profiles, suggesting that they have a mutual design basis.

8. Conclusions

The purpose of this paper is to present a comprehensive review and comparative analysis of hash-based post-quantum signature schemes. Throughout the paper, we emphasize both the theoretical and practical implications of these schemes. Our systematic investigation has highlighted that hash-based post-quantum signature schemes represent a viable choice in the face of quantum adversaries. However, some key disadvantages still need to be addressed, such as large key and signature sizes, statefulness issues, and computational overhead.

The main advantage of these schemes is the robust security foundation, which is rooted in the properties of hash-based functions. Unlike number-theoretic approaches, these schemes remain secure even under quantum threats, as hash functions are not affected by Shor's algorithm. Moreover, enhancing these schemes with techniques like Verkle trees and vector commitments can significantly improve the security and efficiency of hash-based post-quantum signature schemes.

One key limitation of hash-based post-quantum signature schemes is the tradeoffs between stateful and stateless approaches. Stateful schemes such as XMSS and LMS offer

efficiency in terms of signature size at the cost of careful key management to prevent catastrophic failures. On the other hand, stateless schemes like SPHINCS+ simplify implementation by eliminating state management, but the price paid is in increased signature sizes and computational costs. An emerging optimization comes from using advanced data structures, particularly Verkle trees, which promise reduced proof size and communication overhead, thus enhancing the system's performance and scalability.

The extended classification provided in Section 4 provides an overview of the trade-offs inherent in existing schemes. Moreover, the metric introduced in Section 7 offers a novel practical framework to evaluate current schemes, as well as future post-quantum signature candidates.

In summary, the evolving landscape of quantum computing requires a continuous refinement of cryptographic techniques. Hash-based post-quantum signature schemes play a crucial role in securing communications in a quantum-enabled future, especially due to their inherent simplicity and strong security guarantees. Continuing the integration of new structures into existing schemes to improve efficiency and security will help ensure a resilient cryptographic infrastructure.

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

Funding: This research has been supported by the NATO Science for Peace and Security (SPS) grant G7394 "Post-quantum Digital Signature using Verkle Trees".

Data Availability Statement: The original contributions presented in the study are included in the article.

Acknowledgments: The authors would like to thank the reviewers and editors for their valuable comments and suggestions, which helped improve the quality and clarity of this manuscript.

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

References

1. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
2. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
3. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS'94), Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
4. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; Miller, G.L., Ed.; ACM: New York, USA, 1996; pp. 212–219.
5. Bernstein, D.J.; Lange, T. Post-quantum cryptography. *Nature* **2017**, *549*, 188–194. [CrossRef] [PubMed]
6. Fathalla, E.; Azab, M. Beyond Classical Cryptography: A Systematic Review of Post-Quantum Hash-Based Signature Schemes, Security, and Optimizations. *IEEE Access* **2024**, *12*, 175969–175987. [CrossRef]
7. Li, L.; Lu, X.; Wang, K. Hash-based signature revisited. *Cybersecurity* **2022**, *5*, 13. [CrossRef]
8. Srivastava, V.; Baksi, A.; Debnath, S.K. An Overview of Hash-Based Signatures. Cryptology ePrint Archive. Paper 2023/411. 2023. Available online: https://eprint.iacr.org/2023/411 (accessed on 12 March 2025).
9. Lamport, L. *Constructing Digital Signatures from a One-Way Function*; Technical Report SRI-CSL-98; SRI International: Menlo Park, CA, USA, 1979.
10. Merkle, R.C. Secrecy, Authentication, and Public-Key Systems. Ph.D. Dissertation, Stanford University, Stanford, CA, USA, 1979.
11. Merkle, R.C. A certified digital signature. In *Advances in Cryptology—CRYPTO'89*; Springer: New York, NY, USA, 1989; pp. 218–238.

12. Buchmann, J.; Dahmen, E.; Hülsing, A. XMSS—A practical forward secure signature scheme based on minimal security assumptions. In Proceedings of the International Workshop on Post-Quantum Cryptography, Taipei, Taiwan, 29 November–2 December 2011; Volume 7071, pp. 117–129.
13. 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 *Advances in Cryptology– EUROCRYPT 2015*; Oswald, E., Fischlin, M., Eds.; Lecture Notes in Computer Science (LNCS); Springer: Berlin/Heidelberg, Germany, 2015; Volume 9056, pp. 368–397.
14. Bernstein, D.J.; Hulsing, A.; Kolbl, S.; Niederhagen, R.; Rijneveld, J.; Schwabe, P. The sphincs+ signature framework. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2129–2146.
15. NIST. 2016. Submission Requirements and Evaluation Criteria for the PostQuantum Cryptography Standardization Process. Available online: https://csrc.nist.gov/pqc-standardization (accessed on 5 June 2025).
16. Iavich, M.; Kuchukhidze, T. Digital Signature Design Using Verkle Tree. In Proceedings of the 28th International Conference on Information Society and University Studies (IVUS 2023), Kaunas, Lithuania, 12 May 2023; pp. 83–91.
17. Catalano, D.; Fiore, D. Vector commitments and their applications. In Proceedings of the Public-Key Cryptography–PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, 26 February–1 March 2013; Springer: Berlin/Heidelberg, Germany, 2013.
18. Iavich, M.; Kuchukhidze, T.; Bocu, R. A Post-Quantum Digital Signature Using Verkle Trees and Lattices. *Symmetry* **2023**, *15*, 2165. [CrossRef]
19. Iavich, M.; Kapalova, N. Optimizing Post-Quantum Digital Signatures with Verkle Trees and Quantum Seed-Based Pseudo-Random Generators. *Computers* **2025**, *14*, 103. [CrossRef]
20. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *Ann. Internal Med.* **2009**, *151*, 264–269. [CrossRef] [PubMed]
21. NIST Computer Security Division: Post-Quantum Cryptography Standardization—Call for Proposals Announcement. 2017. Available online: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization (accessed on 5 June 2025).
22. Hülsing, A. W-OTS+—Shorter signatures for hash-based signature schemes. In *International Conference on Cryptology in Africa*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 173–188
23. Bernstein, D.J.; Dobraunig, C.; Eichlseder, M.; Fluhrer, S.; Gazdag, S.L.; Hülsing, A.; Kampanakis, P.; Kölbl, S.; Lange, T.; Lauridsen, M.; et al. SPHINCS+, A submission to the NIST Standardization Project on Post-Quantum Cryptography. Discrete MathematicsApplied and Provable SecurityCoding Theory and Cryptology. 2017. Available online: https://research.tue.nl/en/publications/sphincs-submission-to-the-nist-post-quantum-cryptography-project (accessed on 5 June 2025).
24. Hulsing, A.; Kudinov, M.; Ronen, E.; Yogev, E. SPHINCS+C: Compressing SPHINCS+ with (Almost) No Cost. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–24 May 2023; pp. 1435–1453.
25. Kuszmaul, J. Verkle Trees. 2019, pp. 1–12. Available online: https://math.mit.edu/research/highschool/primes/materials/2018/Kuszmaul.pdf (accessed on 5 June 2025).
26. Leighton, F.; Micali, S. Large Provably Fast and Secure Digital Signature Schemes Based on Secure Hash Functions. U.S. Patent 5,432,852, 11 July 1995. Available online: https://www.google.com/patents/US5432852 (accessed on 5 June 2025).
27. Perrig, A. The BiBa one-time signature and broadcast authentication protocol. In Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, PA, USA, 5–8 November 2001; pp. 28–37.
28. Reyzin, L.; Reyzin, N. Better than BiBa: Short one-time signatures with fast signing and verifying. In Proceedings of the Australasian Conference on Information Security and Privacy, Perth, WA, Australia, 3–5 July 2002; Volume 2384, pp. 144–153.
29. Buchmann, J.; García, L.C.C.; Dahmen, E.; Doring, M.; Klintsevich, E. CMSS—An improved Merkle signature scheme. In *INDOCRYPT 2006*; Barua, R., Lange, T., Eds.; Lecture Notes in Computer Science (LNCS); Springer: Berlin/Heidelberg, Germany, 2006; Volume 4329, pp. 349–363.
30. Buchmann, J.; Dahmen, E.; Klintsevich, E.; Okeya, K.; Vuillaume, C. Merkle signatures with virtually unlimited signature capacity. In *Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 31–45.
31. Lee, J.; Kim, S.; Cho, Y.; Chung, Y.; Park, Y. HORSIC: An efficient one-time signature scheme for wireless sensor networks. *Inf. Process. Lett.* **2012**, *112*, 783–787. [CrossRef]
32. Hulsing, A.; Rausch, L.; Buchmann, J. Optimal parameters for XMSS^{MT}. In Proceedings of the Security Engineering and Intelligence Informatics: CD-ARES 2013 Workshops: MoCrySEn and SeCIHD, Regensburg, Germany, 2–6 September 2013; Proceedings 8; Springer: Berlin/Heidelberg, Germany, 2013; pp. 194–208.
33. Kolbl, S.; Lauridsen, M.; Mendel, F.; Rechberger, C. *Haraka v2—Efficient Short-Input Hashing for Post-Quantum Applications*; IACR Cryptology ePrint Archive: Report 2016/098; International Association for Cryptologic Research: Bellevue, WA, USA, 2016.

34. Gueron, S.; Mouha, N. Simpira v2: A family of efficient permutations using the aes round function. In Proceedings of the Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 4–8 December 2016; Proceedings, Part I; Springer: New York, NY, USA, 2016; pp. 95–125.
35. Gueron, S.; Mouha, N. Sphincs-simpira: Fast Stateless Hash-Based Signatures with Post-Quantum Security. Cryptology ePrint Archive. Paper 2017/645. Available online: https://eprint.iacr.org/2017/645 (accessed on 5 June 2025).
36. Aumasson, J.P.; Endignoux, G. Gravity SPHINCS, A Submission to the NIST Standardization Project on Post-Quantum Cryptography. 2017. Available online: https://github.com/gravity-postquantum/gravity-sphincs (accessed on 5 June 2025).
37. Aumasson, J.P.; Endignoux, G. Improving stateless hash-based signatures. In *Cryptographers' Track at the RSA Conference*; Springer: Cham, Switzerland, 2018; pp. 219–242.
38. Kiktenko, E.; Bulychev, A.; Karagodin, P.; Pozhar, N.; Anufriev, M.; Fedorov, A. Sphincs+ postquantum digital signature scheme with streebog hash function. *AIP Conf. Proc.* **2020**, *2241*, 020014.
39. Lee, J.; Park, Y. HORSIC+: An Efficient Post-Quantum Few-Time Signature Scheme. *Appl. Sci.* **2021**, *11*, 7350. [CrossRef]
40. Zhang, K.; Cui, H.; Yu, Y. Sphincs-α: A compact stateless hash-based signature scheme. *Cryptol. Eprint Arch.* **2022**, *2022*, 59.
41. Sim, M.; Eum, S.; Song, G.; Kwon, H.; Jang, K.; Kim, H.; Kim, H.; Yang, Y.; Kim, W.; Lee, W.K.; et al. K-XMSS and K-SPHINCS+: Hash Based Signatures with Korean Cryptography Algorithms. Cryptology ePrint Archive. 2022. Paper 2022/152. Available online: https://eprint.iacr.org/2022/152 (accessed on 5 June 2025).
42. Iavich, M.; Kuchukhidze, T.; Okhrimenko, T. Verkle Tree-based Post-Quantum Digital Signature Scheme using Stateless Updatable Vector Commitment. In Proceedings of the CPITS-2023-II: Cybersecurity Providing in Information and Telecommunication Systems, Kyiv, Ukraine, 26 October 2023; pp. 157–166.
43. Ning, Y.; Dong, J.; Lin, J.; Zheng, F.; Fu, Y.; Dong, Z.; Xiao, F. GRASP: Accelerating Hash-Based PQC Performance on GPU Parallel Architecture. Cryptology ePrint Archive, Paper 2024/1030. 2024. Available online: https://eprint.iacr.org/2024/1030 (accessed on 5 June 2025).
44. Algazy, K.; Sakan, K.; Nyssanbayeva, S.; Lizunov, O. Syrga2: Post-Quantum Hash-Based Signature Scheme. *Computation* **2024**, *12*, 125. [CrossRef]
45. Sedghighadikolaei, K.; Yavuz, A.A.; Nouma, S.E. Signer-Optimal Multiple-Time Post-Quantum Hash-Based Signature for Heterogeneous IoT Systems. *arXiv* **2024**, arXiv:2411.01380.
46. McGrew, D.; Kampanakis, P.; Fluhrer, S.; Gazdag, S.L.; Butin, D.; Buchmann, J. State management for hash-based signatures. In Proceedings of the Security Standardisation Research: Third International Conference (SSR 2016), Gaithersburg, MD, USA, 5–6 December 2016; Proceedings 3; Springer: Berlin/Heidelberg, Germany, 2016; pp. 244–260.
47. Cooper, D.; Apon, D.; Dang, Q.; Davidson, M.; Dworkin, M.; Miller, C. *Recommendation for Stateful Hash-Based Signature Schemes*; Draft NIST Special Publication SP 800–208; NIST: Gaithersburg, MD, USA, 2019.
48. Bruinderink, L.G.; Hulsing, A. 'Oops, I did it again'—Security of one-time signatures under two-message attacks. In Proceedings of the International Conference on Selected Areas in Cryptography, Ottawa, ON, Canada, 16–18 August 2017; Springer: Cham, Switzerland, 2017; pp. 299–322.
49. NIST: FIPS 205 Stateless Hash-Based Digital Signature Standard. 2024. Available online: https://csrc.nist.gov/pubs/fips/205/final (accessed on 5 June 2025).
50. Berger, R.M.; Tiepelt, M. On forging SPHINCS+-Haraka signatures on a fault-tolerant quantum computer. In Proceedings of the Progress in Cryptology—LATINCRYPT 2021: 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, 6–8 October 2021; Proceedings 7; Springer: Cham, Switzerland, 2021; pp. 44–63.
51. Castelnovi, L.; Martinelli, A.; Prest, T. Grafting trees: A fault attack against the SPHINCS framework. In Proceedings of the International Conference on Post-Quantum Cryptography, Fort Lauderdale, FL, USA, 9–11 April 2018; Springer: Cham, Switzerland, 2018; pp. 165–184.
52. 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; pp. 229–237.
53. Genet, A. On Protecting SPHINCS+ Against Fault Attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2023**, *2023*, 80–114. [CrossRef]
54. Perlner, R.; Kelsey, J.; Cooper, D. Breaking category five SPHINCS+ with SHA-256. In Proceedings of the PQCrypto 2022: The Thirteenth International Conference on Post-Quantum Cryptography, Online, 28–30 September 2022; Springer: Cham, Switzerland, 2022; pp. 501–522.