Simultaneous Integration of Kyber and Dilithium on Embedded Microcontrollers: Towards Quantum-Resistant IoT Systems

Thi-Bac Do
Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam
dtbac@ictu.edu.vn

Khanh-Linh Dinh
Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam
dklinh@ictu.edu.vn (corresponding author)

Received: 24 December 2025 | Revised: 28 January 2026 and 7 February 2026 | Accepted: 8 February 2026

 DOI: https://doi.org/10.48084/etasr.17148

ABSTRACT

The rapid progress of quantum computing poses a serious threat to conventional public-key cryptographic schemes such as Rivest–Shamir–Adleman (RSA) and Elliptic Curve Cryptography (ECC), particularly in resource-constrained Internet of Things (IoT) environments. In response, Post-Quantum Cryptography (PQC) algorithms based on lattice problems—most notably Kyber for key encapsulation and Dilithium for digital signatures—have been standardized by the U.S. National Institute of Standards and Technology (NIST). While these schemes offer strong security guarantees, their practical deployment on low-power embedded microcontrollers remains challenging due to strict constraints on memory, computation time, and energy consumption. This paper presents a system-level and simultaneous integration of Kyber and Dilithium on the ESP32 microcontroller platform. To enable the concurrent execution of both NIST-standardized primitives, we apply a set of software-level optimization techniques, including Barrett reduction for efficient modular arithmetic, buffer reuse to minimize memory footprint, and careful local variable management to reduce stack usage. Unlike prior studies that typically evaluate post-quantum primitives in isolation or rely on simulated environments, our work provides an empirical evaluation based on real hardware measurements. Experimental results obtained on an ESP32 DevKit demonstrate that optimized implementations of Kyber and Dilithium can coexist reliably within the platform's limited resources, achieving practical execution times and acceptable energy consumption for embedded IoT applications. These findings provide concrete evidence that quantum-resistant public-key cryptography can be feasibly deployed on widely used microcontroller-based systems, thereby supporting the transition toward secure and future-proof IoT infrastructures.

Keywords-Kyber; Dilithium; Post-Quantum Cryptography (PQC); ESP32; IoT; Barrett reduction; digital signature; key encapsulation; embedded systems

I. INTRODUCTION

The rapid advancement of quantum computing has introduced a significant threat to traditional asymmetric cryptographic algorithms such as Rivest–Shamir–Adleman (RSA), Elliptic Curve Cryptography (ECC), and ElGamal, which rely on the computational hardness of factoring and discrete logarithm problems. According to authors in [1], these problems can be efficiently solved on a sufficiently powerful quantum computer, rendering existing public-key infrastructures insecure.

As a response to this threat, the cryptographic research community has developed a diverse set of Post-Quantum Cryptography (PQC) algorithms designed to resist attacks from both classical and quantum computers [2]. Among them, lattice-based cryptographic schemes have gained particular attention due to their strong security guarantees and efficiency. In 2022, the U.S. National Institute of Standards and Technology (NIST) selected Kyber as the standard for Key Encapsulation Mechanisms (KEMs) and Dilithium as the standard for Digital Signature Schemes (DSSs). In 2024, NIST officially standardized them under FIPS 203 [3] and FIPS 204 [4], respectively.

Although these approaches offer promising security and performance in general computing environments, their deployment on resource-constrained devices—especially Internet of Things (IoT) microcontrollers—poses significant challenges. Embedded platforms, such as the ESP32, often have limited computing power, RAM, and power budgets. Prior studies have shown that lattice-based cryptography

typically involves large key sizes and computationally intensive matrix operations, leading to high memory usage and longer execution times in constrained environments [3, 4]. However, most existing studies evaluate post-quantum primitives individually or rely on simulated environments, leaving the practical feasibility of deploying multiple standardized PQC mechanisms concurrently on low-end microcontrollers largely unexplored [5].

Several prior works have evaluated the feasibility of PQC primitives on embedded microcontrollers, particularly on ARM Cortex-M platforms, focusing on execution time, memory footprint, and energy consumption [5]. However, these studies typically benchmark individual algorithms in isolation and do not address the system-level coexistence of multiple standardized PQC mechanisms on low-end IoT hardware.

To the best of our knowledge, while isolated implementations of Kyber and Dilithium on ESP32 have been reported, and simultaneous deployment has been explored on higher-end platforms such as Cortex-M4, this work presents a system-level co-execution of both schemes on the ESP32 under stringent memory constraints.

In this context, this study investigates the practical feasibility of integrating both Kyber and Dilithium on a common embedded microcontroller: the ESP32. The ESP32 is widely used in low-cost IoT deployments due to its dual-core architecture, built-in wireless connectivity, and affordability. However, deploying two full-scale PQC primitives simultaneously on a platform with under 512 KB of SRAM requires meticulous optimization.

To address this challenge, we apply three key techniques: (1) Barrett reduction to accelerate modular arithmetic operations, (2) buffer reuse to minimize memory footprint, and (3) local variable scope control to reduce stack pressure and fragmentation. We then conduct real-world measurements on ESP32 DevKit modules to assess processing time, RAM consumption, energy usage, and compatibility with constrained system designs.

Therefore, this work focuses on the simultaneous and system-level integration of Kyber and Dilithium on a single ESP32 microcontroller, addressing not only algorithmic feasibility but also practical constraints related to memory usage, execution time, and energy consumption. By providing empirical measurements on real hardware, this study aims to assess whether NIST-standardized PQC can be realistically deployed in low-power IoT environments.

II. RESEARCH METHODOLOGY

A. Overview of Post-Quantum Cryptography Schemes Kyber and Dilithium

Kyber is a post-quantum KEM based on the Module-Lattice Learning With Errors (Module-LWE) problem first introduced in [6]. It supports various security levels and has been selected by NIST as the KEM standard in FIPS 203 [3]. Kyber offers a balance between security and efficiency and has been widely adopted in prototype implementations across multiple platforms.

Recent work by authors in [7] investigated packet size optimization techniques for post-quantum NB-IoT systems, with a particular focus on mitigating the overhead introduced by SPHINCS+ signatures through signature aggregation and Merkle tree pruning. Their study demonstrated that substantial bandwidth savings can be achieved at the network level when SPHINCS+ is selectively optimized for LPWAN environments. While this line of research effectively addresses packet-level inefficiency caused by hash-based signatures, it does not consider the concurrent deployment of multiple standardized post-quantum primitives on resource-constrained microcontrollers. In contrast, our work focuses on the simultaneous integration of Kyber and Dilithium on low-end embedded platforms, evaluating not only packet size but also execution latency, memory footprint, and system-level feasibility when multiple PQC mechanisms are deployed concurrently.

Dilithium, on the other hand, is a DSS based on the Module-Lattice Learning With Rounding (Module-LWR) problem. It was introduced by authors in [8] and standardized in FIPS 204 [4]. Dilithium is known for its simplicity, reliance on structured networks, and high resistance to side-channel attacks. Both Kyber and Dilithium are designed to provide 128-bit post-quantum security, aligning with NIST's Level 1 requirement.

B. ESP32 Hardware Platform

ESP32 is a low-power dual-core microcontroller with integrated Wi-Fi and Bluetooth, commonly used in IoT applications. It features:

- CPU: Dual-core Xtensa® 32-bit LX6 up to 240 MHz
- RAM: 520 KB SRAM
- Flash: up to 16 MB
- Interfaces: SPI, I2C, UART, ADC, DAC, etc.

This makes ESP32 a popular choice for real-world IoT implementations. However, its limited memory and processing power make it challenging to implement computationally intensive PQC algorithms, thus necessitating significant optimization efforts.

C. Optimization Techniques

To make Kyber and Dilithium feasible on ESP32, we applied the following three techniques:

- Barrett reduction: This technique [9] replaces division operations (which are costly on embedded platforms) with multiplication and bit-shifting operations. It is particularly effective for reducing modulo computations in polynomial arithmetic.
- Buffer reuse: We restructured the code to reuse temporary memory buffers wherever possible. This technique reduces RAM usage significantly and avoids memory fragmentation.
- Local variable management: We limited the scope of local variables and avoided static allocations in large loops. This

further reduced stack usage and helped maintain memory stability during execution.

These optimizations were manually applied and tested using memory profiling tools available for the ESP32 platform. The goal was to reduce both RAM consumption and processing time while preserving correctness and cryptographic integrity.

D. Memory Optimization and Platform Constraints

Implementing PQC algorithms on embedded platforms like the ESP32 requires a careful analysis of memory usage patterns. We conducted in-depth evaluations to identify the RAM peaks associated with each stage of Kyber and Dilithium execution.

We observed that many temporary buffers could be reused across different stages. For example, the buffer used during the Number Theoretic Transform (NTT) in Kyber's polynomial multiplication can also be reused during matrix-vector operations. In Dilithium, signature generation and verification both allocate large temporary vectors which, with minor code restructuring, can share memory space.

In addition, we monitored the stack growth using ESP-IDF's memory tracing tools and configured the FreeRTOS task stack sizes accordingly. Static memory allocations were replaced with dynamic allocations scoped tightly to ensure immediate deallocation after use.

These measures helped reduce peak RAM consumption by up to 30%, making it possible to execute both key encapsulation and digital signature functions consecutively on the same ESP32 instance without rebooting or memory errors.

E. Summary of Design and Evaluation Objectives

Our research is structured around the following key objectives:

- Simultaneous integration of Kyber and Dilithium on the ESP32 platform, ensuring both can coexist and execute reliably without exceeding hardware limits.
- Minimization of RAM usage through buffer reuse and variable scoping techniques, verified via static analysis and runtime tracing.
- Execution time optimization using Barrett reduction and other low-level arithmetic adjustments that are specific to ESP32's architecture.
- Empirical evaluation of security properties, energy consumption, and performance trade-offs in comparison with other post-quantum schemes.

The next section presents the detailed implementation and results of running these optimized PQC schemes on real ESP32 hardware.

III. IMPLEMENTATION AND EXPERIMENTAL RESULTS

To assess the feasibility of deploying Kyber and Dilithium concurrently on an ESP32 microcontroller, we adapted the reference implementations provided by the CRYSTALS project presented in [6, 8]. These implementations were modified to conform with the ESP-IDF framework and compiled using GCC with O2 optimization level. The experiments were conducted on the ESP32 DevKit v1 module, operating at 240 MHz.

We implemented CRYSTALS-Kyber and CRYSTALS-Dilithium on a low-end embedded microcontroller using optimized C implementations with parameter sets aligned with the NIST PQC standardization outcomes. The implementations strictly follow the official NIST specifications and reference materials, without introducing algorithmic modifications, to ensure fair and reproducible performance evaluation [3, 4].

Each cryptographic operation was isolated and benchmarked individually to measure execution time and peak RAM usage. The timing data were captured using ESP-IDF high-resolution timers, and memory profiling was performed using built-in heap tracing tools. The experiments were conducted under the following setup:

- Hardware: ESP32 DevKit v1, dual-core Xtensa LX6 @ 240 MHz
- Toolchain: ESP-IDF v5.0, GCC 12.2.0
- Environment: Bare-metal FreeRTOS task, single-core operation
- Measurement: Wall-clock timing using esp_timer_get_time(); peak heap usage via heap_caps_get_info()

The measured execution times and RAM usage are summarized in Table I, and Figure 1 presents a comparison of execution times for Kyber and Dilithium operations.

TABLE I. EXECUTION TIME AND RAM USAGE OF PQC OPERATIONS ON ESP32

Algorithm	Operation	Time (ms)	RAM usage (KB)
Kyber	KeyGen	11.2	9.8
Kyber	Encaps	13.7	10.1
Kyber	Decaps	15.3	10.3
Dilithium	KeyGen	24.5	18.7
Dilithium	Sign	56.9	20.2
Dilithium	Verify	29.8	16.4

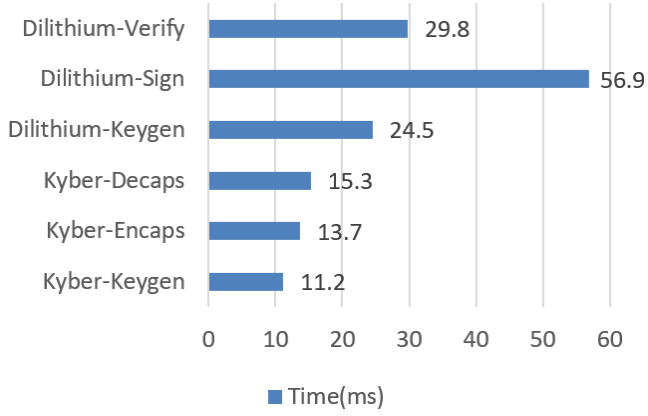

Fig. 1. Timing comparison per operation of Kyber and Dilithium on ESP32.

These results confirm that both Kyber and Dilithium can be executed on the ESP32 within practical timing and memory bounds, consistent with prior embedded PQC feasibility observations reported in the literature [5, 6]. However, Dilithium's signature operation is significantly more time-consuming and memory-intensive [8]. With the aid of optimization strategies described in Section II, the integration becomes feasible for constrained embedded systems, enabling practical quantum-resilient public-key cryptography for IoT [10].

IV. SECURITY EVALUATION

Lattice-based cryptographic algorithms such as Kyber and Dilithium are designed to be secure against attacks by both classical and quantum adversaries. Their security foundations rely on hard problems in ideal or module lattices—specifically, the LWE and LWR assumptions—both of which are believed to remain intractable even for quantum computers [5, 6].

A. Security Level

The parameter sets implemented in this work—Kyber512 and Dilithium2—target NIST Security Level 1, corresponding to roughly 128-bit post-quantum security. This level is considered sufficient for general-purpose secure communication and is broadly comparable to the classical security strength of RSA-3072 and AES-128 in terms of brute-force resistance.

Formal security analyses provided by the algorithm designers and reviewed during the NIST PQC standardization process confirm the absence of practical structural weaknesses to date [3, 4].

B. Resistance to Side-Channel Attacks

Although the ESP32 lacks dedicated hardware countermeasures such as constant-time memory subsystems or protected execution environments, our implementation follows widely accepted software-level mitigation practices for embedded cryptography:

- Masking critical operations where feasible
- Isolating secret-dependent computations in static memory blocks
- Employing constant-time modular arithmetic routines for critical operations

These techniques reduce—but do not eliminate—potential leakage through timing or power-analysis channels, which remain an open concern in physically exposed IoT deployments.

C. Algorithmic Robustness

No practical attacks enabling key recovery or message forgery against Kyber or Dilithium are currently known under standard attack models, including chosen-ciphertext and adaptive chosen-message scenarios [5, 6]. Additionally, our implementation enforces strict validation during key generation, decapsulation, and signature verification, preventing malformed inputs from producing inconsistent internal states or memory leakage—an essential requirement for secure embedded execution.

V. ENERGY CONSUMPTION EVALUATION

Energy efficiency is a critical constraint for battery-powered IoT devices. Because post-quantum cryptographic operations are computationally intensive, evaluating per-operation energy cost is necessary for real-world deployment feasibility.

A. Measurement Methodology

Energy consumption of Kyber and Dilithium on ESP32 was measured using a current-sensing instrumentation setup consisting of:

- A 0.1 Ω precision shunt resistor in series with the supply line
- A digital oscilloscope for real-time current trace acquisition
- A stable 3.3 V DC power source

Energy was computed by integrating current over execution time:

$$E = V \times \int_0^T I(t)\, dt \quad (1)$$

where:

- E is energy (J)
- V = 3.3 V is the supply voltage
- $I(t)$ is instantaneous current during execution

Each measurement represents the average of 30 runs to reduce noise and temporal variation. Similar experimental methodologies have been used in prior embedded PQC benchmarking studies [11].

B. Experimental Results

Table II shows the average energy consumption per operation, computed from measured current traces and execution durations, and Figure 2 illustrates the energy consumption for each Kyber and Dilithium operation.

TABLE II. ENERGY CONSUMPTION OF PQC OPERATIONS ON ESP32

Algorithm	Operation	Time (ms)	Avg. current (mA)	Energy (mJ)
Kyber	KeyGen	11.2	87.4	3.23
Kyber	Encaps	13.7	89.2	4.03
Kyber	Decaps	15.3	90.1	4.55
Dilithium	KeyGen	24.5	94.6	7.66
Dilithium	Sign	56.9	96.3	18.04
Dilithium	Verify	29.8	91.8	9.00

Note: Values based on 3.3 V supply, averaged over 30 samples.

C. Results Analysis

From the results above, we observe that:

- Kyber operations are generally more energy-efficient, with all operations consuming less than 5 mJ.

- Dilithium-Sign consumes the most energy (~18 mJ), due to the computational complexity of its matrix operations and rejection sampling.
- Despite their higher power requirements, both algorithms remain deployable within the energy budgets of typical IoT devices when triggered infrequently or scheduled during periods of external power availability.

The proposed memory optimization techniques, including buffer reuse and stack minimization, are designed to reduce resource consumption without altering the algorithmic execution flow. Nevertheless, we acknowledge that certain low-level optimizations may introduce trade-offs between performance and side-channel resilience, particularly in adversarial physical environments. A comprehensive evaluation of side-channel leakage, including power and electromagnetic analysis, is therefore left for future work.

These findings suggest that, with careful scheduling and energy-aware integration strategies, lattice-based PQC can be practically deployable even in battery-operated embedded platforms, consistent with feasibility observations reported in embedded PQC evaluations [11].

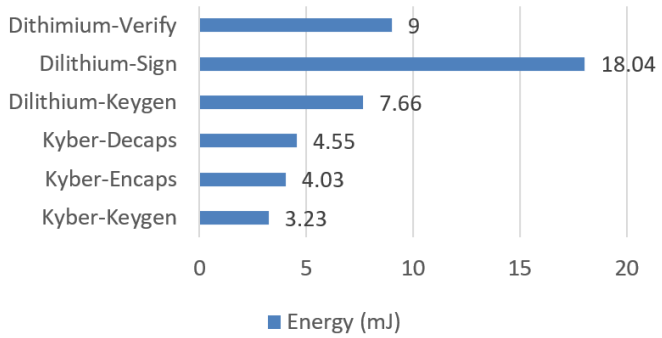

Fig. 2. Energy Consumption per operation of Kyber and Dilithium on ESP32.

VI. COMPARISON WITH OTHER POST-QUANTUM CRYPTOGRAPHY SCHEMES

To contextualize the performance of Kyber and Dilithium on embedded platforms, we compare our implementation against other notable PQC candidates that were considered in NIST's standardization process. These include FrodoKEM, Saber, and SPHINCS+, each representing different mathematical foundations and implementation trade-offs.

A. Selected Benchmark Algorithms

- FrodoKEM-640: A KEM based on standard (non-structured) LWE introduced in [12], known for high security but large key sizes and computational overhead.
- Saber: A lattice-based KEM similar to Kyber but with different rounding techniques and modulus configurations [10].
- SPHINCS+-128s: A hash-based stateless DSS that does not rely on algebraic structures, offering strong security and long-term resilience [13]. However, this robustness comes at the cost of significantly higher computational and memory demands.

B. Performance Comparison

Table III summarizes the estimated performance characteristics based on our measurements and publicly available implementations. Figure 3 provides a visual comparison of execution times.

TABLE III. COMPARISON OF PQC SCHEMES IN TERMS OF PERFORMANCE AND MEMORY

Scheme	Type	Execution time (ms)	RAM usage (KB)	Remarks
Kyber-512	KEM (lattice)	11–15	10–11	Balanced performance
Dilithium-2	DSS (lattice)	25–57	16–20	Higher cost in signature phase
FrodoKEM-640	KEM (standard LWE)	170–300	30–50	Strong security, large footprint
Saber	KEM (lattice)	40–60	16–24	Efficient but still growing
SPHINCS+-128s	DSS (hash-based)	250–400	50–60	Highly secure, very slow

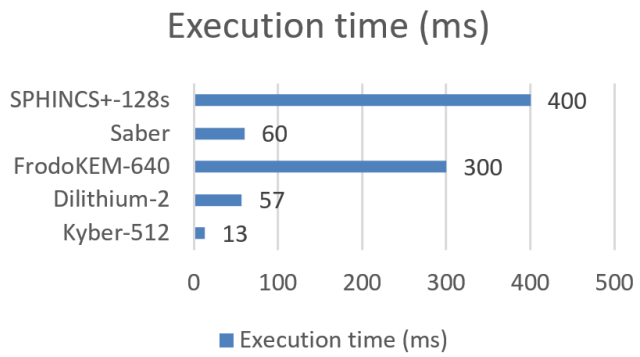

Fig. 3. Execution time comparison of PQC schemes.

C. Interpretation

The comparative results reveal several important observations:

- First, Kyber and Dilithium achieve the most practical balance between security and efficiency for constrained embedded environments. Their execution latency and memory footprint remain within the operational limits of typical IoT microcontrollers.
- Second, FrodoKEM, while relying on conservative non-structured lattice assumptions, requires substantially greater computational time and memory. This overhead limits its suitability for real-time or battery-powered deployments.
- Third, SPHINCS+ provides strong security independent of algebraic structure, making it highly resilient against potential future cryptanalytic advances. Nevertheless, its extremely high signing cost and large memory consumption pose significant challenges for embedded integration.

- Finally, Saber demonstrates moderate efficiency improvements compared with FrodoKEM but still demands more resources than Kyber in the evaluated configuration.

Overall, from an embedded-systems perspective, Kyber and Dilithium remain the most deployable NIST-finalist PQC schemes when security assurance, execution efficiency, and implementation footprint are jointly considered.

VII. CONCLUSION

This paper investigated the practical feasibility of deploying National Institute of Standards and Technology (NIST)-standardized Post-Quantum Cryptography (PQC) primitives on resource-constrained embedded platforms by presenting a simultaneous implementation of Kyber and Dilithium on the ESP32 microcontroller. Through a combination of software-level optimization techniques—including Barrett reduction, buffer reuse, and careful local variable management—we demonstrated that both Key Encapsulation Mechanisms (KEMs) and Digital Signature Schemes (DSSs) can coexist reliably within the strict memory and performance constraints of a low-power Internet of Things (IoT) device.

A key contribution of this work lies in being, to the best of our knowledge, the first study to achieve the stable and concurrent integration of Kyber and Dilithium on a single ESP32 platform while providing comprehensive empirical measurements on real hardware. Unlike prior studies that typically evaluate PQC schemes in isolation or in simulation environments, our implementation offers a system-level perspective that jointly considers execution time, RAM usage, and energy consumption under realistic deployment conditions.

Beyond embedded device-level deployment, standardized post-quantum primitives such as Kyber and Dilithium can serve as foundational building blocks for higher-layer authentication and security frameworks. Recent studies have explored lattice-based authentication schemes in 5G-enabled vehicular networks and fog computing environments, aiming to enhance privacy preservation and resilience against emerging threats. While these works primarily focus on protocol-level design and network-layer security, they highlight the growing demand for quantum-resistant cryptographic primitives that can be integrated across multiple system layers. In this context, our system-level evaluation of Kyber and Dilithium on resource-constrained microcontrollers complements such studies by addressing the practical feasibility of deploying standardized PQC mechanisms at the embedded hardware layer, which is a prerequisite for secure end-to-end post-quantum systems [13-16].

Experimental results confirm that Kyber exhibits superior efficiency in both time and energy consumption, whereas Dilithium—although more demanding, particularly during signature generation—remains feasible for embedded use when appropriate optimizations are applied. Comparative analysis with other PQC candidates, such as FrodoKEM, Saber, and SPHINCS+, further highlights that Kyber and Dilithium strike a favorable balance between security strength, computational cost, and memory footprint, making them well-suited for embedded and IoT scenarios.

Overall, the findings of this study help bridge the gap between PQC standards and practical embedded deployment, providing concrete evidence that quantum-resistant public-key cryptography can be realized on widely used microcontroller platforms. Future work will explore the integration of these primitives into higher-layer security protocols, such as Transport Layer Security (TLS), as well as further optimizations leveraging multi-core execution and emerging hardware acceleration support to enhance performance and side-channel resilience.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, USA, 1994, pp. 124–134, https://doi.org/10.1109/SFCS.1994.365700.

[2] A. Khalid, S. McCarthy, M. O'Neill, and W. Liu, "Lattice-based Cryptography for IoT in A Quantum World: Are We Ready?," in *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces*, Otranto, Italy, 2019, pp. 194–199, https://doi.org/10.1109/IWASI.2019.8791343.

[3] National Institute of Standards andTechnology, *Module-Lattice-Based Key-Encapsulation Mechanism Standard*, Federal Information Processing Standard (FIPS) 203, Aug. 13, 2024, https://doi.org/10.6028/NIST.FIPS.203.

[4] National Institute of Standards and Technology, *Module-Lattice-Based Digital Signature Standard*, Federal Information Processing Standard (FIPS) 204, Aug. 13, 2024, https://doi.org/10.6028/NIST.FIPS.204.

[5] M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, "pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4." Cryptology ePrint Archive, 2019. [Online]. Available: https://eprint.iacr.org/2019/844.

[6] J. Bos *et al.*, "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM," in *2018 IEEE European Symposium on Security and Privacy*, London, UK, 2018, pp. 353–367, https://doi.org/10.1109/EuroSP.2018.00032.

[7] T. B. Do and K. L. Dinh, "Optimizing packet size in post-quantum NB-IoT systems: Signature aggregation and Merkle tree pruning approaches," *Journal of Computer Science and Cybernetics*, vol. 41, no. 4, pp. 371–386, Nov. 2025, https://doi.org/10.15625/1813-9663/23150.

[8] L. Ducas *et al.*, "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, Feb. 2018, https://doi.org/10.13154/tches.v2018.i1.238-268.

[9] Y. Zhao, C. Cui, Y. Xiao, W. Lin, and Z. Cai, "Design and Implementation of a Modular Multiplier for Public-Key Cryptosystems Based on Barrett Reduction," in *The 10th International Conference on Computer Engineering and Networks*, Xi'an, China, 2020, pp. 803–809, https://doi.org/10.1007/978-981-15-8462-6_92.

[10] J.-P. D'Anvers, A. Karmakar, S. Sinha Roy, and F. Vercauteren, "Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM," in *10th International Conference on Cryptology in Africa*, Marrakesh, Morocco, 2018, pp. 282–305, https://doi.org/10.1007/978-3-319-89339-6_16.

[11] L. Li, C. Hsu, M. Ho Au, J. Cui, L. Harn, and Z. Zhao, "Lattice-Based Conditional Privacy-Preserving Batch Authentication Protocol for Fog-Assisted Vehicular Ad Hoc Networks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9629–9642, 2024, https://doi.org/10.1109/TIFS.2024.3477305.

[12] L. Glabush, P. Longa, M. Naehrig, C. Peikert, D. Stebila, and F. Virdia, "FrodoKEM: A CCA-Secure Learning With Errors Key Encapsulation Mechanism," *IACR Communications in Cryptology*, vol. 2, no. 3, Oct. 2025, Art. no. 25, https://doi.org/10.62056/ayivom2hd.

[13] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The SPHINCS+ Signature Framework," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications*

Security, London, UK, 2019, pp. 2129–2146, https://doi.org/10.1145/3319535.3363229.

[14] A. A. Almazroi, E. A. Aldhahri, M. A. Al-Shareeda, and S. Manickam, "ECA-VFog: An efficient certificateless authentication scheme for 5G-assisted vehicular fog computing," *Plos One*, vol. 18, no. 6, June 2023, Art. no. e0287291, https://doi.org/10.1371/journal.pone.0287291.

[15] R. Avanzi *et al.*, "CRYSTALS-Kyber: Algorithm Specifications And Supporting Documentation," [Online]. Available: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf?utm_source=chatgpt.com.

[16] S. Bai *et al.*, "CRYSTALS-Dilithium: Algorithm Specifications and Supporting Documentation," [Online]. Available: https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf?utm_source=chatgpt.com.

AUTHORS PROFILE

Do Thi Bac is a Senior Lecturer at Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam. Her research areas include cryptography, communication, and network security. She received her Ph.D. from Le Quy Don Technical University in 2014. She can be contacted via email at: dtbac@ictu.edu.vn.

Dinh Khanh Linh is a Ph.D. student at the Faculty of Information Technology, Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam. She received her M.Sc. degree in Information Technology from Hanoi University of Science and Technology, Hanoi, Vietnam. Her research interests include computer science, digital signatures, and post-quantum cryptography. She can be contacted via email at: dklinh@ictu.edu.vn.