|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reference (Author, Year) | Name of the Paper | Main Contribution/Focus | Identified Problem/Vulnerability | Proposed Solution/Approach | Relevance to the Research Topic |
| 1. Rasheed et al. (2024) | Artificial intelligence and machine learning-enabled security for industrial internet of things in the context of industry 5.0 | Explores the security challenges of the Industrial Internet of Things (IIoT) in Industry 5.0, emphasizing the role of AI and ML. | Current security protocols are vulnerable to sophisticated cyber-attacks, including those potentially enabled by quantum computers in the future. | Proposes a multi-layered, AI-driven security framework for threat detection and response in IIoT networks. | Establishes the security-critical nature of Industry 5.0 environments, which is the foundational context for human-robot workspaces. |
| 2. PQC for IoT (Book, 2021) | Post-Quantum Cryptography for IoT | Provides a comprehensive overview of Post-Quantum Cryptography (PQC) tailored for resource-constrained IoT devices. | Traditional public-key cryptography (RSA, ECC) will be broken by quantum computers. IoT devices lack the resources for complex PQC algorithms. | Analyzes various PQC candidates (lattice-based, code-based, etc.) for their suitability in IoT, focusing on efficiency and memory footprint. | Directly addresses the need for lightweight PQC, a core requirement for securing real-time communication in robotic systems. |
| 3. Mahbub (2020) | A Brain-Computer Interface-Based Human-Robot Interaction for Smart Factory | Discusses brain-computer interfaces (BCI) for human-robot interaction in smart factories. | Security and privacy of sensitive neural data transmitted between humans and robots are paramount and often overlooked. | Suggests implementing robust encryption and authentication mechanisms for BCI data streams. | Highlights the critical need for secure communication in advanced HRC, which will be susceptible to quantum threats. |
| 4. Bécue et al. (2020) | A flexible architecture for safe human-robot collaboration in an industrial environment | Proposes an architecture for a safe and collaborative human-robot workspace. | Focuses on physical safety but acknowledges that cybersecurity breaches can lead to unsafe robot actions, compromising human safety. | An IoT-based architecture using sensors to monitor the workspace and ensure the robot operates safely around humans. | Underscores the link between cybersecurity and physical safety in HRC, reinforcing the need for unbreakable, real-time communication. |
| 5. Bellini et al. (2022) | A Survey on 5G-IoT and an AI-Based Application for Smart Industries | Investigates the integration of 5G, IoT, and AI for smart industrial applications. | High-speed, low-latency communication in 5G-enabled factories creates new attack surfaces. | A framework for leveraging 5G's capabilities while integrating security-by-design principles at the network level. | Relevant for ensuring the real-time aspect of communication in HRC, a key requirement that quantum-resistant security must not compromise. |
| 6. Rathore & Park (2021) | A lightweight authentication scheme for industrial internet of things | Presents a lightweight authentication scheme for the Industrial Internet of Things. | Existing authentication protocols are often too computationally expensive for IIoT devices and are not quantum-resistant. | A lightweight, elliptic-curve-based mutual authentication scheme. | While not post-quantum, it directly addresses the lightweight requirement, showing the type of efficiency needed for PQC in robotics. |
| 7. Lv et al. (2022) | Security and Privacy in Industry 4.0 | Explores security and privacy challenges in the context of Industry 4.0 and the transition to 5.0. | Data integrity and confidentiality in smart manufacturing systems are threatened by increasingly sophisticated attacks. | Recommends a zero-trust security model for industrial networks, where no device is trusted by default. | This zero-trust concept must be extended with a PQC layer to be future-proof, fitting the research topic's problem statement. |
| 8. Chelladurai et al. (2021) | Human-Centricity in Industry 5.0 | Focuses on the human-centric principles of Industry 5.0, including human-robot collaboration. | Emphasizes that trust is crucial for effective HRC. This trust can be broken by cyber-attacks that manipulate robot behavior. | Advocates for resilient and human-aware systems that prioritize worker well-being and safety. | Provides the "why" for the research topic: securing HRC is not just a technical problem but a necessity for the success of the Industry 5.0 paradigm. |
| 9. Yan et al. (2022) | A Survey of Lattice-Based Cryptography for IoT Security | Discusses lattice-based cryptography for secure communications in constrained environments. | Need for quantum-resistant algorithms that can perform efficiently on devices with limited processing power and memory. | Analyzes the performance of lattice-based PQC schemes (like Kyber) on microcontrollers. | Directly supports the feasibility of a lightweight PQC layer by showing that lattice-based methods are viable for embedded systems like those in robots. |
| 10. Breitenbacher et al. (2019) | A survey on security for collaborative robots | A survey on the security of collaborative robots (cobots). | Cobots are vulnerable to network-based attacks (e.g., man-in-the-middle) that can alter commands and cause harm. | A comprehensive threat model for cobots and a review of existing security measures. | Explicitly details the vulnerabilities in HRC workspaces that a unified PQC security layer would aim to solve. |

**A Literature Review on the Design of a Post-Quantum Cryptographic Layer for Secure Human-Robot Collaboration in Industry 5.0**

**Abstract**

The transition towards Industry 5.0 marks a paradigm shift from pure automation to a synergistic model of human-robot collaboration (HRC), emphasizing human-centricity, resilience, and sustainability. The security of these collaborative workspaces is paramount, as a compromise can lead to significant safety risks, intellectual property theft, and operational disruption. However, the foundational cryptographic primitives, such as RSA and Elliptic Curve Cryptography (ECC), that currently secure industrial systems are demonstrably vulnerable to future attacks by large-scale quantum computers. This impending threat necessitates a proactive migration to post-quantum cryptography (PQC). This report presents an exhaustive literature review synthesizing contemporary research to propose a conceptual, multi-layered PQC security architecture tailored for the unique demands of HRC environments. A critical analysis of leading PQC algorithms, including the NIST-standardized CRYSTALS-Kyber, CRYSTALS-Dilithium, and SPHINCS+, alongside alternatives like Falcon, reveals a complex landscape of trade-offs between security assurance, computational performance, and resource consumption. The investigation delves into the profound architectural challenges PQC introduces for resource-constrained embedded systems, such as limited memory in Hardware Security Modules (HSMs) and protocol-level overhead. Furthermore, it examines the practical threats of side-channel attacks against PQC implementations and the performance cost of necessary countermeasures. The synthesized architecture addresses these challenges by adopting a heterogeneous approach, strategically deploying different quantum-resistant algorithms across various system layers: Falcon for secure boot, a Kyber-Dilithium combination for secure communication channels, extensions for securing middleware like ROS 2, lightweight signatures for edge devices, and an exploration of Fully Homomorphic Encryption (FHE) for privacy-preserving analytics. This review culminates in a set of actionable recommendations and identifies critical directions for future research, providing a foundational blueprint for developing robust, performant, and future-proof security for the next generation of industrial systems.

**Introduction**

The trajectory of industrial development has entered a new, transformative phase known as Industry 5.0. This evolution represents a significant departure from the automation-centric focus of Industry 4.0, reintroducing a human-centric element to the core of manufacturing and production processes. The vision of Industry 5.0 is not to replace human workers with machines but to foster a deeply integrated, collaborative environment where human ingenuity and critical thinking are augmented by the precision, strength, and data-processing capabilities of robots and cyber-physical systems. This paradigm is built on three pillars: human-centricity, which places worker well-being and involvement at the forefront; sustainability, which demands environmentally conscious and resource-efficient processes; and resilience, which calls for adaptable and robust production systems capable of withstanding disruptions like supply chain shocks or pandemics. Achieving this vision depends critically on a foundation of secure, trustworthy, and seamless interaction between humans and their robotic counterparts.

**The Role of Human-Robot Collaboration (HRC) and its Inherent Security Challenges**

At the heart of Industry 5.0 lies the concept of Human-Robot Collaboration (HRC). Unlike traditional industrial automation where robots are confined to caged-off areas, HRC involves shared workspaces where humans and collaborative robots (cobots) operate in close proximity, often performing tasks on the same workpiece simultaneously. This intimate level of interaction creates unprecedented opportunities for efficiency and flexibility but also introduces a new class of safety and security challenges. The communication channels that govern a robot's behavior—its motion planning, sensor interpretation, and physical actions—must be protected with the highest level of integrity and authenticity. A compromised robot in an HRC setting is not merely a data breach; it is a direct and immediate physical threat to the human worker beside it. The security requirements are therefore stringent and time-critical. Real-time guarantees are needed to ensure that control commands are executed as intended, that safety overrides are instantaneous and cannot be spoofed, and that sensor data reflecting the state of the shared environment is accurate and untampered. Any security architecture for HRC must prioritize these real-time integrity and authenticity guarantees above all else.

**The Impending Quantum Threat to Cyber-Physical Systems**

The security of virtually all modern digital infrastructure, from internet communications to industrial control systems, is built upon public-key cryptography (PKC). Algorithms such as RSA and Elliptic Curve Cryptography (ECC) are used to establish secure communication channels, verify software integrity, and authenticate identities. Their security relies on the computational difficulty of certain mathematical problems, namely integer factorization and the discrete logarithm problem, which are intractable for even the most powerful classical supercomputers. However, the advent of large-scale, fault-tolerant quantum computers will render these foundational algorithms obsolete. In 1994, Peter Shor developed a quantum algorithm that can solve these underlying mathematical problems in polynomial time, meaning a sufficiently powerful quantum computer could break RSA and ECC encryption with ease.

This looming "Quantum Apocalypse" poses an existential threat to the long-term security of cyber-physical systems. Industrial equipment, including robots, often has a lifespan measured in decades. A system deployed today with classical cryptography could be vulnerable long before it is decommissioned. This risk is amplified by the "Harvest Now, Decrypt Later" (HNDL) attack strategy, where adversaries can capture and store encrypted data today with the intention of decrypting it once a quantum computer becomes available. For industrial environments, this means sensitive data such as proprietary manufacturing processes, intellectual property, and operational telemetry are already at risk of future exposure. The urgency to transition to quantum-resistant cryptographic solutions, collectively known as Post-Quantum Cryptography (PQC), is therefore not a distant concern but an immediate strategic imperative for ensuring the future security and safety of industrial infrastructure.

**Objective and Structure**

The objective of this report is to conduct an exhaustive review of the existing literature to synthesize a robust, performant, and practical design for a post-quantum cryptographic security layer specifically tailored for HRC workspaces in Industry 5.0. This involves moving beyond a high-level survey of PQC algorithms to a deep, analytical exploration of their suitability for the unique constraints and requirements of industrial embedded systems. The report is structured to guide the reader through a logical progression, beginning with a foundational analysis of the PQC landscape and the leading quantum-resistant algorithms. It then examines the specific architectural and security imperatives of HRC systems and the profound system-level challenges that arise when integrating PQC. Following this, a detailed investigation into the performance, implementation overhead, and side-channel vulnerabilities of PQC on resource-constrained hardware is presented. Finally, these disparate threads of analysis are woven together into a synthesized, multi-layered architectural proposal, providing a concrete blueprint for securing HRC systems. The report concludes with actionable recommendations for system architects and identifies critical gaps in the current body of research that warrant future investigation.

**1. The Post-Quantum Cryptographic Landscape for Industrial Systems**

The transition to a quantum-resistant security posture begins with a thorough understanding of the available cryptographic tools. The field of Post-Quantum Cryptography (PQC) is not a monolithic entity but a diverse collection of cryptographic families, each based on different mathematical problems believed to be hard for both classical and quantum computers. The selection of an appropriate PQC algorithm is a complex engineering decision, involving a delicate balance of security assumptions, performance characteristics, and resource requirements. This section provides a foundational analysis of the PQC landscape, focusing on the most promising candidates for deployment in demanding industrial environments.

**1.1. An Overview of Quantum-Resistant Algorithm Families**

The international cryptographic community, led by standardization bodies like the U.S. National Institute of Standards and Technology (NIST), has been evaluating several families of PQC algorithms. These families are distinguished by the underlying mathematical problem that provides their security. The primary candidates include:

* **Lattice-based Cryptography:** This family relies on the presumed hardness of problems on mathematical structures called lattices, such as the Shortest Vector Problem (SVP) and the Learning With Errors (LWE) problem. Lattice-based schemes have emerged as the front-runners in the NIST standardization process due to their strong security foundations and, crucially, their excellent all-around performance, offering relatively small key and signature sizes combined with fast computational speeds.
* **Hash-based Cryptography:** The security of these schemes is derived directly from the properties of cryptographic hash functions (e.g., collision resistance, preimage resistance). Hash-based signatures are highly regarded for their conservative security assumptions; their security can be reduced directly to that of the underlying hash function, a well-understood primitive. However, this strong security often comes at the cost of larger signature sizes and slower performance, particularly in signing operations.
* **Code-based Cryptography:** This is one of the oldest families of PQC, with the McEliece cryptosystem being proposed in 1978. Its security is based on the difficulty of decoding a general linear error-correcting code. While known for its strong security and fast encryption/decryption, code-based schemes have historically been hampered by very large public key sizes, which has limited their practical adoption.
* **Multivariate Cryptography:** These schemes are based on the difficulty of solving systems of multivariate polynomial equations over a finite field. They can produce very short signatures but have had a mixed history of security, with several prominent proposals being broken over the years.
* **Isogeny-based Cryptography:** This approach uses maps (isogenies) between elliptic curves. It was once a promising candidate due to its remarkably small key sizes. However, a significant cryptanalytic breakthrough against a leading isogeny-based scheme, SIKE, has cast doubt on the maturity and security of this family, leading to its removal from the later stages of the NIST competition.

Among these families, lattice-based cryptography has demonstrated the most versatile and balanced profile, making it the primary focus for general-purpose key establishment and digital signature applications in the forthcoming PQC standards.

**1.2. Analysis of NIST-Standardized Algorithms**

After a multi-year, multi-round public competition, NIST announced its first set of PQC algorithms for standardization in 2022, with draft standards released in 2024. These selections are poised to become the global benchmark for quantum-resistant security and are thus the most critical algorithms to analyze for industrial deployment.

**CRYSTALS-Kyber (ML-KEM)**

The selected algorithm for public-key encryption and key establishment is CRYSTALS-Kyber, officially standardized as FIPS 203, the Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM). Kyber is a Key Encapsulation Mechanism (KEM) whose security is based on the hardness of the Module Learning With Errors (MLWE) problem over polynomial rings. As a KEM, its primary function is not to encrypt arbitrary data directly but to establish a shared secret between two parties. In a typical protocol, one party (the encapsulator) uses the recipient's public key to generate a shared secret and a ciphertext. The recipient (the decapsulator) then uses their private key to derive the same shared secret from the ciphertext. This shared secret is then used to key a highly efficient symmetric encryption algorithm (like AES) for the actual communication. Kyber has been shown to be suitably fast and efficient, with key generation, encapsulation, and decapsulation operations typically completing in a few milliseconds on modern microcontrollers, making it a strong candidate for establishing secure communication channels in real-time systems.

**CRYSTALS-Dilithium (ML-DSA)**

For digital signatures, NIST selected CRYSTALS-Dilithium, standardized as FIPS 204, the Module-Lattice-Based Digital Signature Standard (ML-DSA). Like Kyber, Dilithium's security is also based on the MLWE problem. Its purpose is to provide authenticity and integrity. A signer uses their private key to generate a digital signature for a message, and a verifier can use the signer's public key to confirm that the signature is valid and that the message has not been altered. Dilithium is designed for general-purpose use, offering a good balance between the size of its keys and signatures and the speed of its signing and verification operations. Its structured design, which uses a technique called rejection sampling, helps provide resilience against certain implementation attacks and avoids the need for stateful processes, which can be problematic in embedded systems.

**SPHINCS+ (SLH-DSA)**

As a conservative alternative to the lattice-based signatures, NIST also standardized SPHINCS+, published as FIPS 205, the Stateless Hash-Based Digital Signature Standard (SLH-DSA). SPHINCS+ is a stateless hash-based signature scheme. Its primary strength is its minimal and well-understood security assumptions; its security relies solely on the properties of the underlying cryptographic hash function. This makes it an attractive option for applications requiring the highest level of long-term security assurance, as it is believed to be resilient against a wide range of future cryptanalytic attacks. However, this robustness comes at a significant practical cost. Compared to Dilithium, SPHINCS+ signatures are much larger, and its signing process is orders of magnitude slower. While verification is reasonably fast, the slow signing and large signature size make it unsuitable for applications with tight latency, bandwidth, or storage constraints, such as high-frequency message authentication in a robotic control loop.

**1.3. Evaluation of Alternative Candidates: The Case for Falcon Signatures**

Beyond the primary standards, the NIST competition also advanced other algorithms as finalists, one of which, Falcon, presents a compelling set of trade-offs for specific industrial use cases. Falcon is a lattice-based digital signature scheme, but unlike Dilithium, its design is based on the NTRU framework and the Short Integer Solution (SIS) problem over NTRU lattices. Its most notable features are its exceptionally compact signatures and fast verification speeds. Performance benchmarks conducted in the context of automotive systems—a close proxy for the embedded controllers found in robotics—indicate that while Falcon's key generation can be slow, its verification process is significantly faster and consumes less RAM than Dilithium's.

This performance profile makes Falcon a highly attractive alternative for use cases that are verification-heavy and where signature size is a critical constraint. For example, in a secure boot process, a device must verify a firmware signature at every startup. Fast verification is essential to minimize boot time. Similarly, for secure over-the-air (OTA) updates, smaller signature sizes reduce bandwidth consumption and download times. In these scenarios, the one-time cost of a slower key generation (which can be done offline in a secure environment) is easily offset by the repeated benefits of fast verification and compact signatures on the embedded device.

**1.4. Comparative Analysis of PQC Primitives: A Trade-off Study**

The analysis of these leading PQC candidates reveals that there is no single "best" algorithm. The choice is a context-dependent engineering decision that requires a careful evaluation of the trade-offs between security, performance, and size.

The selection of a cryptographic primitive is not a one-size-fits-all decision but rather a nuanced process of matching an algorithm's characteristics to the specific requirements of an application. The data from performance benchmarks and security analyses of the leading PQC candidates clearly illustrates a divergence in their suitability for different tasks. For instance, SPHINCS+ offers the highest level of security assurance, rooted in the simple and trusted properties of hash functions. This makes it a "worst-case" benchmark for performance but a "best-case" for conservative security design. However, its practical utility is severely hampered by extremely long signing times and large signature sizes, rendering it impractical for any latency-sensitive or resource-constrained HRC application, such as real-time command authentication.

At the other end of the spectrum, lattice-based schemes present a much more practical balance. CRYSTALS-Dilithium has emerged as a strong general-purpose standard, offering good all-around performance for key generation, signing, and verification. This makes it a reliable workhorse for a wide range of applications. However, a deeper look at the benchmarks reveals a critical distinction when compared to Falcon. For automotive use cases like secure boot and diagnostic access, where a constrained Electronic Control Unit (ECU) must frequently

*verify* signatures, the literature explicitly notes that "FALCON Verification is faster and requires less RAM than CRYSTALS-Dilithium. Therefore, FALCON fits this use case better". Falcon's compact signatures also provide a significant advantage in environments with limited storage or bandwidth.

This leads to a crucial architectural conclusion: a robust and optimized security layer for a heterogeneous HRC environment should not standardize on a single signature algorithm. Instead, it should employ a *hybrid* approach, strategically deploying different schemes based on the specific function and constraints of each component. A powerful central controller that signs many messages might use Dilithium, while a resource-constrained robotic arm controller that primarily verifies incoming commands or firmware updates would be better served by Falcon. This application-specific selection allows the overall system to benefit from the distinct advantages of each algorithm, creating a more efficient and effective security architecture than a monolithic approach would allow.

The following table synthesizes the key characteristics of these algorithms to provide a clear reference for such decision-making.

| Algorithm | Type | Underlying Hard Problem | Public Key Size (bytes) | Secret Key Size (bytes) | Signature/Ciphertext Size (bytes) | Relative Performance (KeyGen/Sign/Verify) | Key Strengths | Key Weaknesses/Constraints |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CRYSTALS-Kyber-768** | KEM | Module-LWE | 1184 | 2400 | 1088 | Fast / Fast / Fast | Excellent all-around performance; NIST standard for KEM. | Keys and ciphertexts are larger than classical counterparts. |
| **CRYSTALS-Dilithium-3** | Signature | Module-LWE | 1952 | 4000 | 3293 | Fast / Medium / Fast | Good general-purpose performance; NIST standard for signatures. | Larger signatures and keys compared to Falcon and classical schemes. |
| **Falcon-512** | Signature | NTRU-SIS | 897 | 1281 | 666 | Slow / Medium / Very Fast | Very small signatures; extremely fast verification; low RAM usage for verification. | Slow key generation; more complex implementation. |
| **SPHINCS+-128f** | Signature | Hash Function Security | 64 | 128 | 17088 | Medium / Very Slow / Fast | Very conservative security assumptions; relies only on hash functions. | Extremely large signatures; very slow signing process. |

Export to Sheets

Note: Sizes are for NIST security level 3 (comparable to AES-192) where applicable. Performance is a qualitative summary based on embedded benchmarks.

**2. Architectural Imperatives for Secure HRC Workspaces**

Having established the properties of the available post-quantum cryptographic tools, the focus must now shift to the system they are intended to protect. Designing a security layer for HRC workspaces is not merely a matter of replacing old algorithms with new ones. It requires a holistic architectural approach that considers the unique operational requirements, physical safety constraints, and hardware limitations of industrial cyber-physical systems. The integration of PQC, with its distinct performance and size characteristics, introduces a new set of challenges that ripple through the entire system stack, from the hardware root of trust to the application-level protocols.

**2.1. Defining the Security Requirements of HRC: Real-Time Integrity, Authenticity, and Safety**

The security objectives for an HRC environment must be defined with a precision that goes beyond the traditional Confidentiality, Integrity, and Availability (CIA) triad. While all three are important, their priority and specific meaning are shaped by the physical nature of the system.

* **Integrity and Authenticity:** These are the paramount security requirements. Every command sent to a robot and every piece of sensor data it reports must be cryptographically protected to ensure it has not been modified in transit (integrity) and that it originates from a legitimate source (authenticity). A failure of integrity or authenticity could lead to a robot executing an unintended and potentially dangerous physical action, directly endangering the human collaborator. These guarantees must be provided in real-time, as even a small delay in detecting a malicious command could be catastrophic.
* **Availability:** The system must remain operational. Security mechanisms cannot introduce so much latency that they interfere with the robot's real-time control loops or create a single point of failure that could halt production. Denial-of-service attacks, whether through network flooding or by exploiting computational bottlenecks in the security protocols, are a significant threat to availability.
* **Confidentiality:** While often the primary focus in IT security, confidentiality is typically a lower priority than integrity and availability in operational technology (OT) control networks. The content of a real-time motor command is less sensitive than the assurance that the command is authentic. However, confidentiality remains critical for certain types of data, such as proprietary manufacturing recipes, intellectual property embedded in firmware, or sensitive diagnostic information.
* **Safety:** This is an emergent property that depends on the successful implementation of the above security requirements. A secure HRC system is a prerequisite for a safe HRC system.

**2.2. Foundational Security Services**

To meet these requirements, a set of foundational security services must be established and made quantum-resistant. These services form the bedrock of trust for the entire HRC workspace.

**Secure Boot**

The root of trust for any embedded device, including a robot, is its boot process. A secure boot mechanism ensures that the device starts up using only authentic and unmodified firmware from the manufacturer. The process typically involves a digital signature: the firmware image is signed with the manufacturer's private key, and the device's bootloader uses the corresponding public key to verify this signature before executing the code. If the signature is invalid, the boot process is halted, preventing the execution of malicious or corrupted software. Migrating this service to PQC requires replacing the classical signature algorithm (e.g., ECDSA) with a quantum-resistant one.

**Authenticated Key Exchange (AKE)**

Secure communication between the robot, its controllers, and any human-machine interfaces (HMIs) requires the establishment of an authenticated, encrypted channel. An Authenticated Key Exchange (AKE) protocol is used for this purpose. A modern AKE protocol must provide both authentication (ensuring the parties are who they claim to be) and confidentiality for the session keys it establishes. This is typically achieved by combining a Key Encapsulation Mechanism (KEM) to establish a shared secret with a digital signature to authenticate the parties and the key exchange transcript, thereby preventing man-in-the-middle attacks. This is the fundamental mechanism used in protocols like Transport Layer Security (TLS) to secure network connections.

**Secure Over-the-Air (OTA) Updates**

Industrial robots and controllers have long lifecycles and must be updatable in the field to patch security vulnerabilities, fix bugs, and add new features. Secure Over-the-Air (OTA) updates are a critical capability for maintaining the security posture of deployed systems. The process is heavily reliant on digital signatures: the update package is signed by the manufacturer, and the device verifies the signature before applying the update. This ensures that only legitimate updates can be installed, preventing attackers from deploying malicious firmware.

**2.3. The Role of Hardware Security**

Software-based security measures alone are insufficient to protect the most critical assets of a cryptographic system: the private keys. If an attacker can gain control of the main processor and read the memory where private keys are stored, they can impersonate the device, sign malicious updates, and decrypt communications. To prevent this, a hardware root of trust is essential.

**Hardware Security Modules (HSMs) and Trusted Execution Environments (TEEs)**

HSMs are dedicated cryptographic processors designed to securely manage, store, and use cryptographic keys. They provide physical isolation, often in a separate, tamper-resistant chip, ensuring that private keys never leave the secure boundary of the module. Cryptographic operations like signing or decapsulation are performed inside the HSM, with only the results being returned to the main processor. TEEs provide a similar function through logical isolation, creating a secure "enclave" within the main CPU where sensitive code and data can be protected even from a compromised operating system. In the context of HRC, the private keys for PQC algorithms should ideally be provisioned and stored within an HSM or TEE on the robot's controller. This provides the strongest possible protection against key theft and is a fundamental requirement for a high-assurance security architecture.

**2.4. Addressing PQC-Induced Challenges**

The transition to PQC is not a simple drop-in replacement. The significantly larger key and signature sizes of quantum-resistant algorithms create substantial system-level challenges, particularly for the resource-constrained embedded systems common in robotics.

**Key Storage Limitations**

A critical conflict arises between the increased size of PQC keys and the limited secure storage available in the hardware designed to protect them. Automotive-grade microcontrollers, which are frequently used in robotics due to their robustness and safety features, often include HSMs with very limited dedicated flash memory—typically in the range of 128 KB to 256 KB. A single Dilithium-3 private key is 4 KB, and its public key is nearly 2 KB. While one key pair may fit, a system often needs to store multiple keys: a device identity key, keys for secure communication, keys for verifying updates, etc. Storing a certificate chain, where each certificate contains a large PQC public key and signature, can quickly exhaust the available secure memory. This forces architects into difficult trade-offs: use more expensive hardware with larger HSMs, reduce the number of keys stored securely (weakening the security model), or develop complex schemes to manage keys outside the HSM, which defeats its purpose.

**Protocol Constraints**

The larger cryptographic primitives of PQC also impact network protocols. For example, a TLS handshake involves exchanging certificates and public keys. With PQC, the size of these handshake messages can increase dramatically. This can exceed the maximum transmission unit (MTU) of a network link, leading to IP fragmentation, which can increase latency and reduce reliability. In some industrial or automotive diagnostic protocols, such as ISO-TP, older versions of the standard have message size limits that would require large PQC key containers to be segmented across multiple frames, adding complexity and overhead. This increased data transmission also has a direct cost implication for systems that rely on cellular connectivity.

**RAM and Flash Consumption**

Beyond secure storage, PQC also places a greater burden on the general-purpose memory of an embedded system. The code size of the PQC cryptographic libraries themselves is larger, consuming more of the limited program flash memory. During operation, cryptographic algorithms require significant RAM for intermediate computations. Buffering large public keys or signatures before they are processed or stored also consumes precious RAM. These increased memory requirements can strain the resources of low-cost microcontrollers and may require hardware upgrades, increasing the overall cost of the system.

The integration of PQC into embedded systems is thus a challenge that extends far beyond the algorithm itself. It necessitates a re-evaluation of the entire hardware and software architecture. The hardware, once seen as the ultimate solution for cryptographic security, now presents a primary bottleneck due to its physical memory limitations. This shift means that PQC migration is not just a software update; it is an architectural problem that demands careful co-design of hardware, protocols, and key management strategies to create a system that is both secure and practical.

The following table summarizes the cascading impact of PQC's characteristics on the foundational security services required for HRC.

| Security Function | PQC-Related Challenge | Impact on System Component | Consequence for HRC | Potential Mitigation Strategy |
| --- | --- | --- | --- | --- |
| **Secure Boot** | Large Signature Size, Large Public Key Size | HSM/TEE Storage, Non-Volatile Memory (NVM) | Increased boot time, Higher hardware cost for secure storage. | Use Falcon for its compact signatures and fast verification. Store only public keys, not full certificates, in secure memory. |
| **Authenticated Key Exchange (AKE)** | Large Key/Ciphertext Size, Computational Latency | Network Bandwidth, CPU, RAM | Increased communication latency, Potential for protocol fragmentation. | Optimize PQC libraries for the target platform. Use hardware acceleration. |
| **Secure OTA Update** | Large Signature Size | Network Bandwidth, NVM Storage | Longer download times, Increased data transmission costs. | Use Falcon to minimize signature size. Implement delta updates to reduce package size. |
| **Secure Logging** | Computational Latency (for signing logs) | CPU, Flash Memory (for log storage) | Potential for log message loss if signing cannot keep up with generation rate. | Use efficient symmetric MACs for high-frequency logs, anchored by periodic PQC signatures. |

**3. Performance and Implementation Challenges in Resource-Constrained Environments**

While architectural considerations define the macro-level challenges of PQC integration, a successful deployment hinges on understanding the micro-level performance and security issues of executing these complex algorithms on the resource-constrained hardware typical of industrial robotics. The processors in robots, sensors, and controllers are not high-end servers; they are often embedded microcontrollers like the ARM Cortex-M series, chosen for their low power consumption, real-time capabilities, and cost-effectiveness. The feasibility of PQC in HRC workspaces is therefore directly tied to its performance and security on these platforms.

**3.1. Benchmarking PQC on Embedded Platforms**

Recognizing the importance of embedded performance, the cryptographic community has invested significant effort in optimizing and benchmarking PQC algorithms on relevant hardware. NIST designated the ARM Cortex-M4 as a primary optimization target during its standardization process, leading to a wealth of performance data for this platform. Research has focused heavily on accelerating the core arithmetic operations that dominate the runtime of lattice-based schemes. For CRYSTALS-Kyber and CRYSTALS-Dilithium, this core operation is the Number-Theoretic Transform (NTT), a variant of the Fast Fourier Transform used for efficient polynomial multiplication. Optimization techniques have included hand-crafted assembly code, novel instruction sequences for modular reduction, and clever use of hardware features like floating-point registers for caching intermediate values. These efforts have yielded substantial speed-ups, with some studies demonstrating improvements of 15% to 37% in the performance of key arithmetic functions compared to baseline implementations.

Beyond CPUs, Field-Programmable Gate Arrays (FPGAs) offer another promising platform for PQC implementation. FPGAs allow for the creation of custom hardware accelerators, which can perform cryptographic operations in parallel, leading to significant performance gains over software-only implementations. They represent a middle ground between software on a general-purpose CPU and a full Application-Specific Integrated Circuit (ASIC), offering a path to hardware acceleration that is crucial for meeting the real-time demands of HRC systems.

**3.2. The Overhead of Quantum Resistance: Latency, Memory Footprint, and Power Consumption**

Despite these impressive optimization efforts, PQC algorithms inherently carry more overhead than their classical predecessors. Even on an optimized Cortex-M4, the execution times for NIST-standardized schemes are non-trivial. Key encapsulation and decapsulation for Kyber are typically measured in the low milliseconds, while signing and verification for Dilithium can take tens to hundreds of milliseconds. While this level of performance is acceptable for many applications like secure boot or establishing a communication session, it can become a bottleneck for tasks requiring very high-frequency cryptographic operations, such as signing every message in a hard real-time control loop.

Furthermore, performance optimizations often come with their own costs. For example, some of the most effective speed-up techniques for Kyber and Dilithium involve using larger intermediate data types to delay costly modular reduction operations or caching pre-computed values. These techniques trade faster execution time for increased RAM and stack usage. In an environment where every kilobyte of memory is precious, this trade-off must be carefully managed. An algorithm that is fast but requires more RAM than is available on the target microcontroller is not a viable solution.

**3.3. Lightweight PQC for the IoT Edge**

The HRC workspace is a heterogeneous environment. While the main robot controller may be equipped with a relatively powerful processor like a Cortex-A or a multi-core system, the periphery is often populated with much simpler, deeply embedded devices. These can include proximity sensors, smart actuators, RFID/NFC tags for asset tracking, or wearable safety devices for human operators. These "IoT edge" devices often use low-power 8-bit or 16-bit microcontrollers with extremely limited memory (kilobytes of RAM) and processing power, and are frequently battery-operated.

For these highly constrained devices, even an optimized implementation of a NIST standard like Dilithium may be too resource-intensive. This has spurred research into "lightweight" PQC, which aims to design algorithms specifically for this class of hardware. One promising approach involves asymmetric design, creating signature schemes that are "signer-optimal." Schemes like LiteQSign are designed to make the signing operation on the constrained device as fast and energy-efficient as possible, often reducing it to just a few symmetric-key operations (e.g., hashes). The computational burden is shifted to the verification step, which is performed on a more powerful device like an edge gateway or the central robot controller. This model is an excellent fit for the typical data flow in an HRC environment, where many simple devices periodically send authenticated data to a central hub. Additionally, the use of lightweight symmetric ciphers, such as the Ascon suite, for authenticated encryption is a critical component for securing the data payloads from these devices once a channel is established.

**3.4. The Practical Threat of Side-Channel Attacks (SCA)**

A cryptographic algorithm can be mathematically unbreakable but practically insecure. Real-world implementations of cryptographic algorithms run on physical hardware, which can inadvertently leak information about the secret operations being performed. These "side channels" include variations in power consumption, electromagnetic (EM) emissions, timing, or even sound. An attacker with physical proximity to a device can measure these side channels and use statistical analysis to recover secret keys, completely bypassing the mathematical security of the algorithm. This is not a theoretical threat; side-channel attacks (SCAs) are a well-established and practical attack vector against embedded systems.

Lattice-based PQC schemes have been shown to be particularly susceptible to SCAs. Their complex arithmetic operations, which often involve secret-dependent branching or memory access patterns, can create distinct, measurable leakages. Recent research has demonstrated highly efficient attacks against standard implementations of CRYSTALS-Kyber. By sending a series of specially crafted ciphertexts to the device and observing its EM emissions during the decapsulation process, an attacker can modulate the leakage in a way that reveals the coefficients of the secret key one by one. These attacks can be devastatingly effective, with some demonstrations showing full key recovery using only a handful of traces and simple analysis techniques, avoiding the need for complex machine learning or profiling. This proves that deploying a mathematically secure PQC algorithm with a naive or unhardened implementation is insufficient; it can create a false sense of security while leaving the system wide open to physical attacks.

**3.5. The Necessity and Cost of Countermeasures**

To defend against side-channel attacks, implementations must be hardened with specific countermeasures. The most common and effective countermeasure is "masking." Masking involves splitting every sensitive intermediate value (including the secret key) into multiple, randomly generated "shares." The cryptographic computations are then performed on these shares in a way that the result is correct, but no individual share reveals information about the underlying secret value. To recover the secret, an attacker would need to simultaneously measure the leakages corresponding to all shares, which is exponentially more difficult.

However, these countermeasures are not free. Implementing masked arithmetic is significantly more complex than standard arithmetic, especially for PQC schemes that mix different types of mathematical operations (e.g., Boolean and arithmetic). This requires special conversion algorithms to move between different masking schemes, adding further complexity. The result is a substantial performance penalty. Masked implementations can be several times slower and require more memory than their unmasked counterparts.

This creates a fundamental and challenging cycle for developers of secure embedded systems. The initial drive is to optimize PQC algorithms for raw speed to meet real-time constraints. However, these very optimizations, which often aim for regular, predictable execution paths, can introduce or amplify side-channel leakages. To patch these leakages, masking and other countermeasures must be applied, which in turn degrades performance, potentially pushing the execution time back outside the acceptable limits for the application. This cyclical tension demonstrates that security and performance cannot be treated as independent goals. A secure and functional PQC implementation for an HRC system requires a holistic co-design approach, where the algorithm, the implementation techniques (including countermeasures), and the target hardware platform are considered together from the outset. A software-only approach that ignores physical security is likely to be either insecure or too slow to be practical. Hardware acceleration of masked operations, for example on an FPGA, may be a necessary step to break this cycle and achieve both high performance and robust side-channel resistance.

**4. Designing a PQC Security Layer: A Synthesis of Approaches**

The preceding analysis has established the characteristics of post-quantum algorithms, the security imperatives of HRC workspaces, and the profound challenges of implementing PQC on resource-constrained hardware. The final step is to synthesize these findings into a coherent and practical architectural design. A successful PQC security layer for HRC cannot be a monolithic, one-size-fits-all solution. The environment is inherently heterogeneous, comprising powerful controllers, real-time actuators, and simple, low-power sensors. An effective architecture must reflect this heterogeneity, strategically applying different cryptographic solutions to different parts of the system based on their specific constraints and functional requirements. This section proposes such a multi-layered architecture, drawing directly from the trade-offs and best practices identified in the literature.

**4.1. A Proposed Multi-Layered Security Architecture for HRC Systems**

The proposed architecture is structured in layers, addressing security from the foundational hardware root of trust up to the application and data analytics levels. This layered approach provides defense-in-depth and allows for the pragmatic selection of the most appropriate PQC primitives for each specific task. The goal is to build a system that is secure against quantum threats, performant enough for real-time collaboration, and scalable to the needs of a modern industrial facility.

**4.2. Layer 1: The Root of Trust – A PQC-Based Secure Boot Mechanism**

* **Component:** This layer applies to all computationally capable devices whose software integrity is critical, primarily the main robot controller and any associated safety Programmable Logic Controllers (PLCs).
* **Requirement:** The highest priority is to establish a root of trust by ensuring that the device boots with authentic, unmodified firmware. The verification of the firmware signature is a critical step in the boot sequence, and its execution time directly contributes to the overall system startup time. In an industrial setting, minimizing downtime is crucial, so a fast boot process is highly desirable.
* Proposed Solution: The firmware images for these critical components should be signed using the **Falcon** digital signature algorithm. The rationale for this choice is threefold and directly supported by the comparative analysis in the literature. First, Falcon produces exceptionally small signatures compared to other PQC candidates like Dilithium and especially SPHINCS+. This minimizes the storage overhead in the device's non-volatile memory, which is often a constrained resource. Second, and most importantly for this use case, Falcon's verification operation is significantly faster and requires less RAM than that of Dilithium. This directly translates to a shorter boot time. Third, the primary drawback of Falcon—its slow key generation—is irrelevant in this context, as key generation is a one-time, offline process performed by the manufacturer in a secure environment. The device itself only ever performs the fast verification operation. The public verification key must be stored in a secure, immutable location on the device, ideally provisioned into a Hardware Security Module (HSM) or protected write-once memory to prevent tampering.

**4.3. Layer 2: Secure Communication Channels – A Hybrid PQC Authenticated Key Exchange Protocol**

* **Component:** This layer secures the primary communication links within the HRC workspace, such as the network connection between the robot controller, edge gateways that aggregate sensor data, and the Human-Machine Interfaces (HMIs) used by human operators.
* **Requirement:** The goal is to establish a mutually authenticated, confidential communication channel to protect the integrity and privacy of commands, telemetry, and other operational data. The protocol must be resilient to man-in-the-middle (MitM) attacks.
* Proposed Solution: An Authenticated Key Exchange (AKE) protocol should be implemented using a combination of **CRYSTALS-Kyber** and **CRYSTALS-Dilithium**. This construction mirrors the approach being taken to create PQC-enabled versions of standard protocols like TLS. The protocol would proceed as follows:
  1. The initiator (e.g., HMI) generates an ephemeral key pair and uses the responder's (e.g., robot controller's) long-term Kyber public key to encapsulate a shared secret.
  2. The initiator sends the resulting ciphertext to the responder.
  3. The responder uses its long-term Kyber private key to decapsulate the shared secret.
  4. To provide authentication and prevent MitM attacks, both parties use their long-term Dilithium private keys to sign the entire transcript of the exchange (or key parts thereof).
  5. The final session key is derived from the shared secret established via Kyber. This approach leverages the strengths of both NIST standards: Kyber's high efficiency for key establishment and Dilithium's balanced performance as a general-purpose signature for authentication.

**4.4. Layer 3: Application-Level Security – Securing ROS 2 and DDS Communications**

* **Component:** This layer addresses the security of the middleware, which forms the communication backbone for many modern robotic systems. The Robot Operating System (ROS), particularly its latest version ROS 2, is a de facto standard in robotics research and is increasingly used in industrial applications. ROS 2 is built on top of the Data Distribution Service (DDS) standard for its underlying message-passing.
* **Requirement:** The default configurations of ROS often lack robust security features, leaving the publish-subscribe topics, services, and actions vulnerable to unauthorized access, sniffing, and message injection on the local network. It is essential to secure this middleware layer to protect the integrity of intra-robot communication (e.g., between the perception stack and the motion planner).
* Proposed Solution: Security should be integrated by extending the DDS Security specifications to support PQC algorithms. Emerging research demonstrates the feasibility of creating a "PQSec-DDS" by integrating PQC primitives into the five security plugins defined by the DDS standard: Authentication, Access Control, Cryptography, Logging, and Data Tagging. In this model,

**CRYSTALS-Kyber** and **CRYSTALS-Dilithium** would be used to secure the discovery process and establish encrypted sessions between DDS participants (nodes). This involves creating PQC-based identity certificates and using the PQC AKE protocol from Layer 2 to secure the underlying data channels. This approach provides a standards-compliant way to enforce fine-grained security policies within the robotics application itself, building upon the secure network channels established at Layer 2.

**4.5. Layer 4: Securing the Edge – Lightweight Authentication for Constrained Devices**

* **Component:** This layer focuses on the vast number of simple, resource-constrained devices at the edge of the HRC workspace. This includes proximity sensors, RFID/NFC readers, simple actuators, and wearable safety devices (e.g., a smart vest or wristband) for human workers.
* **Requirement:** These devices need to provide authenticity for the data they transmit (e.g., "this proximity alert is from sensor X") with minimal impact on their computational resources, memory, and battery life. Real-time verification is not always necessary, as the data is often consumed by a more powerful upstream system.
* Proposed Solution: A lightweight, signer-optimal PQC signature scheme should be employed. A strong candidate is a hash-based signature scheme or a specialized construction like **LiteQSign**. These schemes are designed to make the signing operation on the constrained device extremely efficient, often requiring only a small, constant number of hash function calls. The private key can be a single small seed, minimizing storage requirements. The computationally intensive verification process is offloaded entirely to the recipient of the data, such as an edge gateway or the main robot controller, which has ample resources for the task. This asymmetric distribution of computational load creates a highly scalable and energy-efficient model for authenticating data from potentially thousands of simple devices in a factory, ensuring their trustworthiness without overburdening their limited capabilities.

**4.6. Layer 5: Privacy-Preserving Collaboration – Exploring Fully Homomorphic Encryption (FHE)**

* **Component:** This is a forward-looking layer that addresses data analytics platforms, which may reside at the edge or in the cloud, processing data from multiple robots, sensors, and workers.
* **Requirement:** A key goal of Industry 5.0 is to use data to optimize processes and enhance worker well-being. This may involve analyzing sensitive operational data or even biometric data from worker wearables to identify ergonomic risks or predict potential collisions. Performing such analysis requires strong privacy guarantees.
* Proposed Solution: The exploration and eventual adoption of **Fully Homomorphic Encryption (FHE)** should be pursued for non-real-time, privacy-critical analytics. FHE is a transformative cryptographic technique that allows computations to be performed directly on encrypted data without ever decrypting it. A third party, such as a cloud service, can process encrypted sensor data from robots and workers to train machine learning models or run statistical analyses. The service receives an encrypted result, which can only be decrypted by the data owner. This allows for valuable insights to be derived from sensitive data without compromising the privacy of individuals or the confidentiality of proprietary processes. While FHE is still considered computationally intensive and not yet feasible for real-time control, the literature identifies it as the "holy grail" of cryptography. Its application in offline optimization tasks aligns perfectly with the human-centric and data-driven goals of Industry 5.0, representing a crucial technology for building truly trustworthy and privacy-preserving collaborative systems in the future.

**5. Recommendations and Future Research Directions**

The migration to post-quantum cryptography is one of the most significant security transitions in the history of computing. For the safety-critical and long-lived systems of Industry 5.0, this transition is not merely an option but a necessity. The synthesis of the current literature provides a clear, albeit challenging, path forward. This concluding section distills the architectural proposals into a set of actionable recommendations for system designers and identifies the most pressing gaps in the current research landscape that must be addressed to ensure a secure and efficient transition.

**5.1. Algorithm Selection Guidance for a Heterogeneous HRC Environment**

The core takeaway from this review is that a "one-size-fits-all" approach to PQC algorithm selection is suboptimal and impractical for a complex HRC environment. System architects must adopt a heterogeneous strategy, choosing the right tool for the right job based on the specific constraints and requirements of each component. The following provides a summary of the recommendations derived from the proposed multi-layered architecture:

* **For High-Assurance Firmware Integrity (Secure Boot):** Prioritize fast verification and small signature size. **Falcon** is the recommended choice for signing firmware images for robot controllers and safety PLCs due to its superior verification performance and compact signatures, which minimize boot time and storage overhead.
* **For General-Purpose Secure Channels (AKE/TLS):** Prioritize balanced, standardized algorithms. A combination of **CRYSTALS-Kyber** for key encapsulation and **CRYSTALS-Dilithium** for authentication provides a robust, high-performance solution for securing network links between controllers, gateways, and HMIs.
* **For Highly Constrained Edge Devices (Sensors, Wearables):** Prioritize signer-side efficiency (computation and power). Employ a lightweight, signer-optimal signature scheme like **LiteQSign** or a stateful hash-based signature if state management is feasible. This offloads the heavy computational work to more powerful verifiers, enabling scalability and preserving the battery life of edge devices.
* **For Conservative, Long-Term Archiving:** Where performance is not a concern but maximum security assurance is required (e.g., signing a root certificate authority key), **SPHINCS+** should be considered as a fallback due to its minimal security assumptions.

**5.2. A Phased Migration Strategy Towards a Fully Quantum-Resistant Architecture**

The transition from classical cryptography to PQC cannot happen overnight. It will be a gradual process that requires careful planning to maintain security and interoperability. A pragmatic, phased migration strategy is recommended:

1. **Inventory and Assessment:** The first step for any organization is to conduct a thorough inventory of all cryptographic assets within their industrial systems. This means identifying every instance where public-key cryptography is used, what it protects, and the hardware it runs on.
2. **Hybrid Deployment:** The initial deployment phase should focus on a "hybrid" approach. This involves using a PQC algorithm alongside a classical one (e.g., encapsulating a secret with both Kyber and ECDH and concatenating the ciphertexts). This strategy provides immediate protection against "Harvest Now, Decrypt Later" attacks while maintaining backward compatibility with existing systems that have not yet been upgraded.
3. **Transition to PQC-Only:** As the ecosystem matures and all components within the HRC workspace become PQC-capable, the classical algorithms can be phased out. This final step to a PQC-only mode reduces computational overhead and protocol complexity.
4. **Embrace Crypto-Agility:** The PQC landscape is still evolving. It is crucial to design systems to be "crypto-agile," meaning that cryptographic algorithms can be updated or replaced easily without requiring a complete system redesign. This is essential for responding to future cryptanalytic breakthroughs or changes in standards.

**5.3. Identifying Gaps in the Literature**

This review has synthesized the current state of knowledge, but it has also highlighted several areas where further research is critically needed:

* **Real-World HRC Benchmarking:** While performance benchmarks on platforms like the Cortex-M4 exist, there is a lack of comprehensive studies that evaluate PQC performance on actual robotics-grade hardware, running within the context of a real-time operating system (RTOS) and a full robotics middleware stack like ROS 2. Such studies are needed to understand the true end-to-end latency impact of PQC.
* **Side-Channel Resistance of PQC in Industrial Settings:** The threat of side-channel attacks is well-documented in laboratory settings. Research is needed to assess the practical risk of these attacks in noisy industrial environments and to develop efficient, low-overhead countermeasures suitable for embedded robotics platforms.
* **Standardization of PQC for Industrial Protocols:** While efforts are underway to integrate PQC into internet protocols like TLS, there is a need for focused standardization of PQC profiles for industrial middleware, particularly DDS. A standardized way to integrate PQC into the DDS Security plugins would greatly enhance interoperability and security for robotic systems.
* **Hardware-Software Co-Design for SCA-Resistant Accelerators:** The tension between performance and side-channel resistance points to a critical need for hardware-software co-design. Future research should focus on developing PQC accelerators for FPGAs or ASICs that are not only fast but are also designed from the ground up to be resistant to side-channel leakage, for example, by implementing masked logic directly in hardware.

**5.4. Concluding Remarks**

The emergence of Industry 5.0 promises a future of manufacturing that is more efficient, resilient, and collaborative. The security and safety of the human-robot workspaces at the heart of this vision are non-negotiable. The threat posed by quantum computing represents a fundamental challenge to the cryptographic foundations upon which this security is built. The transition to Post-Quantum Cryptography is a complex and multifaceted undertaking, fraught with architectural, performance, and implementation challenges. However, as this review has shown, the cryptographic community has provided a powerful toolkit of quantum-resistant algorithms, and a path forward is visible. By adopting a strategic, heterogeneous, and hardware-aware approach to security design, it is possible to build a PQC layer that can meet the demanding requirements of HRC systems. This proactive integration of PQC is not merely a technical upgrade; it is an essential investment in ensuring the long-term safety, trustworthiness, and resilience of the future industrial landscape.