A Systematic Literature Review on Optimising CRYSTALS-Dilithium (ML-DSA) Performance for IoT Devices via Lightweight Hashing

Ceasar Njuguna Ngunu* Edward Ombui
United States International University–Africa, Nairobi, Kenya
*Corresponding author: cnngunu@usiu.ac.ke

Abstract

Background: The migration to post-quantum cryptography confronts resource-constrained Internet of Things (IoT) devices with a material performance cost. CRYSTALS-Dilithium, standardised as the Module-Lattice-Based Digital Signature Algorithm (ML-DSA) in FIPS 204, fixes the Keccak-based SHAKE functions as its only symmetric primitives, and profiling on embedded platforms identifies hashing as the largest single contributor to the scheme's software cost. This review synthesises the performance evidence for ML-DSA on constrained platforms, classifies the optimisation strategies pursued in the literature, and tests whether any published work substitutes a standardised lightweight extendable-output function for SHAKE within the scheme.

Methods: Following Kitchenham's guidelines and the PRISMA 2020 statement, we searched IEEE Xplore, the ACM Digital Library, Scopus, and SpringerLink for peer-reviewed studies published from January 2020 onwards, complemented by backward and forward snowballing and by targeted update searches through July 2026. A protocol was prepared in advance of the search. From 115 database records and 22 records identified through other methods, 40 primary studies met the inclusion criteria.

Results: On the ARM Cortex-M4, optimised software implementations of Dilithium3 require 10,667 kilocycles on average for signing and 2,321 kilocycles for verification; on the Cortex-M7, Dilithium-2 verification averages 1,429 kilocycles (6.6 ms at 216 MHz), with signing spanning 1,835 to 16,440 kilocycles due to rejection sampling. Optimisation efforts fall into four categories: hardware acceleration, platform-specific software optimisation, protocol-level adaptation, and optimisation of the incumbent Keccak primitive itself. Architecture-specific Keccak optimisation reduces hashing's share of Dilithium's runtime on the Cortex-M4 by only 2.46 to 5.03 percentage points, indicating that the bottleneck largely survives direct attack. Replacing Keccak with Ascon inside the sibling scheme Kyber yields a 24 to 25 % cycle reduction and a 2 to 8 % memory reduction on the Cortex-M4. No peer-reviewed study applies this substitution to ML-DSA.

Conclusions: With FIPS 204 and NIST SP 800-232 both final, the cost of ML-DSA's primitive choice on constrained platforms is a well-posed and unanswered question on both sides. We specify a per-call-site Dilithium–Ascon evaluation, including its security constraints and non-conformance status, as the priority direction for software-only optimisation of post-quantum signatures on IoT devices.

Keywords: post-quantum cryptography; ML-DSA; CRYSTALS-Dilithium; Ascon; lightweight

cryptography; Internet of Things; systematic literature review

1 Introduction

1.1 Background

The proliferation of Internet of Things (IoT) devices across consumer electronics, industrial monitoring, and critical infrastructure has created a large and long-lived attack surface whose security rests on classical public-key cryptography (Castiglione et al., 2025; Señor et al., 2024). A cryptographically relevant quantum computer running Shor's algorithm would solve the integer factorisation and discrete logarithm problems in polynomial time, breaking RSA and elliptic-curve schemes, and the harvest-now, decrypt-later adversary model makes the migration urgent even before such a machine exists (Malina et al., 2023). In response, the U.S. National Institute of Standards and Technology (NIST) standardised a first suite of post-quantum algorithms, publishing the Module-Lattice-Based Digital Signature Standard (ML-DSA), derived from CRYSTALS-Dilithium, as FIPS 204 in August 2024 (National Institute of Standards and Technology, 2024). Throughout this review the names Dilithium and ML-DSA are used interchangeably, preferring the pre-standard name when discussing pre-2024 literature.

Constrained devices carry this migration at a disadvantage. Post-quantum schemes have larger keys, larger signatures, and heavier computation than the classical schemes they replace, and evaluations across microcontroller-class platforms consistently report material latency, memory, and energy costs (Sajimon et al., 2022; Señor et al., 2022; Abbasi et al., 2025), a picture already visible in the pre-standardisation embedded literature (Güneysu et al., 2018). The cost concentrates in specific places. FIPS 204 fixes the Keccak-based SHAKE-128 and SHAKE-256 extendable-output functions as the scheme's only symmetric primitives (National Institute of Standards and Technology, 2024), and profiling on platforms without SHA-3 instruction support identifies hashing as the largest single contributor to the scheme's software runtime, ahead of the number-theoretic transform (Kannwischer et al., 2019; Huang et al., 2024). For device-local, latency-sensitive functions such as secure boot and firmware verification, where every boot pays the verification cost, this primitive-level bottleneck is a direct obstacle to adoption (Kumar et al., 2020; Gupta et al., 2023).

A separate NIST process addressed symmetric cryptography for the same device class. The Ascon family (Dobraunig et al., 2021), winner of the lightweight cryptography standardisation process, was published in final form as NIST SP 800-232 in August 2025, defining among other primitives the extendable-output functions Ascon-XOF128 and Ascon-CXOF128 (National Institute of Standards and Technology, 2025). The two standards therefore now meet in the middle: a signature standard whose embedded cost is dominated by its hash, and a hashing standard designed for exactly the platforms where that cost bites.

1.2 Motivation and problem statement

A recent study replaced the Keccak functions inside CRYSTALS-Kyber, ML-DSA's sibling key-encapsulation mechanism, with Ascon and measured the effect on an ARM Cortex-M4, reporting a 24 to 25 % reduction in clock cycles and a 2 to 8 % reduction in memory usage (Shehzadi et al.,

2025). This demonstrates that primitive substitution is an effective, software-only optimisation for module-lattice schemes on constrained hardware. Whether the same holds for the signature scheme, whose hashing workload differs in structure and security parameters, is unknown: to the best of our knowledge, no peer-reviewed study instantiates Dilithium or ML-DSA with Ascon or any standardised lightweight extendable-output function. This review establishes that gap systematically and specifies the study that would close it.

1.3 Research questions

RQ1: What are the documented performance characteristics (latency, memory usage, energy consumption) of ML-DSA (CRYSTALS-Dilithium) on resource-constrained platforms such as the ARM Cortex-M family, RISC-V cores, and low-cost microcontrollers?
RQ2: What optimisation strategies does the literature pursue for post-quantum algorithms in lightweight environments?
RQ3: Does the literature contain evidence of optimising ML-DSA by replacing its internal Keccak/SHAKE functions with a standardised lightweight alternative such as Ascon?

1.4 Contributions

This review contributes, first, a verified synthesis of ML-DSA performance benchmarks across constrained platforms, including the memory and energy dimensions; second, a four-category taxonomy of optimisation strategies that separates optimisation of the incumbent primitive from its replacement; third, the systematic identification of the Dilithium–Ascon gap, scoped precisely against its nearest boundary cases; and fourth, a design specification for the study that would close the gap, including per-call-site substitution constraints, security-level analysis, and standards-conformance status.

1.5 Scope and limitations

The review addresses performance (speed, memory, energy) of post-quantum signature schemes on constrained devices, with ML-DSA as its focus and internal hash primitives as the optimisation vector under examination. Cryptanalysis of the underlying lattice problems is out of scope. Implementation security is out of scope as an evaluation dimension, although the side-channel literature is acknowledged where it bears on the recommendation (Qiao et al., 2023; Wang et al., 2025).

2 Methodology

The review follows Kitchenham's guidelines for systematic literature reviews in software engineering (Kitchenham and Charters, 2007), with reporting aligned to the PRISMA 2020 statement (Page et al., 2021). A review protocol specifying the research questions, search strategy, and selection criteria was prepared in advance of the search, dated 28 October 2025, and is available from the authors.

2.1 Search strategy

The primary databases were IEEE Xplore, the ACM Digital Library, Scopus, and SpringerLink, with Google Scholar used for citation searching. Search strings combined terms from four categories (algorithm, concept, context, and metric); the base string was ("CRYSTALS-Dilithium" OR "Dilithium" OR "ML-DSA") AND ("performance" OR "benchmarking" OR "optimization") AND ("IoT" OR "embedded system" OR "Cortex-M" OR "resource-constrained"), with a companion string substituting the algorithm terms ("Ascon" OR "lightweight cryptography" OR "SHAKE" OR "Keccak"). The per-database adaptations are given in Appendix B. Database searches, concluded in November 2025, were complemented by backward snowballing over the reference lists of included studies and forward snowballing over their citing papers, and by targeted update searches, re-executing the RQ3 gap terms (Dilithium and ML-DSA combined with Ascon and lightweight hash terms) across the same databases and citation indexes in July 2026 to confirm the currency of the RQ3 finding. During the July 2026 pass, every citation in the review was additionally verified against its source of record, and the study set was corrected accordingly.

2.2 Study selection

Studies were included when they satisfied all of the following: publication from January 2020 onwards in a peer-reviewed venue; quantitative performance data (cycle counts, execution time, memory usage, or energy); a resource-constrained or embedded target platform, or a primitive-level result directly applicable to such platforms; and availability in English. Studies were excluded when they were purely theoretical cryptanalysis without implementation data, unavailable in full text, or grey literature not corroborated by peer-reviewed sources. Titles and abstracts were screened first, followed by full-text assessment. Screening, extraction, and quality assessment were performed by a single author; as mitigation, all selection decisions and every extracted data point were re-examined in a second full verification pass against the source documents, and the review's principal claims were checked against the primary sources rather than database metadata. The residual limitation is discussed in Section 4.6. Figure 1 presents the PRISMA 2020 flow.

2.3 Quality assessment

Each included study was assessed against three questions: QA1, whether the target platform and experimental setup are described sufficiently for reproducibility; QA2, whether metrics were measured on hardware or cycle-accurate simulation rather than estimated; and QA3, whether the study provides a direct benchmark or optimisation relevant to ML-DSA or its primitives. Scores (yes, partial, no) per study are reported alongside the study list in Appendix A. Studies scoring partial on QA3 contribute contextual rather than headline evidence in the synthesis; one report was excluded on venue-quality grounds during full-text assessment, as recorded in the flow.

2.4 Data extraction and synthesis

A structured form captured bibliographic details, target platform and clock frequency, parameter set, implementation type (reference, optimised software, hardware-accelerated), performance metrics with their reporting basis (average, median, or distribution), and the optimisation strat-

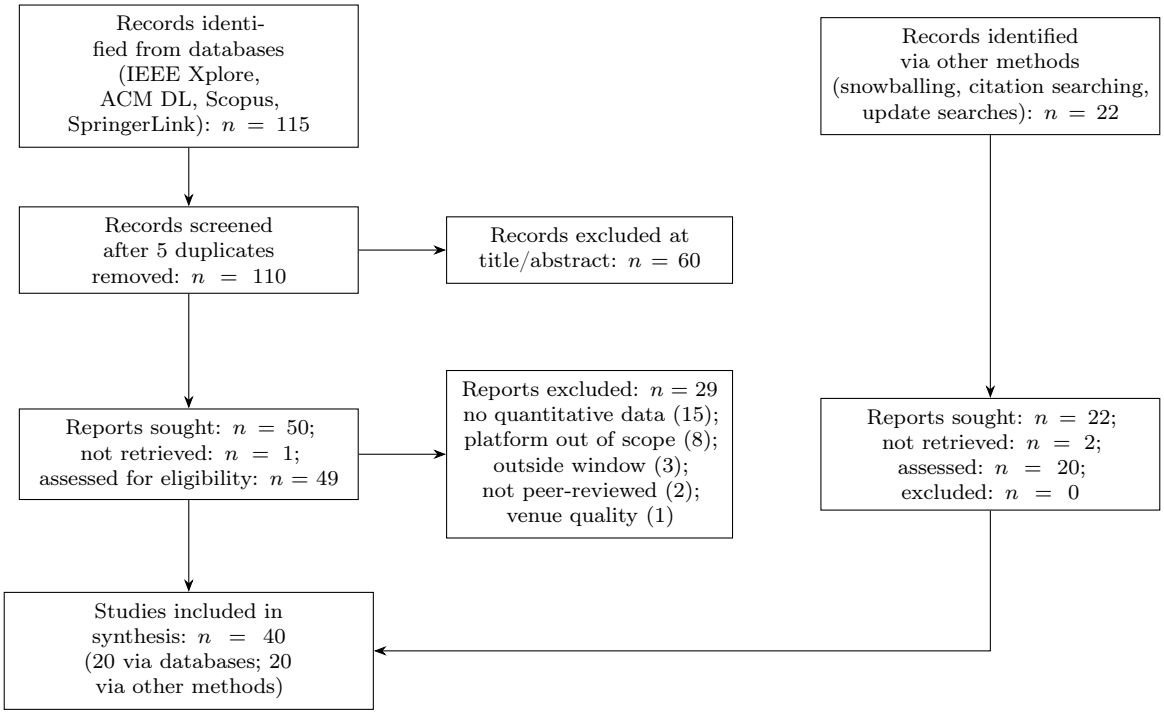

Figure 1: PRISMA 2020 flow diagram of the study selection process, covering the November 2025 search and the July 2026 verification and update pass.

egy employed. Synthesis was narrative and thematic: performance data were tabulated for RQ1, strategies were categorised for RQ2, and the full evidence base was analysed against RQ3.

3 Results

3.1 RQ1: Performance of ML-DSA on constrained platforms

The evidence establishes a pronounced performance divide between low-power microcontrollers and application-class processors, and within the former, a strong dependence on implementation type. Table 1 reports the headline verified benchmarks.

On the ARM Cortex-M4, the reference platform of the NIST embedded evaluation, optimised software implementations of Dilithium3 require 10,667 kilocycles on average for signing and 2,321 kilocycles for verification (Greconici et al., 2021), with subsequent arithmetic optimisations delivering further gains of 5 to 6 % in the transforms and roughly a third in the small-polynomial operations of signing (Abdulrahman et al., 2022), and parallel small-polynomial multiplication adding 18 to 19 % to signing and verification for Dilithium-2 (Zheng et al., 2022). On an FRDM-K64 board (Cortex-M4F at 120 MHz), optimised Level-2 verification takes 2,189 kilocycles, or 18.2 ms (Ortega L. and Dominguez Perez, 2021). On the Cortex-M7, native benchmarking shows that Dilithium profits little from the larger core beyond its clock, in contrast to Falcon, whose floating-point workload gains factors of 6.2 to 8.3 in cycles from the M7's double-precision unit (Howe and Westerbaan, 2023). The same study documents the width of Dilithium's signing distribution, from 1,835 to 16,440 kilocycles for Dilithium-2, a direct consequence of rejection sampling that any evaluation must report distributionally rather than as a single mean. On application-class 64-bit cores the picture inverts: Neon-vectorised implementations set speed records on the Cortex-

Table 1: Verified ML-DSA (Dilithium) performance benchmarks on constrained platforms. kcc denotes kilocycles; values are averages unless stated.

Study	Platform	Set	Operation	Result
Howe and Westerbaan (2023)	Cortex-M7, 216 MHz	Dil.-2	Key gen	1,437 kcc (6.7 ms)
Howe and Westerbaan (2023)	Cortex-M7, 216 MHz	Dil.-2	Sign	3,658 kcc; min 1,835, max 16,440
Howe and Westerbaan (2023)	Cortex-M7, 216 MHz	Dil.-2	Verify	1,429 kcc (6.6 ms)
Howe and Westerbaan (2023)	Cortex-M7, 216 MHz	Falcon-512	Verify	559 kcc (2.6 ms)
Greconici et al. (2021)	Cortex-M4	Dil.3	Sign	10,667 kcc
Greconici et al. (2021)	Cortex-M4	Dil.3	Verify	2,321 kcc
Ortega L. and Dominguez Perez (2021)	Cortex-M4F, 120 MHz	Level 2	Verify	2,189 kcc (18.2 ms)
Kim et al. (2022)	ARMv8 MCU	Dil.3	Sign	+113.25 % vs reference
Gupta et al. (2023)	RISC-V (Ariane), 50 MHz	Dil.	Secure boot	3.48 ms (HW) vs 14.1 s (SW)
Fitzgibbon and Ottaviani (2024)	Raspberry Pi 4 (A72)	Dil.3	Sign / Verify	0.90 ms / 0.31 ms
Halak et al. (2024)	Pico W (Cortex-M0+)	Dil.-2	Verify (in TLS)	42.4 ms
Shehzadi et al. (2025)	Cortex-M4	Kyber512+Ascon	KEM ops	−24–25 % cycles; −2–8 % memory
Huang et al. (2024)	Cortex-M4	Dil.	Full scheme	−13.94–15.52 % via Keccak opt.

A72 and Apple M1 (Becker et al., 2022), an ARMv8 microcontroller implementation improved key generation, signing, and verification by 43.83 %, 113.25 %, and 41.92 % respectively over the reference (Kim et al., 2022), and on a Raspberry Pi 4 (Cortex-A72) Dilithium3 signs in 0.90 ms and verifies in 0.31 ms, the fastest of the standardised signatures on that class of device (Fitzgibbon and Ottaviani, 2024). At the opposite end of the spectrum, client-side Dilithium2 verification within a TLS handshake takes 42.4 ms on a Cortex-M0+ class Raspberry Pi Pico W (Halak et al., 2024), extending the evidence base to the most constrained deployable parts. Feasibility extends to low-cost parts: Dilithium-5 signing has been demonstrated on ESP32-class microcontrollers within a blockchain key-management design (Castiglione et al., 2025), and system-level studies confirm deployability across wireless sensor networks and heterogeneous environments, with constrained devices showing up to twelvefold variation in computational demand between algorithms at equal security level, and naïve protocol integration inflating TLS 1.3 handshakes by up to seven times (Señor et al., 2024; Abbasi et al., 2025).

Memory constitutes a second constraint dimension. Reference and optimised Cortex-M4 implementations occupy roughly 50 to 100 KiB, and dedicated memory-optimisation work has reduced this to under 7 KiB for key and signature generation and under 3 KiB for verification at the recommended parameter set (Bos et al., 2022); on the Cortex-M7, unoptimised stack usage for Dilithium-2 spans 36 to 52 KiB across operations (Howe and Westerbaan, 2023). Energy constitutes a third. At the primitive level, ASIC evaluation of the lightweight cryptography finalists places Ascon among the most energy-efficient, achieving 10 to 25 times better energy efficiency than the weakest finalists at equal workloads (Elsadek et al., 2022b), with pipelining and parallelisation adding up to 49 % further (Elsadek et al., 2022a). At system level, mutual-authentication TLS measurements on Raspberry-Pi-class and Cortex-M0+ clients identify the Kyber-512 with Dilithium-2 pairing as the most resource-efficient configuration in latency, memory, and energy, outperforming even the classical ECDHE-ECDSA baseline because encapsulation replaces the costlier Diffie-Hellman exchange (Halak et al., 2024), and battery-lifetime modelling of a Cortex-M4 edge device shows fully post-quantum TLS sustaining multi-year runtimes at realistic duty cycles (Schöffel et al., 2022).

Where hardware acceleration is available, the cost changes regime entirely. A Dilithium secure-boot architecture around the open-source Ariane RISC-V core adds 3.48 ms to the boot sequence against 14.1 s for the software path on the same system (Gupta et al., 2023). The synthesis for RQ1 is therefore sharper than a general observation of overhead: on unaccelerated 32-bit microcontrollers, ML-DSA costs milliseconds per operation and tens of kilobytes of memory, its signing time is highly variable, and profiling attributes the largest share of the software cost to a single primitive, Keccak, whose permutation was not designed for such platforms (Kannwischer et al., 2019; Huang et al., 2024).

3.2 RQ2: Optimisation strategies in the literature

The included studies pursue four distinguishable strategies. The first is hardware acceleration, offloading the expensive kernels, principally the number-theoretic transform and the Keccak permutation, to dedicated logic: full accelerators and co-designs for Dilithium and Falcon (Beckwith et al., 2023; Gupta et al., 2023), tightly coupled RISC-V instruction-set extensions that cut NTT cycle counts by more than 80 % (Karl et al., 2024), lightweight coprocessors for the LWC portfolio (Nguyen et al., 2024), and compact accelerators for alternative lattice instances (He et al., 2023). The second is platform-specific software optimisation, exploiting the target microarchitecture: Neon vectorisation on ARMv8-A (Becker et al., 2022; Kim et al., 2022), improved modular arithmetic and register scheduling on the Cortex-M family (Greconici et al., 2021; Abdulrahman et al., 2022), algorithm-level parallelism in the small-polynomial multiplications (Zheng et al., 2022), and memory-footprint reduction for small parts (Bos et al., 2022). The third is protocol and application-level adaptation, restructuring how and where the expensive operations are invoked: KEM-only authentication for MQTT on 8-bit AVR nodes (Kim and Seo, 2025), immediately deployable post-quantum key delivery for the Web of Things (Kim et al., 2025), backward-compatible BLE key exchange (Liu et al., 2024), offloading the network stack to data-processing units (Lawo et al., 2024), split-key migration paths for device boot integrity (Ahola et al., 2025), blockchain-mediated key-state management (Seng et al., 2024), domain-specific deployments in vehicular and infrastructure settings (Figlarz and Hessel, 2022; Malina et al., 2023), and heterogeneous public-key infrastructures that assign different signature schemes to different certificate roles: combining Falcon-based server and certificate-authority certificates with conventional client certificates reduces the energy overhead of migrating a low-power TLS deployment from 230 % to 30 % and the latency overhead from 115 % to 50 % relative to a purely classical baseline (Schöffel et al., 2022). The fourth, absent from earlier taxonomies, is optimisation of the incumbent primitive itself: architecture-specific Keccak improvements on ARMv7-M yield whole-scheme Dilithium speedups of 13.94 to 15.52 % on the Cortex-M4 without touching the specification (Huang et al., 2024); the parallel LWC hardware literature plays the analogous role on the Ascon side, characterising the candidate replacement's efficiency across FPGA and ASIC targets (Bîrleanu and Bizon, 2022; Elsadek and Tawfik, 2024; El-Hadedy et al., 2024; Malal, 2024; Wu, 2024; Elsadek et al., 2025).

The synthesis for RQ2 carries the weight of this review's argument. Every identified strategy either accepts Keccak and accelerates around it, accepts Keccak and accelerates it directly, or restructures the protocol so the signature is invoked less. The fourth category's own results

bound what direct acceleration can achieve: even architecture-specific optimisation of Keccak reduces hashing's share of Dilithium's runtime on the Cortex-M4 by only 2.46 to 5.03 percentage points (Huang et al., 2024). The bottleneck survives direct attack. Beyond it, the remaining moves are dedicated hardware, which carries non-recurring engineering cost and forfeits deployment on existing silicon, or a different primitive, and only the latter is a software-only path.

3.3 RQ3: The identified research gap

Three converging lines of evidence establish the gap. The first is precedent within the same algorithm family: substituting Ascon for Keccak inside Kyber reduced clock cycles by 24 to 25 % and memory by 2 to 8 % on the Cortex-M4, with the largest gains in the most optimised configuration (Shehzadi et al., 2025). The second is precedent within Dilithium itself: the round-three submission specified AES-based variants that substituted the symmetric primitive for performance on AES-accelerated platforms (Bai et al., 2021), variants not carried into the published standard, which retains SHAKE exclusively (National Institute of Standards and Technology, 2024). The scheme's designers thus treated the symmetric layer as replaceable in principle. The third is standardisation convergence: with SP 800-232 final, Ascon-XOF128 and Ascon-CXOF128 are NIST-standardised extendable-output functions expressly intended for constrained devices (National Institute of Standards and Technology, 2025), so an ML-DSA variant instantiated with Ascon would compose two published standards rather than pair a standard with an experimental primitive.

Against this, the search identified no peer-reviewed study that implements or benchmarks Dilithium or ML-DSA with Ascon, or with any standardised lightweight extendable-output function, on any platform. The nearest boundary objects do not close the gap. Ascon-Sign, submitted to NIST's additional-signatures process, instantiates a hash-based construction with Ascon rather than modifying a lattice scheme (Ascon-Sign Team, 2023). Open-source experiments substituting non-standardised primitives into ML-DSA exist as uncorroborated grey literature and fall under this review's exclusion criteria. And the Keccak-optimisation line (Huang et al., 2024) improves the incumbent rather than replacing it, while demonstrating that improvement alone barely moves the primitive's share of the cost. The gap claim is therefore stated precisely: as of the search close, no peer-reviewed performance evaluation exists of ML-DSA instantiated with a NIST-standardised lightweight extendable-output function, despite a quantified precedent in the sibling KEM, a within-scheme precedent for primitive substitution, and profiling evidence that the primitive in question dominates the scheme's software cost on the platforms of interest.

4 Discussion

4.1 Summary of findings

On unaccelerated microcontrollers, ML-DSA is expensive in exactly the dimensions constrained deployments care about: milliseconds of latency per operation, tens of kilobytes of memory before memory-specific optimisation, high signing-time variance, and an energy cost that primitive-level choices influence directly. The optimisation literature is mature along four lines, hardware, platform-specific software, protocol adaptation, and incumbent-primitive optimisation, but all

four leave the standard's primitive choice unexamined as a variable.

4.2 Confirmation of the research gap

The most consequential finding is that the RQ3 gap is real, current, and newly timely. The precedent is quantitative (Shehzadi et al., 2025), the bottleneck is quantified (Kannwischer et al., 2019; Huang et al., 2024), and the timeliness follows from standardisation: FIPS 204 fixed SHAKE into ML-DSA in August 2024 (National Institute of Standards and Technology, 2024) and SP 800-232 standardised Ascon's extendable-output functions in August 2025 (National Institute of Standards and Technology, 2025). The question of what a standardised lightweight XOF does to the standardised signature scheme on constrained hardware is now well posed on both sides, and unanswered.

4.3 Design constraints for a Dilithium–Ascon evaluation

The recommended study is not a mechanical substitution, and stating its constraints strengthens the proposal. ML-DSA invokes extendable-output functions at call sites with different security parameters: SHAKE-128 for expansion of the public matrix, and SHAKE-256 for seed expansion, mask generation, and challenge derivation (National Institute of Standards and Technology, 2024). Ascon-XOF128 provides 128-bit security (National Institute of Standards and Technology, 2025), which aligns naturally with the SHAKE-128 call sites but requires an explicit argument, or a restriction to the ML-DSA-44 parameter set, before it can stand in for SHAKE-256 call sites at higher security categories. A per-call-site substitution design is therefore the correct experimental object, with the security rationale documented for each site. Two further constraints bound interpretation. A variant instantiated with Ascon is by construction non-conformant with FIPS 204 and must be positioned as a research artefact quantifying the cost of the standard's primitive choice, not as a deployable replacement. And because substitution alters the inputs to the rejection-sampling loops, the evaluation must report signing-time distributions, a requirement the verified benchmarks make concrete: Dilithium-2 signing on the Cortex-M7 spans 1,835 to 16,440 kilocycles around a 3,658-kilocycle average (Howe and Westerbaan, 2023).

4.4 The verification-side alternative

An honest account of the design space notes Falcon. On the same Cortex-M7 board, Falcon-512 verification takes 559 kilocycles against Dilithium-2's 1,429, and its signatures are smaller (Howe and Westerbaan, 2023); Falcon also yields the smallest post-quantum TLS handshake measured between constrained devices, 7,480 bytes with Kyber768 against 11,694 for Dilithium3 (Fitzgibbon and Ottaviani, 2024), and system-level work reaches the same conclusion for bandwidth-tight, verify-heavy deployments (Lawo et al., 2024). The asymmetry cuts both ways: Falcon's signing function is the dominant cost on client devices, which is why measured optima assign Falcon to server and authority certificates while clients sign with Dilithium or a classical scheme (Schöffel et al., 2022). For verify-only functions such as secure boot, Falcon is therefore a credible alternative. Three considerations keep the present recommendation on ML-DSA: FIPS 204 is the primary signature standard with the broadest implementation and validation ecosystem; Falcon's floating-point signing exhibits constant-time irregularities on the same class of devices (Howe and

Westerbaan, 2023); and the hashing cost this review targets is common to the lattice family, so a validated substitution methodology transfers.

4.5 Implications

Hardware acceleration delivers the largest constant factors (Gupta et al., 2023; Beckwith et al., 2023; Karl et al., 2024) but at non-recurring engineering cost and with no path to the installed base. A validated software-only optimisation of the standard's dominant primitive would be deployable on existing silicon by firmware update, which is precisely the deployment model of the secure-boot and over-the-air-update functions that motivate this review (Kumar et al., 2020; Ahola et al., 2025). The distinction between deployment contexts sharpens the point. In network protocols, the dominant post-quantum cost is communication rather than computation: handshake latency is driven by the larger keys and signatures (Schöffel et al., 2022; Halak et al., 2024), with handshake sizes inflating up to sevenfold (Abbasi et al., 2025), and the same measurement campaigns conclude that dedicated hardware accelerators are not mandatory for feasible deployment on off-the-shelf microcontrollers (Schöffel et al., 2022). In device-local verification, by contrast, the cost is computation (Gupta et al., 2023), and within that computation, hashing (Kannwischer et al., 2019; Huang et al., 2024), which is precisely where a software-only primitive substitution acts. The energy evidence compounds it: Ascon's order-of-magnitude energy advantages at the primitive level (Elsadek et al., 2022b, 2025) suggest the substitution's benefit grows in battery-powered deployments, a hypothesis the recommended study should measure directly. Side-channel behaviour of the substituted primitive requires dedicated analysis before any deployment claim, given the demonstrated leakage attack surface of Dilithium implementations (Qiao et al., 2023; Wang et al., 2025).

4.6 Limitations of this review

The search covered four databases and English-language, peer-reviewed publications from January 2020, updated through July 2026 with targeted searches on the RQ3 terms; proprietary and grey-literature implementations may exist outside this scope, and one open-source primitive substitution in ML-DSA is known to us but excluded by design. Screening, quality assessment, and extraction were performed by a single author; although all decisions and data points were re-verified against source documents in a second full pass, the absence of an independent second reviewer remains the review's principal methodological limitation, and inter-rater reliability statistics cannot be reported. Finally, implementation-security consequences of a primitive substitution were out of scope and are flagged as required analysis in the recommended study.

5 Conclusion

This review synthesised the verified performance evidence for ML-DSA (CRYSTALS-Dilithium) on constrained platforms, classified the optimisation literature into four strategies, and established a precisely scoped gap: no peer-reviewed evaluation instantiates ML-DSA with a NIST-standardised lightweight extendable-output function, despite a quantified precedent in Kyber, a within-scheme precedent for primitive substitution, and profiling evidence that the targeted primitive dominates

the scheme's embedded software cost. With FIPS 204 and SP 800-232 both final, we recommend, as the priority direction for software-only post-quantum signature optimisation on IoT devices, the implementation and distributional benchmarking of a per-call-site Dilithium–Ascon variant on Cortex-M-class hardware, with its security constraints and non-conformance status stated as specified in Section 4.3.

Declarations

Protocol: A review protocol was prepared in advance of the search, dated 28 October 2025, and is available from the authors. The review was not registered, as PROSPERO does not accept computing reviews.
Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.
Author contributions: Conceptualisation, methodology, investigation, data curation, writing (original draft), and writing (review and editing): C.N.N. Supervision, writing (review and editing), and approval of the final manuscript: E.O.
Competing interests: The authors declare no competing financial or non-financial interests.
Data availability: The included-study list with quality-assessment scores and the search strings are provided in the appendices; the full search log and extraction table are available from the authors.
Declaration of generative AI in the writing process: During the preparation of this work the authors used Claude (Anthropic) to assist with reference verification, drafting, and editing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Abbasi, M., Cardoso, F., Váz, P., Silva, J., and Martins, P. (2025). A practical performance benchmark of post-quantum cryptography across heterogeneous computing environments. *Cryptography*, 9(2):32.

Abdulrahman, A., Hwang, V., Kannwischer, M. J., and Sprenkels, A. (2022). Faster Kyber and Dilithium on the Cortex-M4. In *Applied Cryptography and Network Security (ACNS 2022)*, volume 13269 of *Lecture Notes in Computer Science*, pages 853–871. Springer.

Ahola, J., Sovio, S., and Ekberg, J.-E. (2025). Device boot-up security restoration with post-quantum split-key KEM. In *2025 26th International Symposium on Quality Electronic Design (ISQED)*.

Ascon-Sign Team (2023). Ascon-sign. Submission to the NIST Post-Quantum Cryptography: Additional Digital Signature Schemes, Round 1.

Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., and Stehlé, D. (2021). CRYSTALS-Dilithium: Algorithm specifications and supporting documentation (version 3.1). Submission to the NIST Post-Quantum Cryptography Standardization Process, Round 3.

Becker, H., Hwang, V., Kannwischer, M. J., Yang, B.-Y., and Yang, S.-Y. (2022). Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):221–244.

Beckwith, L., Nguyen, D. T., and Gaj, K. (2023). Hardware accelerators for digital signature algorithms Dilithium and FALCON. *IEEE Design & Test*.

Bîrleanu, F. G. and Bizon, N. (2022). Quick analysis of the NIST lightweight cryptography standardization process finalists. In *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*.

Bos, J. W., Renes, J., and Sprenkels, A. (2022). Dilithium for memory constrained devices. In *Progress in Cryptology – AFRICACRYPT 2022*. Springer.

Castiglione, A., Esposito, J. G., Loia, V., Nappi, M., Pero, C., and Polsinelli, M. (2025). Integrating post-quantum cryptography and blockchain to secure low-cost IoT devices. *IEEE Transactions on Industrial Informatics*, 21(2):1674–1683.

Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2021). Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34.

El-Hadedy, M., Hua, R., Yoshii, K., and Hwu, W.-M. (2024). Optimizing ASCON permutation in multi-clock domains with Chisel: Resource efficiency and critical path reduction. In *2024 IEEE 17th Dallas Circuits and Systems Conference (DCAS)*.

Elsadek, I., Aftabjahani, S., Gardner, D., MacLean, E., Wallrabenstein, J. R., and Tawfik, E. Y. (2022a). Energy efficiency enhancement of parallelized implementation of NIST lightweight cryptography standardization finalists. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*.

Elsadek, I., Aftabjahani, S., Gardner, D., MacLean, E., Wallrabenstein, J. R., and Tawfik, E. Y. (2022b). Hardware and energy efficiency evaluation of NIST lightweight cryptography standardization finalists. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*.

Elsadek, I., Elgendy, E., Abouzeid, S., Ghonem, A. Z., Wallrabenstein, J. R., MacLean, E., Gardner, D., Aftabjahani, S., Cammarota, R., and Tawfik, E. Y. (2025). State-of-the-art ASCON ASIC achieving 4.3 Gbps@0.8V and 3.5 Tb/J@0.6V outperforming AES by 25 times. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 72(8).

Elsadek, I. and Tawfik, E. Y. (2024). Efficient programable architecture for LWC NIST FIPS standard ASCON. In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*.

Figlarz, G. R. and Hessel, F. P. (2022). Applied post-quantum secure method for IoT devices: A case study for autonomous vehicles communication. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*.

Fitzgibbon, G. and Ottaviani, C. (2024). Constrained device performance benchmarking with the implementation of post-quantum cryptography. *Cryptography*, 8(2):21.

Greconici, D. O. C., Kannwischer, M. J., and Sprenkels, A. (2021). Compact Dilithium implementations on Cortex-M3 and Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(1):1–24.

Güneysu, T., Krausz, M., Oder, T., and Speith, J. (2018). Evaluation of lattice-based signature schemes in embedded systems. In *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*.

Gupta, N., Jati, A., and Chattopadhyay, A. (2023). CRYSTALS-Dilithium on RISC-V processor: Lightweight secure boot using post-quantum digital signature. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–7.

Halak, B., Gibson, T., Henley, M., Botea, C.-B., Heath, B., and Khan, S. (2024). Evaluation of performance, energy, and computation costs of quantum-attack resilient encryption algorithms for embedded devices. *IEEE Access*, 12:8791–8805.

He, P., Bao, T., Xie, J., and Amin, M. (2023). FPGA implementation of compact hardware accelerators for Ring-Binary-LWE-based post-quantum cryptography. *ACM Transactions on Reconfigurable Technology and Systems*, 16(3):Article 45.

Howe, J. and Westerbaan, B. (2023). Benchmarking and analysing the NIST PQC lattice-based signature schemes standards on the ARM Cortex M7. In *Progress in Cryptology – AFRICACRYPT 2023*, volume 14064 of *Lecture Notes in Computer Science*, pages 442–462. Springer.

Huang, J., Adomnicăi, A., Zhang, J., Dai, W., Liu, Y., Cheung, R. C. C., Koç, Ç. K., and Chen, D. (2024). Revisiting Keccak and Dilithium implementations on ARMv7-M. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(2).

Kannwischer, M. J., Rijneveld, J., Schwabe, P., and Stoffelen, K. (2019). pqm4: Testing and benchmarking NIST PQC on ARM Cortex-M4. Cryptology ePrint Archive, Paper 2019/844.

Karl, P., Schupp, J., Fritzmann, T., and Sigl, G. (2024). Post-quantum signatures on RISC-V with hardware acceleration. *ACM Transactions on Embedded Computing Systems*.

Kim, J., Kang, J.-H., Lee, H., and Seo, S.-H. (2025). ExpressPQDelivery: Toward efficient and immediately deployable post-quantum key delivery for Web-of-Things. In *Proceedings of the ACM Web Conference 2025 (WWW '25)*.

Kim, Y. and Seo, S. C. (2025). An optimized instantiation of post-quantum MQTT protocol on 8-bit AVR sensor nodes. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25)*.

Kim, Y., Song, J., Youn, T.-Y., and Seo, S. C. (2022). Crystals-Dilithium on ARMv8. *Security and Communication Networks*, 2022:Article 5226390.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Kumar, V. B. Y., Gupta, N., Chattopadhyay, A., Kasper, M., Krauß, C., and Niederhagen, R. (2020). Post-quantum secure boot. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE.

Lawo, D. C., Frantz, R., Cano Aguilera, A., Arnal i Clemente, X., Podleś, M. P., Imaña, J. L., Tafur Monroy, I., and Vegas Olmos, J. J. (2024). Falcon/Kyber and Dilithium/Kyber network stack on Nvidia's data processing unit platform. *IEEE Access*, 12.

Liu, T., Ramachandran, G., and Jurdak, R. (2024). Towards quantum resilient IoT: A backward-compatible approach to secure BLE key exchange against quantum threats. In *2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI)*.

Malal, A. (2024). High-performance FPGA implementations of lightweight ASCON-128 and ASCON-128a with enhanced throughput-to-area efficiency. In *2024 17th International Conference on Information Security and Cryptology (ISCTürkiye)*.

Malina, L., Dobias, P., Hajny, J., and Choo, K.-K. R. (2023). On deploying quantum-resistant cybersecurity in intelligent infrastructures. In *Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES 2023)*.

National Institute of Standards and Technology (2024). Module-lattice-based digital signature standard. Technical Report FIPS 204, U.S. Department of Commerce.

National Institute of Standards and Technology (2025). Ascon-based lightweight cryptography standards for constrained devices: Authenticated encryption, hash, and extendable output functions. Technical Report NIST SP 800-232, U.S. Department of Commerce.

Nguyen, N. H., Le, D. H. A., Le, V. T. D., Nguyen, V. T., Vu, T. H., Pham, H. L., and Nakashima, Y. (2024). LI-RV: A fast and efficient RISC-V based coprocessor for lightweight cryptography. In *2024 21st International SoC Design Conference (ISOCC)*.

Ortega L., K. D. and Dominguez Perez, L. J. (2021). Implementing CRYSTAL-Dilithium on FRDM-K64. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372:n71.

Qiao, Z., Liu, Y., Zhou, Y., Ming, J., Jin, C., and Li, H. (2023). Practical public template attacks on CRYSTALS-Dilithium with randomness leakages. *IEEE Transactions on Information Forensics and Security*, 18.

Sajimon, P. C., Jain, K., and Krishnan, P. (2022). Analysis of post-quantum cryptography for Internet of Things. In *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*.

Schöffel, M., Lauer, F., Rheinländer, C. C., and Wehn, N. (2022). Secure IoT in the era of quantum computers: Where are the bottlenecks? *Sensors*, 22(7):2484.

Seng, V. L. K., Au, T. W., and Newaz, S. H. S. (2024). Blockchain-based key state management for IoT devices in post-quantum era. In *2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*.

Señor, J., Portilla, J., and Mujica, G. (2022). Analysis of the NTRU post-quantum cryptographic scheme in constrained IoT edge devices. *IEEE Internet of Things Journal*, 9(19).

Señor, J., Portilla, J., and Portela-García, M. (2024). Performance analysis of postquantum cryptographic schemes for securing large-scale wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 20(10).

Shehzadi, S., Whaley, N., Khalid, A., Ghafoor, A., Arshad, S., Ul Islam, F., and O'Neill, M. (2025). Kyber-KEM-Ascon: Benchmarking a lightweight post-quantum KEM on IoT devices. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*.

Wang, R., Gärtner, J., and Dubrova, E. (2025). Decompressing Dilithium's public key with fewer signatures using side channel analysis. In *2025 IEEE 55th International Symposium on Multiple-Valued Logic (ISMVL)*.

Wu, Y. (2024). Hardware-software co-design of the lightweight authenticated encryption algorithm ASCON for Internet of Things. In *2024 International Conference on Information Technology, Communication Ecosystem and Management (ITCEM)*.

Zheng, J., He, F., Shen, S., Xue, C., and Zhao, Y. (2022). Parallel small polynomial multiplication for Dilithium: A faster design and implementation. In *Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC 2022)*, pages 304–317.

A Included primary studies and quality assessment

Table 2 lists the 40 included primary studies with their focus, strategy category (B: benchmark or evaluation; H: hardware acceleration; S: software optimisation; P: protocol or application adaptation; L: lightweight-cryptography primitive evidence), and quality-assessment scores (Y: yes; P: partial).

Table 2: Included primary studies with categories and QA scores.

ID	Study	Platform / focus	Cat.	QA1	QA2	QA3
P01	Greconici et al. (2021)	Cortex-M3/M4 Dilithium	S	Y	Y	Y
P02	Ortega L. and Dominguez Perez (2021)	FRDM-K64 Dilithium	B	Y	Y	Y
P03	Kumar et al. (2020)	Secure boot, HW/SW	H	Y	Y	Y
P04	Becker et al. (2022)	Cortex-A72 / M1	S	Y	Y	Y
P05	Kim et al. (2022)	ARMv8 MCU	S	Y	Y	Y
P06	Abdulrahman et al. (2022)	Cortex-M4	S	Y	Y	Y
P07	Zheng et al. (2022)	Small-poly. mult.	S	Y	Y	Y
P08	Bos et al. (2022)	Memory-constrained M4	S	Y	Y	Y
P09	Howe and Westerbaan (2023)	Cortex-M7 Dil./Falcon	B	Y	Y	Y

ID	Study	Platform / focus	Cat.	QA1	QA2	QA3
P10	Gupta et al. (2023)	RISC-V secure boot	H	Y	Y	Y
P11	Karl et al. (2024)	RISC-V ISA extensions	H	Y	Y	Y
P12	Beckwith et al. (2023)	Dil./Falcon accelerators	H	Y	Y	Y
P13	He et al. (2023)	RBLWE FPGA	H	Y	Y	P
P14	Huang et al. (2024)	Keccak on ARMv7-M	S	Y	Y	Y
P15	Elsadek et al. (2022b)	LWC finalists ASIC	L	Y	Y	P
P16	Elsadek et al. (2022a)	LWC energy ASIC	L	Y	Y	P
P17	Bîrleanu and Bizon (2022)	LWC finalists analysis	L	P	P	P
P18	Elsadek and Tawfik (2024)	Ascon programmable arch.	L	Y	Y	P
P19	El-Hadedy et al. (2024)	Ascon permutation	L	Y	Y	P
P20	Malal (2024)	Ascon FPGA	L	Y	Y	P
P21	Elsadek et al. (2025)	Ascon ASIC	L	Y	Y	P
P22	Wu (2024)	Ascon co-design	L	P	Y	P
P23	Shehzadi et al. (2025)	Kyber+Ascon, M4	L, S	Y	Y	Y
P24	Sajimon et al. (2022)	PQC for IoT analysis	B	P	Y	P
P25	Señor et al. (2022)	NTRU on edge devices	B	Y	Y	P
P26	Malina et al. (2023)	Intelligent infrastructures	B, P	Y	Y	P
P27	Abbasi et al. (2025)	Heterogeneous benchmark	B	Y	Y	P
P28	Señor et al. (2024)	Large-scale WSN	B	Y	Y	P
P29	Castiglione et al. (2025)	ESP32 blockchain	P	Y	Y	Y
P30	Figlarz and Hessel (2022)	Autonomous vehicles	P	P	Y	P
P31	Liu et al. (2024)	BLE key exchange	P	Y	Y	P
P32	Kim and Seo (2025)	MQTT on 8-bit AVR	P	Y	Y	Y
P33	Kim et al. (2025)	Web-of-Things delivery	P	Y	Y	P
P34	Lawo et al. (2024)	DPU network stack	P, H	Y	Y	Y
P35	Ahola et al. (2025)	Boot security restoration	P	Y	Y	P
P36	Seng et al. (2024)	Key-state management	P	P	Y	P
P37	Nguyen et al. (2024)	RISC-V LWC coprocessor	H, L	Y	Y	P
P38	Schöffel et al. (2022)	Cortex-M4 TLS system, energy	B, P	Y	Y	Y
P39	Fitzgibbon and Ottaviani (2024)	Raspberry Pi 4 benchmark	B	Y	Y	Y
P40	Halak et al. (2024)	Pi 3b+/Pico W energy testbed	B	Y	Y	Y

B Search strings

The base Boolean string, adapted to each database's syntax, was: ("CRYSTALS-Dilithium" OR "Dilithium" OR "ML-DSA") AND ("performance" OR "benchmarking" OR "optimization" OR "implementation") AND ("IoT" OR "Internet of Things" OR "embedded system" OR "microcontroller" OR "Cortex-M" OR "RISC-V" OR "resource-constrained"). The companion lightweight-cryptography string substituted the first clause with ("Ascon" OR "lightweight cryptography" OR "LWC" OR "SHAKE" OR "Keccak" OR "SHA-3"). Metadata-only fields (title, abstract, keywords) were searched in Scopus; full-text search was used in IEEE Xplore, the ACM Digital Library, and SpringerLink with results limited to 2020 onwards. The July 2026 update pass re-executed the conjunction of the two algorithm clauses (("Dilithium" OR "ML-DSA") AND ("Ascon" OR "lightweight hash")) across all four databases and Google Scholar. The complete per-database search log is available from the authors.