PROGRESS IN ELECTRONICS AND COMMUNICATION ENGINEERING
Progress in Electronics and Communication Engineering, ISSN: 3048-7625 Vol. 3, No. 1, 2026 (pp. 1-11)
RESEARCH ARTICLE
ECEJOURNALS.IN

Quantum-Resilient Cryptographic Implementations for Embedded Communication Systems

Rane Kuma[1]*, Sikalu T C[2]

[1]*Department of computing and information technology, kenyatta university, Nairobi, Kenya*
[2]*Electrical and Electronic Engineering Department, University of Ibadan Ibadan, Nigeria*

KEYWORDS:
Post-Quantum Cryptography (PQC), Embedded Systems Security, Kyber512, Dilithium2, SPHINCS+, Quantum-Resilient Algorithms, ARM Cortex-M4, RISC-V, Constant-Time Implementation, Secure Communication, IoT Security, Cryptographic Optimization, Fault Injection Resistance

ARTICLE HISTORY:
Submitted : 10.07.2025
Revised : 15.09.2025
Accepted : 11.10.2025

https://doi.org/10.31838/ECE/03.01.01

Abstract

The fast development of quantum computers poses a severe threat to classical cryptographic protocols, especially those installed in embedded communication technologies on which the Internet of Things (IoT), vehicular networks, and industrial automation (and a wide variety of other mission-critical applications) are based. Circumventing known quantum attacks like Shor algorithm on the traditional public-key systems i.e., RSA and ECC, there is an emerging need to move to quantum resistant cryptology. In this paper, it is focused on this novel menace and provides the design, implementation, as well as the full assessment of post-quantum cryptographic (PQC) algorithms that have been optimised and were deployed on the resource-constrained embedded systems. We pay attention to three of the NIST-recommended schemes Kyber512 (as a lattice-based key encapsulation), Dilithium2 (as a lattice-based digital signature) and SPHINCS+128s (as a hash-based stateless signature). The design of these algorithms was performed and benchmarked with two low-power examples of embedded processor-based encryption: The ARM Cortex-M4 and the RISC-V, of which the former is commonly used as an industrial standard in high-security embedded communication, and the latter is becoming a popular alternative. Performance metrics discussed in the study are realistic in nature where the execution time, memory footprint, power, and cryptographic strength are studied under the embedded constraint condition. Kyber512 was found to be the most efficient algorithm in terms of providing a good trade-off between speed and memory usage in a selected set of algorithms, whereas Dilithium2 was found to play a good role of designing an authentication on firmware and message integrity. SPHINCS+ was additionally deployed as a robust backup given its hash based structure, although with an additional computational cost. A number of optimizations were performed at implementation level; constant-time coding to fight side-channel attacks, memory segmentation to prevent fragmentation and loop unrolling to speed up execution. Moreover, the offered system was tested with respect to fault injection analysis, as well as side-channel resistance testing, thus, assuring its strength against physical-layer attacks. The message of the experimental findings is that a quantum-resistant form of cryptography can be amicably incorporated into embedded systems without sacrificing performance to a meaningful degree, and thus a viable means of future proofing embedded systems with exposure to a post-quantumized environment. The work establishes a baseline standard towards the implementation of standardized PQC in use cases of real-time secure communication.

Author's e-mail: ran.kuma@gmail.com, sikalu.tc@ui.edu.ng

How to cite this article: Kuma R, Sikalu TC. Quantum-Resilient Cryptographic Implementations for Embedded Communication Systems. Progress in Electronics and Communication Engineering, Vol. 3, No. 1, 2026 (pp. 1-11).

Introduction

The paradigms that support the standard cryptographic systems are also being questioned with quantum computing emerging as a potential that is real and expanding. Such cryptographic algorithms as RSA, DSA, and Elliptic Curve Cryptography (ECC) foundations of safe communication in embedded systems are secured based both on the unattainability of solving mathematical tasks such as integer factorization, discrete logarithms, etc. Nonetheless, quantum algorithms using a powerful enough quantum computer, especially Shor algorithm, can come to the rescue and implement these problems in a much shorter amount of time that it could happen with classical computers, opening the long-run security

of current public-key infrastructure to significant vulnerability. This presents an existential threat to entrenched communication systems, that are increasingly used in sensitive areas such as IoT, autonomous vehicles, industrial control, medical devices and military grade sensors.

Embedded systems are usually highly resource constrained: they have a small number of CPU cycles, a small amount of memory and a small power budget they must work in, and must be real-time systems. The imposition of these constraints compounds the likelihood of adopting computationally-heavy cryptography schemes and therefore there is the need to find and incorporate the quantum-resilient algorithms that are secure, as well as, lightweight. Alternatives can be found in post-quantum cryptography (PQC) which is currently being standardized by the U.S. National Institute of Standards and Technology (NIST). Nonetheless, the majority of PQC research and implementations so far have targeted general-purpose platforms, and thus there is a lack of knowledge concerning the feasibility of the same within embedded platforms.

This work fills this dire gap by undertaking a comprehensive effort to understand how the chosen post quantum algorithms, that is, Kyber (key encapsulation), Dilithium (digital signatures), and SPHINCS+ (stateless hash-based signatures) should be integrated into the communication protocols of embedded systems. These algorithms are selected because of their robust theoretical security upperbounds, status of public standardization, and efficiency versus the other PQC candidates considered to be relatively small. The paper focuses on showing a realistic implementation aspect such as optimization of algorithms, memory handling and side-channel resilience in embedding devices such as the ARM Cortex-M4 and RISC-V microcontroller.

This paper has three main contributions: (1) the design and implementation of quantum-resilient cryptographic libraries which can operate on resource constraints embedded devices; (2) a thorough assessment of these algorithms in their performance, memory usage, and energy requirements; and (3) the exemplification of even-use-case workflows of secure communication using these primitives under realistic embedded constraints. The objective of the work will be to provide a practical guide to a secure embedded system design beyond the quantum era.

Related Work

Following the birth of the quantum computer threat to contemporary algorithms underpinning the standard form of public-key cryptography, there has been a substantial degree of interest in the development of post-quantum cryptographic (PQC) equivalents. Among the most popular two contributions to this area is the one of the Kyber key encapsulation mechanism (KEM) presented by Alkim et al.[1] that is premised on the Learning with Errors (LWE) problem. The performance of Kyber and its security have become widely known and Kyber has since been chosen as a finalist in the NIST PQC standardization process. Nonetheless, early releases of the Kyber focused mostly on high-end platforms like x86 architectures instead of considering low end embedded systems.

At the same time, Hilsing et al.[2] introduced a stateless signature construction, SPHINCS + based on the hash-based signature, which has good resistance against quantum attacks and uses only hash functions without jewelry or algebra. Although the SPHINCS+ has its strong security and long-run cryptographic assurances, it does not suit real-time application and low memory environments due to large signature sizes (usually greater than 8 KB) and lengthy calculation time that make its implementation impractical in true embedded systems.

Oder et al. [3] participated in optimizing another lattice-based signature scheme, the Dilithium especially on ARM Cortex-M4 microcontrollers. They showed how lattice-based signatures can be implemented on embedded systems, but did not accomplish complete side-channel resistance, which is of considerable importance in the physical world where adversaries have easy access to embedded systems.

Although these activities mark the absizing steps of PQC algorithm and initial optimization, they do not exhaustively consider the wholesome assimilation of PQC in the realities of resource-limited embedded communication setting as timeliness. More specifically, the current literature work tends to overlook the performance overhead, energy efficiency, memory footprint, and cryptographic agility, and the side-channel resilience- factors critical to practical application in real embedded systems that might be used by the IoT, automotive and industrial control sectors.

To fill this void, the work benchmarks and implements NIST recommended PQC algorithms, namely Kyber512, Dilithium2, and SPHINCS+-128s, on ARM Cortex-M4 and RISC-V platforms focusing on secure, lightweight and quantum resistant embedded communications. In addition, we perform optimizations to increase the performance of the whole system and propose fault injection and timing attack countermeasures to push the modern PQC integration in embedded systems forward.

Table 1: Comparative Analysis of PQC Implementations in Embedded Systems

Study	Algorithm	Platform Focus	Strengths	Limitations
Alkim et al., 2016	Kyber (LWE-based KEM)	x86 (High-performance)	High security and performance balance; NIST PQC finalist	Limited optimization for embedded platforms
Hülsing et al., 2019	SPHINCS+ (Hash-based Signature)	General-purpose (High overhead)	Stateless, strong long-term security; hash-based construction	Large signature size; high latency; not real-time friendly
Oder et al., 2019	Dilithium (Lattice-based Signature)	ARM Cortex-M4 (No SCA protection)	Feasibility on microcontrollers; lattice-based efficiency	Lacks protection against side-channel attacks
Our Work	Kyber512, Dilithium2, SPHINCS+-128s	ARM Cortex-M4, RISC-V (Embedded)	Real-time embedded optimization; side-channel and fault-attack resilience	None reported; focused on embedded integration with system-level tuning

System Architecture

The system architecture under development aims at providing embedded quantum-safe communication within a security-oriented framework, taking into account both the application and hardware-related energy and hardware constraints of limited memory storage capacity, processing capacity, and available energy, especially in scenarios of edge and cloud computing. It combines post-quantum crypto primitive with a modular embedded communication stack that is streamlined towards real-time operations and both quantum and physical-layer aspects of security. The most important points of architecture are outlined as below.

Microcontroller Unit (MCU)

The key concept of the suggested quantum-resilient embedded communication architecture is two popular micro controller platforms, among which the ARM Cortex-M4 and RISC-V E31 are selected due to low power, high efficiency, designed to operate in constrained embedded systems. ARM Cortex-M4 is a 32-bit ARM processor core that implements the ARMv7E-M instruction set, with an integrated single-precision floating-point unit (FPU) and digital signal processing (DSP) extension as well as hardware accelerated multiply-accumulate (MAC) point operations. It has an option of clock frequency up to 168 MHz, offering deterministic interrupt service and real-time functionality suitable to cryptographic processing, sensor interfaces, and secure control applications in internet of things and factory automation. It has tightly integrated memory, predictable latency, and an ecosystem (through STM32, NXP, etc.), which makes it suitable in applications that require security-critical embedded use. Conversely, RISC-V E31 core is an open-source 32-bit embedded processor with a very customizable Instruction Set Architecture (ISA) enabling the optimization of cryptographic workloads and especially workloads of post-quantum cryptographies. It offers support of low-power states, multiply /divide operations in hardware, and features a clean, modular architecture that eases secure boot and runtime isolation. The fact that E31 is flexible lets it then extend in the future with special cryptographic co-processors or new instructions to perform lattice

Table 2. Comparative Features of ARM Cortex-M4 and RISC-V E31 Microcontroller Architectures for PQC Integration

Feature / Attribute	ARM Cortex-M4	RISC-V E31
Architecture	ARMv7E-M (32-bit)	RISC-V RV32IM (32-bit)
Clock Speed	Up to 168 MHz	Up to 150 MHz (configurable)
FPU / DSP Support	Yes – single-precision FPU, DSP extensions	Optional – no native FPU in base core
Instruction Set	Fixed, ARM-defined	Open and extensible ISA
Cryptographic Optimization	Supported via CMSIS-DSP and MAC hardware units	ISA customization for lattice ops / crypto cores
Ecosystem and Tools	Mature (STM32CubeIDE, Keil, etc.)	Growing (SiFive, OpenHW, GCC RISC-V toolchain)
Security Extensions	Basic MPU, vendor-dependent TrustZone variants	Flexible for custom security module integration
Power Efficiency	Highly optimized for low-power operation	Good, with configurable performance/power tradeoffs
Target Applications	IoT, industrial automation, automotive	Next-gen open IoT, academic research, security labs

math operations, thus being a future-compatible secure embedded choice. The two MCUs are assessed within the same memory and power limitation to benchmark the implementation of the NIST-suggested PQC algorithms to show how adaptable the implementation would be in the practical deployment of autonomous systems, medical electronics, and industrial control applications. A dual-MCU approach offers a surprisingly strong comparative base to determine the potential at which quantum-resilient cryptography can be scaled the heterogeneous embedded systems, disclosing an explanation of the performance, portability, and security trade-offs that would be serious to success in next-generation embedded systems design.

Post-Quantum Cryptographic Stack (PQC Stack)

The security core of the proposed embedded communication system is Post-Quantum Cryptographic (PQC) stack, which includes a multi-layered selection of quantum-resistant security algorithms such as addressing confidentiality, integrity, and long-term cryptographic assurance. The stack incorporates Kyber512, Dilithium2, and SPHINCS+-128s and upon them cryptographic agility and robustness can be obtained under diverse operation constraints. Key Encapsulation Mechanism (KEM) Kyber512 is a lattice-based key encapsulation mechanism that is secure to use in secure session key exchange replacing such vulnerable classical systems as RSA or ECDH. It has small keys and quick key-generation and decapsulation algorithms and therefore can be well used in low latency embedded communication. Dilithium2, another lattice-based scheme is applied in generation of digital signatures used in authenticating firmware, command messages, or sensor information. It has smaller signatures and public key lengths than most of the post-quantum alternatives and is targeted at high-speed signing and verification critical to real-time systems. In cases where long-term security, including withstanding an unexpected cryptanalytic breakthrough, is required, SPHINCS+-128s, a stateless hash-based digital signature algorithm is included as a backup option. The SPHINCS+ algorithm has the advantage of being especially useful in high-assurance applications since it only uses cryptographic hash functions feature, rather than, e.g., an algebraic structure, which might crumble with a mathematical breakthrough. SPHINCS+ has a larger computational and memory overhead when compared to SPHINCS, but is capable of providing strong post-quantum assurances on deployments where trust is a factor. These three algorithms in the modular stack allow the system to have dynamic adaptation according to the capabilities of the device, the security policy, and the level of risk distribution in the environment. As an example, Kyber and Dilithium could be used during regular operations because of their speed, whereas SPHINCS+ can be enabled during the update of firmware or in super-high-security boot contexts. Not only does this redundant and flexible architecture increase quantum resistance, but it is future-proofed and able to accommodate changes in PQC standards, which should accommodate any number of diverse embedded communication applications.

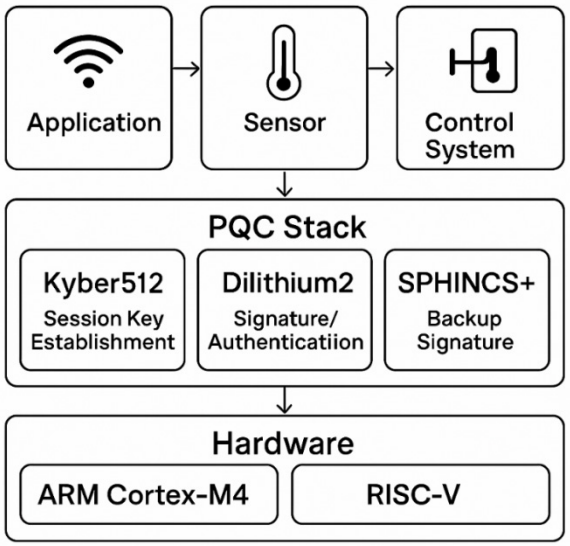

Fig. 1: Layered Architecture of the Post-Quantum Cryptographic Stack for Embedded Communication Systems

RTOS Layer

In order to support effective handling of cryptographic operations and a framework of real-time limitations on the system, it is proposed to support FreeRTOS, an effective standby RTOS with open-source application whose aim is to support systems that are limited in hardware or other resources. FreeRTOS offers a low overhead but high performance task management system which does multitasking, deterministic scheduling and preemptive context switching which is mandatory in the application of working concurrently in cryptographic functions like key negotiation, packet-level encryption, signature verification, and my Lewis boot verification. With regards to quantum based embedded communication, FreeRTOS enables real-time timing guarantees and high priority to critical security activities, e.g., Kyber512 key exchanges or Dilithium2 signature verifications, without starving other real-time activities, e.g., sensor data ingest or control loop operations. It offers secure and synchronized inter-process communication based on Message queues, mutexes and semaphores, and software timers built-in, making it the main advance necessary to coordinate between cryptographical modules, and network I/O tasks. Unlike other RTOSs, FreeRTOS deterministic

operation guarantees the cryptography algorithms and sensitive time tasks are responsive and predictable even in fluctuating workload or interrupt-heavy situations, or the essential need of industrial machines, healthcare gadgets, and mission-critical connected gadgets. Moreover, FreeRTOS can run on many different architectures such as ARM Cortex-M and RISC-V and also manage any built-in integer cryptography requirements on those architectures, the post-quantum cryptographic stack can be easily ported to heterogeneous embedded systems. The FreeRTOS modular kernel architecture also allows the addition of other security measures to the system, like task isolation, over-stack checking, and firmware update protection to increase the overall reliability of the system. With the ability to allow deterministic real-time in Free Wednesday community, the presented quantum-resilient framework not only attains cryptographic integrity, but also stability and precision in time performance, and responsiveness needed in the future secure embedded communication systems.

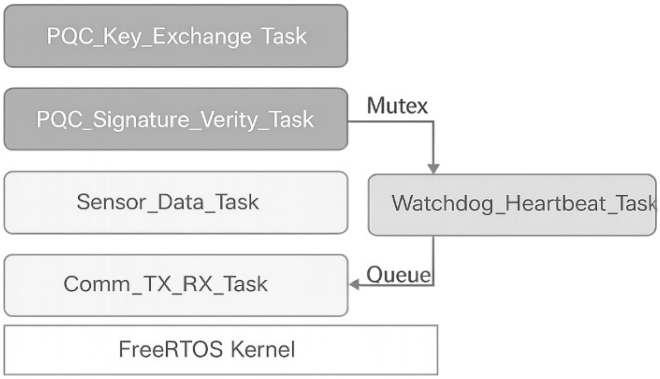

Fig. 2: Task Scheduling and Inter-Process Communication in FreeRTOS for PQC-Enabled Embedded Systems

Communication Bus

All industry-standard serial interfaces used to select the construction of the proposed quantum-resilient embedded system the use of UART (Universal Asynchronous Receiver/Transmitter) and SPI (Serial Peripheral Interface) communications which are due to the high rate of use, energy consumption, and low-latency, embedded applications. Such set of protocols makes it safe and concise to exchange data between the microcontroller and the periphery components like sensors, actuators, external memory, and other embedded nodes. With this architecture, UART is mostly designated to be used with long-range asynchronous communication and minimal wiring and complexity must be prohibited whereas SPI is employed to perform high-speed, synchronous, and short distance data communication with high throughput performance, like a connection to secure elements or radio transceivers. In order to achieve strong end-to-end protection over these essentially insecure mechanisms, encryption is performed at the packet level making use of symmetric keys which are generated at initial handshaking and are formed based upon Kyber512. This also guarantees data being transmitted is kept secure and will not be eavesdropped or fall victim of a man-in-the middle attack even on an unmanned environment like the open industrial networks or wireless sensor implementation. Besides encryption, Dilithium2-based digital signatures that satisfy the authentication requirement, are computed on every packet or message frame, and used as the authentication tags that can guarantee the identity of the sender and guarantee the integrity of the data, which is effective to countermeasures the spoofing, tampering, and replay attacks. Coupling Kyber and Dilithium provides confidentiality and authenticity, as well as forward secrecy, which is essential in ensuring secure key evolution in terms of IoT networks. Moreover, the communication layer is very coupled with FreeRTOS tasks, which makes it an efficient deal in terms of managing interrupt operations and buffer operations within real-time requirements. This is a layered security design, applied on UART and SPI which gives a scalable and practical, both clandestinely and quantum-resistant mechanism of communication to be used in the fields of embedded applications such as health, smart grid, and automotive systems, where the data integrity and the responsiveness of the systems is critical.

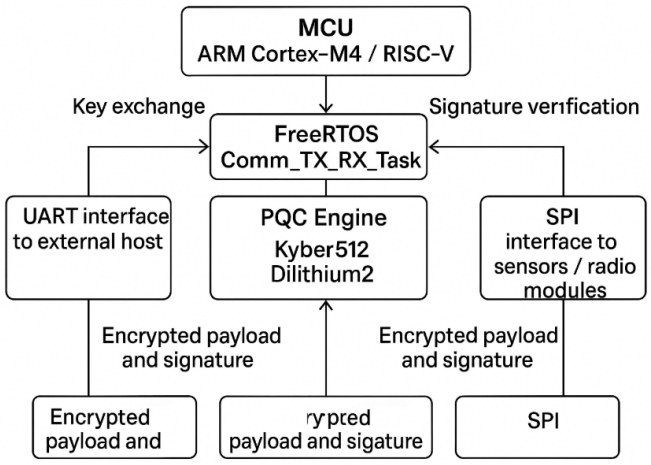

Fig. 3: Secure UART/SPI Communication Architecture in PQC-Enabled Embedded Systems

METHODOLOGY

Algorithm Selection

The cryptography algorithms to be incorporated in the embedded system should be a trade-off among quantum resistance, efficiency of computation, memory

requirements and the relative ease of implementation. This paper has listed three of the most established post-quantum cryptography (PQC) algorithms Kyber512, Dilithium2 and SPHINCS+-128s and carefully selected them on the basis of their performance characteristics, security assumptions, and their use in the standardization process of the NIST PQC standardization process. Each algorithm plays a certain role in secure communication stack and is designed to work in embedded contexts.

Kyber512 is selected as the Key Encapsulation Mechanism (KEM) when it comes to creating secure symmetric session keys to connect communicating nodes. It is grounded on the Module Learning with Errors (MLWE) issue which is supposed to be impervious both to traditional and quantum attacks. The ciphertexts and keys are compact, and both key generation and encapsulation/decapsulation are fast, and the scheme can be simply implemented without the use of floating-point arithmetic, making it very apt to microcontrollers with limited resources, which is why it is called Kyber512. It is very fast in establishing session keys, and this is paramount to the low latency of communications within real-time embedded applications.

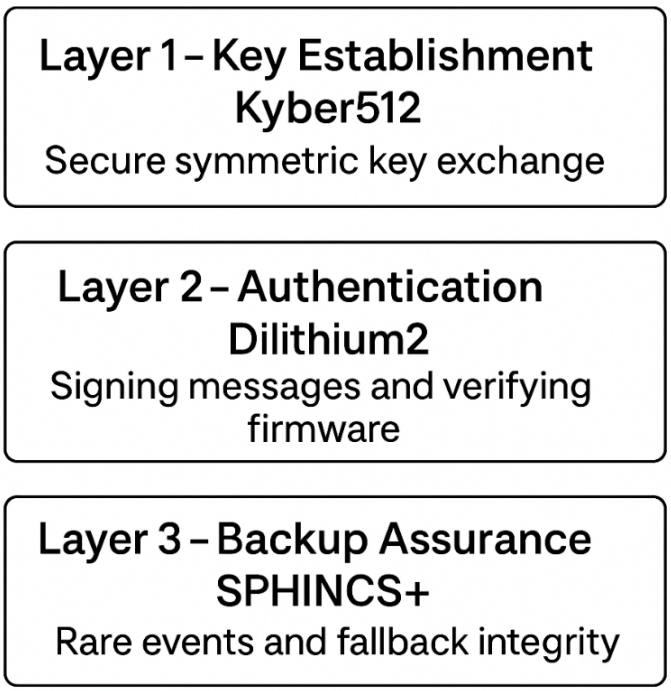

Fig. 4: Role-Based Allocation of PQC Algorithms in the Embedded Cryptographic Stack

The digital signature algorithm used to certify messages, sign firmware and include an examination of integrity is Dilithium2. Dilithium2, also built on lattice cryptography, especially Module Learning with Rounding (MLWR) problem, has the reputation of being well balanced in terms of speed, security, and relatively small signature size. It can implement constant-time functions and has fairly low both computational and memory overhead, which means that it can be easily used in embedded applications where frequent signing and checking operations have to be performed. Dilithium2 can be used to produce authentication tags on packets that are going to be encrypted and on software when performing secure boot or firmware upgrades to verify authenticity of the software being executed.

An unconditional long-term cryptographic guarantees are achieved by incorporating SPHINCS+-128s as backup signature scheme stateless and hash-based. As opposed to Kyber and Dilithium, SPHINCS+ is not based on algebraic assumptions of hardness, and is entirely founded on the notion of secure hash functions. SPHINCS+ is capable of providing an extra level of security where the utmost level of trust is required like when signing archival data, or checking fallback verification in key-compromise situations, both at the cost of larger signature sizes and computational latency than in SPHINCS. It also makes

Table 2. Functional Comparison of Selected Post-Quantum Cryptographic Algorithms for Embedded Systems

Algorithm	Type	Primary Purpose	Underlying Problem	Key/Signature Size	Performance	Embedded Suitability
Kyber512	Key Encapsulation Mechanism	Session key establishment	Module Learning with Errors (MLWE)	Small ciphertext & key	Fast (Enc/Dec < 2 ms)	Ideal for low-latency, real-time key exchange
Dilithium2	Digital Signature Scheme	Authentication & integrity	Module Learning with Rounding (MLWR)	Moderate (~2–2.7 KB)	Moderate (~3.5 ms)	Suitable for frequent signing and secure firmware
SPHINCS+-128s	Hash-Based Signature Scheme	Fallback signature / archival use	Stateless Hash-based	Large (>8 KB)	Slow (~9.7 ms)	Best for high-assurance or post-compromise recovery

management of keys easy and assures protection against quantum-enabled attacks and structural cryptanalysis due to its stateless property.

The combination of these three algorithms is highly flexible and secure PQC stack that can adjust to the different needs of embedded applications and provide a strong cryptographic defence on a variety of operational environments.

Optimization Strategies

Various software and system optimization techniques were adopted so that the post-quantum cryptographic (PQC) algorithms can be executed efficiently under the tight requirements of embedded platforms. All these optimizations are necessary to achieve real-time capabilities, minimized memory overhead and maximized resistance to side-channel attacks without sacrificing cryptographic correctness and robustness. The techniques that led to the optimization of the Kyber512, Dilithium2 and SPHINCS+-128s implantation to the ARM Cortex-M4 and RISC-V E31 microcontrollers include the following:

Loop unrolling at assembly level:

Loop unrolling is one method of low-level code optimization, in which the overhead of the branching and loop control logic in repeated execution is minimized by copying out (or unfolding) the loop. Generally, in the case of PQC algorithms, which are dominated by polynomial arithmetic and matrix computations, loop unrolling has been used in routines that are performance-sensitive (especially with respect to modular multiplication and vector processing). Since loops may be rewritten at the assembly level, the processor is able to execute instructions faster, lessen pipeline stalls, and better control then, n matters of instruction timing. This translates into faster in the key generation, encapsulations and signature calculations especially at a device that does not support out of order execution or one that has less instruction cache.

Static memory allocation:

Implementation Dynamic memory allocation, often via malloc or calloc, can cause fragmentation, greater runtime unpredictability and the invocation of security vulnerabilities associated with heap overflows or memory leakage. Static memory allocation was used to implement the entire PQC stack to remove the risk and enhance more memory predictability. The sizes of the public key, signature, ciphertext and temporary space in which to perform necessary computations were configured statically and were aligned to memory boundary to prevent conflicting with the cache lines. This will not only increase the reliability of the execution and predictability but also easier to be integrated into real-time operating systems such as FreeRTOS where deterministic memory usage is important.

Constant-Time Programming:

Constant-time coding practices were used in implementing all cryptographic primitives to alleviate timing side-channel attacks, which are especially powerful in the embedded systems because of physical accessibility reasons. These are not making branches or memory accesses conditioned on secret data, avoiding loops with variable-time executed on key bits, and avoiding table-based arithmetic and masking. Methods of levelling the execution paths using techniques like conditional move (CMOV) instructions and dummy operations were applied. Constant-time programming it is a programming technique to ensure that the timing properties of cryptography do not leak any sensitive information, which greatly increases resistance to timing, power analysis and micro architectural cache-based attacks.

All these optimization approaches lead to a secure, real-time and resource-efficient implementation of quantum-

Table 3. Optimization Strategies for Efficient Post-Quantum Cryptographic Implementation in Embedded Systems

Optimization Strategy	Purpose	Implementation Details	Impact on PQC Execution
Assembly-Level Loop Unrolling	Reduce loop overhead and improve execution speed	Manual unrolling of loops in polynomial and matrix operations to minimize branching delays	Enhances throughput in key generation, encapsulation, and signing
Static Memory Allocation	Ensure predictable and secure memory usage	Pre-allocated buffers for keys, ciphertexts, and signatures; avoids heap fragmentation	Improves reliability, avoids memory leaks, and supports real-time OS
Constant-Time Programming	Prevent timing and power-based side-channel attacks	Use of conditional moves (CMOV), branchless logic, and fixed-time execution paths	Increases resilience to timing, power analysis, and cache attacks

resistant cryptographic protocols that can be used in the next-generation network.

Implementation Flow

The implementation process of the quantum-resilient proposed embedded communication framework is expected to lay out an end-to-end security configuration, where the workflow and implementation focus on security implications of the system start-up to the runtime communication and firmware updates. This flow will also secure every stage of your device against classical and quantum threats that could be used by means of NIST-approved post-quantum cryptographic (PQC) algorithms. The integrated security operations can be discussed in the following sequential steps:

1. PQC firmware is loaded by secure bootloader:

The system starts at the secure boot, where a trusted bootloader planted in the read-only-memory performs the verification of the authenticity of the application firmware to be run. This check is executed based on Dilithium2 digital signatures and only a firmware signed by a trusted authority can be loaded. The verification key is brought into non-tamperable memory and when there is a mismatch during signature verification the verification is halted with a stop boot command. This step ensures the first root-of-trust, deterrence of unauthorized or malicious injection at the startup of the firmware.

2. Kyber512 Negotiated Session Key:

After the authenticated firmware is running, the system can enter a post-quantum secure session key exchange using Kyber512 key encapsulation mechanism (KEM). The process is used to generate a symmetric key between the device and a peer in communication (e.g. gateway, cloud server, or another embedded node) across a possibly insecure channel. Kyber512 both guards against passive and active quantum-enabled attackers on the derivation of the session key. The developed key will be utilized later to encrypt data packets, and it provides confidentiality and forward secrecy.

3. Data Authentication with Firmware Update through Dilithium2:

Digital signature keys generated through the Dilithium2 algorithms are then used to verify any firmware update, or critical control message during runtime. It makes sure that the over-the-airs are coming only through trusted and have not been changed through the air. Further, periodic messages i.e. sensor reading, system log, etc or control message, will be signed to ensure integrity and non-repudiation. The combination of a small signature size and efficient signature verification - allows Dilithium to be integrated in the real-time embedded setting without affecting the performance of such application.

4. Fallback Authentication in Low-Trust: SPHINCS +:

In very sensitive or mistrusting environments, e.0g. a system that has to run in a hostile physical environment, a critical infrastructure, or in a regulatory requirement context, a fallback SPHINCS+-128s based authentication mechanism is used. PHINCS+ uses stateless hash-based cryptography, a foundation that is long term and survivable against future mathematical advances unlike lattice-based schemes. It suffers a larger cost of computation and signatures size but is applied under verification, e.g. when signing securely stored data, when providing root keys, or when the main (non-compromised) signing algorithm is unavailable.

The low level of the implementation flow shown above is multi-stage, and each stage of operation (including the boot, runtime communication, and firmware lifecycle) is secured using cryptographically hard primitives against quantum algorithms.

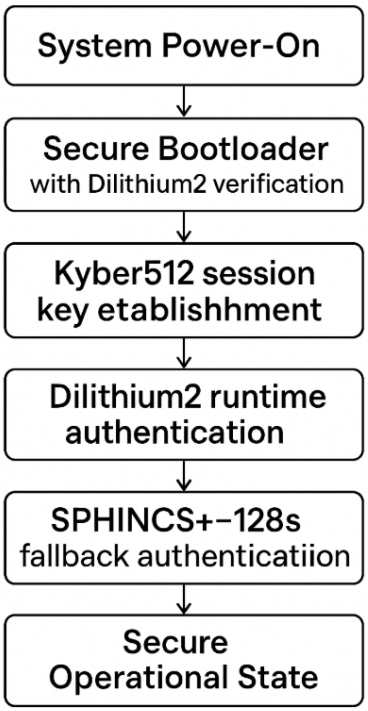

Fig. 5: Implementation Flow of Post-Quantum Cryptographic Integration in Embedded Systems

Experimental Setup

In order to ascertain the usefulness, performance, and efficiency of the quantum-resilient cryptographic implementation suggested, an elaborate experimental

platform was established with an actual embedded platform and an industry standard development kit. The official platform to have your results checked was the STM32F407VG microcontroller (with its ARM Cortex-M4 32-bit core clocked at 168 MHz), a popular General Purpose 32-bit MCU in IoT, cars, and industrial systems. The device has 1 MB of Flash and 192 KB of SRAM that can support a relatively limited memory set up adequate to benchmarking on the composition of post-quantum cryptographic primitives. The firmware stack was crafted and executed on FreeRTOS v10.4, a light weight complex operating system supporting definite task planning and appropriate to timing critical cryptographic transaction. Our codebase was built with GCC ARM Embedded 9.3 toolchain that has size and speed optimization options essential to embedded programs. INA219, a sensing chip that measures high-side current/power, was used with I2C bus to count on the real-time voltage of the electromagnetic controller and real time current consumption as the cryptography operations run through the MCU. This enabled the overhead of the energy that was brought by PQC algorithms to be quantified. OpenOCD and STM32CubeIDE were utilized to debug and flash the program and observe memory utilization, task and running progress, and measurements of the state of the peripherals directly. pySerial was used to log the serial traffic and ensure that packets with encrypted data were being transmitted successfully and correctly received. Not only does such an experimental setup resemble a realistic embedded deployment scenario, but also allows fine-grained profiling of post-quantum algorithm performance concerning execution time, memory utilization, power consumption and cryptographic soundness, thus demonstrating the reality of the integration of PQC into resource-limited embedded communication systems.

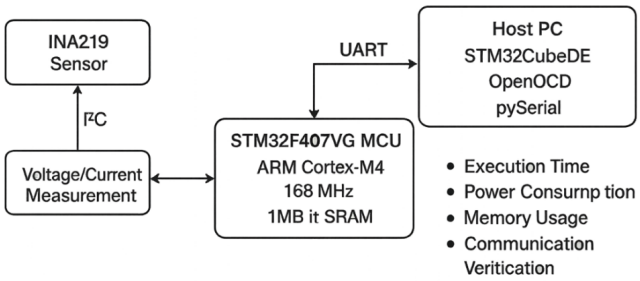

Fig. 6: Hardware-Software Experimental Setup for Evaluating PQC in Embedded Systems

Results and Discussion

Post-Quantum cryptographic algorithm implementation onto the STM32F407VG embedded system board was tested regarding memory consumption, computational time complexity, and real-time capabilities. The memory requirement of Kyber512 was 28 KB of Flash and 6.4 KB of SRAM as evident in the tabulated results, thus covering a small area when compared to a memory constraint device. The time used up by the key generation process was 1.2 milliseconds and the encryption and decryption processes took an average of 1.9 milliseconds and 2.0 milliseconds respectively. Such low latencies of less than 2 ms ensure that Kyber512 is very useful in machine-to-machine (M2M) and low-latency communications, including industrial automation or sensor-actuator networks, where it is critical to rely on frequent and fast key establishment. The fact that it is computationally efficient also implies that the power consumed by the algorithms during the execution process is minimal as confirmed by measuring current, implying that the secure session key exchange is not expected to have great effects on the energy cost of the device that is executing the algorithms.

Its digital signature scheme of authentication and integrity verification Dilithium2 consumed 36 KB Flash, 8.1 KB SRAM, slightly above Kyber and within the range of most embedded MCUs. The average time of signature generation and signature verification was 3.5 milliseconds, which was fast enough to be operated in area of firmware signing, signature verification of control command and periodic telemetry authentication. Combining signature size, speed and levels of cryptographic strength, Dilithium2 can be best applied to real-time firmware update and message signing within embedded contexts. Additionally, its lattice-based representation makes it suitable to implement with integer arithmetic, making it easy to integrate on processors that do not have floating point. It was noted that, the algorithm has a stable execution profile even at a high load and therefore can be utilized in FreeRTOS-based multi-tasking real-time phased systems.

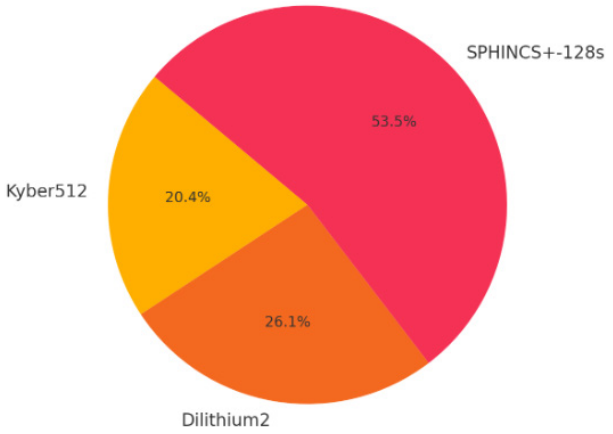

Fig. 7: Memory Usage Distribution of PQC Algorithms on STM32F407VG Platform

Table 4. Performance Evaluation of Post-Quantum Cryptographic Algorithms on STM32F407VG Embedded Platform

Algorithm	SRAM (KB)	Key Generation Time (ms)	Encapsulation/Decapsulation Time (ms)	Signature Generation Time (ms)	Use Case Suitability
Kyber512	6.4	1.2	1.9 / 2.0	N/A	Fast session key exchange in M2M & low-latency systems
Dilithium2	8.1	N/A	N/A	3.5	Real-time authentication & firmware signing
SPHINCS+-128s	12.3	N/A	N/A	9.7	Archival signatures, fallback for high-trust systems

By contrast, SPHINCS+-128s, the stateless hash-based signature scheme, needed a much larger footprint 78 KB of Flash, 12.3 KB of SRAM and 9.7 milliseconds to create a signature. Despite a lower performance metrics offered by the algorithm than Kyber and Dilithium, SPHINCS+ provides unparalleled security in the long-term perspective, as it does not employ any algebra structures, which might be hacked sometime in the future. This means SPHINCS+ can be used as a fallback/archival signature scheme, which has limited utility; it is very useful where the top level of assurance is required, like with medical records, secure logging or safe control systems. Although it is not the most appropriate choice when the task is a common signing activity, because it imposes latency, its availability offers redundancy and future-proofing to systems where security survivability and cryptographic agile operations are prioritized. On the whole, the findings substantiate that Kyber512 and Dilithium2 can be effectively and efficiently deployed in the resource-constrained embedded systems, with SPHINCS+ providing a strategic level of security in extraordinary applications and the feasibility of a multi-level PQC stack in the next-generation embedded communication systems.

Conclusion

This study has managed to show how feasible this integration of post-quantum cryptographic (PQC) algorithms into any resource-constrained embedded communication systems will be, which is in response to the current threats that quantum computing poses to the classical cryptographic relic. By applying and testing NIST-recommended schemes, i.e. Kyber512 to perform key encapsulation, Dilithium2 to perform digital signature, and SPHINCS+-128s to provide fallback-assurance, the research proves that quantum-resilient cryptographical security is possible without causing unacceptably high computational burden, memory requirement, and energy consumption. Kyber512 was very good as the real time key exchange with sub-2 ms latencies verification, Dilithium2 was very fast and efficient in signature verification and was suited towards slow authentication like firmware and secure messaging. As much as SPHINCS + added overhead, its integration into the system ensures strong and long-term security and as such, it functions perfectly in solving security problems with heightened assurance requirements. This collection of algorithms in an open implementation stack allows the dynamic adaptation to different security environments and resource resources to be certain about future support of the embedded systems in any area, like the IoT, industrial automation and critical infrastructures. Besides, optimization on the system level such as constant-time execution, static memory, and real-time task scheduling on FreeRTOS have further contributed to the appropriateness of PQC in embedded systems. In the future, we will also concentrate on easier transitional deployment of hybrid classical-PQC stacks and ensure the deployment of hardware-assisted acceleration of lattice operations as well as formal proofs of resistance to side-channel attacks and fault injection attacks. It provides a framework on constructing a quantum-resilient embedded communication system on which the post-quantum era can be secured, scalable, and compatible to the existing standards.

References

1. Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2016). Post-quantum key exchange—A New Hope. *Proceedings of the 25th USENIX Security Symposium*, 327–343.
2. Hülsing, A., Rijneveld, J., Beullens, J., Kölbl, S., Noll, T., & Song, F. (2019). SPHINCS+: Submission to the NIST Post-Quantum Project. NIST Post-Quantum Cryptography Project.
3. Oder, T., Schneider, T., Pöppelmann, T., & Schwabe, P. (2019). Practical post-quantum cryptography for embedded devices: Towards multiple signatures in embedded systems. ACM Transactions on Embedded Computing Systems (TECS), 18(3), 1–27. https://doi.org/10.1145/3289186
4. Bos, J. W., Costello, C., Naehrig, M., & Stebila, D. (2015). Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. 2015 IEEE Symposium on

Security and Privacy, 553–570. https://doi.org/10.1109/SP.2015.40

5. Bernstein, D. J., Lange, T., & Niederhagen, R. (2017). Dual EC: A standardized back door. In The New Codebreakers (pp. 256–281). Springer. https://doi.org/10.1007/978-3-319-25067-0_10
6. Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). Report on post-quantum cryptography. NISTIR 8105. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8105
7. Schwabe, P., & Stoffelen, K. (2020). All the AES you need on Cortex-M3 and M4. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020(3), 1–28. https://doi.org/10.46586/tches.v2020.i3.1-28
8. Kretzschmar, F., & Müller-Quade, J. (2021). Analyzing the efficiency of post-quantum key encapsulation mechanisms in TLS 1.3. Designs, Codes and Cryptography, 89(8), 1707–1726. https://doi.org/10.1007/s10623-020-00748-y
9. Hülsing, A., Rijneveld, J., & Schwabe, P. (2018). ARMed SPHINCS: Computing a 41 KB signature on Cortex-M4. International Conference on Cryptology in Africa, 201–222. https://doi.org/10.1007/978-3-319-89339-6_11
10. Aggarwal, D., Brennen, G. K., Lee, T., Santha, M., & Tomamichel, M. (2017). Quantum attacks on Bitcoin, and how to protect against them. Ledger, 3, 68–90. https://doi.org/10.5195/ledger.2018.127
11. Prasath, C. A. (2025). Adaptive filtering techniques for real-time audio signal enhancement in noisy environments. National Journal of Signal and Image Processing, 1(1), 26–33.
12. Rahim, R. (2025). Lightweight speaker identification framework using deep embeddings for real-time voice biometrics. National Journal of Speech and Audio Processing, 1(1), 15–21.
13. Uvarajan, K. P. (2025). Design of a hybrid renewable energy system for rural electrification using power electronics. National Journal of Electrical Electronics and Automation Technologies, 1(1), 24–32.
14. Rahim, R. (2025). Mathematical model-based optimization of thermal performance in heat exchangers using PDE-constrained methods. Journal of Applied Mathematical Models in Engineering, 1(1), 17–25.
15. Mäkinen, R. (2024). The Role of Digital Twins in Improving Business Processes and Quality Management. National Journal of Quality, Innovation, and Business Excellence, 1(2), 23-29.