Computers & Security 167 (2026) 104917

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

Cybersecurity in the quantum era: Assessing the impact of quantum computing on infrastructure

Yaser Baseri [a,*], Vikas Chouhan [b], Ali Ghorbani [b]

[a] Department of Computer Science and Operations Research, Universite de Montreal, Canada
[b] Canadian Institute for Cybersecurity (CIC), University of New Brunswick, Canada

ARTICLE INFO

Keywords:
Quantum computing
Cyber threats
Cryptographic techniques
Infrastructure security
Quantum-resistant cryptography

ABSTRACT

The advent of quantum computing marks a transformative era in cybersecurity, challenging traditional cryptographic frameworks and broadening the horizons of computational capabilities. This paper navigates the transition to a quantum-resistant security framework, offering a critical analysis of encryption methods essential for the protection of critical infrastructure and cloud services in the quantum era. It meticulously evaluates the effects of quantum computing across various infrastructure layers, including applications, data, runtime, middleware, operating systems, virtualization, hardware, storage, and networks. Through a nuanced exploration of potential vulnerabilities and the evolving threat landscape conferred by quantum technologies, our study delivers strategic insights for the development of effective countermeasures. Our comprehensive inquiry underscores the pivotal shift necessitated by the disruptive potential of quantum computing, advocating for preemptive and inventive security strategies. It calls for concerted, multi-sectoral collaboration to cultivate resilient, quantum-resistant cryptographic practices. Through meticulous research, we articulate the emergent threat vectors posed by quantum technologies and evaluate post-quantum cryptographic solutions, laying the groundwork for safeguarding both current and prospective infrastructural and cloud ecosystems. Significantly, this study introduces a tailored security blueprint encompassing nine critical infrastructure components, reinforcing each domain's defenses against quantum-induced cyber threats. Our strategic vulnerability assessment provide essential countermeasures, equipping stakeholders with the ability to navigate the complex, quantum-threatened landscape. These contributions are integral for those tasked with critical infrastructure stewardship, influencing design, implementation, and policy formulation. In essence, this paper not only forecasts quantum threats but also offers a sophisticated, actionable framework for strengthening infrastructure and cloud environments against the multifaceted challenges of the quantum era.

1. Introduction

The emergence of quantum computing represents a paradigm shift in the landscape of infrastructure security. Quantum computing's significant implications infiltrate every layer of our digital infrastructure, casting a shadow of uncertainty over the realm of cybersecurity. Renowned for their unparalleled computational capabilities, quantum computers present a formidable challenge to the traditional cryptographic methods that have long served as the cornerstone of data protection. Notably, encryption techniques like RSA and ECC, which have historically safeguarded data integrity across myriad infrastructure and cloud systems, now face an unprecedented and imminent threat (Shor, 1999; Bernstein et al., 2009; Pacific Northwest National Laboratory, 2023).

Quantum computing's extraordinary capabilities in tackling complex problems, such as factorizing large numbers and computing discrete logarithms, poses a significant adversary to the security of our digital infrastructure and cloud-based systems (Lipman, 2023; Lee, 2023). As quantum computers continue to advance in power and capacity, they assume the role of potential adversaries capable of undermining well-established encryption techniques. The repercussions of such a scenario are profound, with malicious actors potentially gaining unauthorized access to and control over critical data. This vulnerability extends its impact from individuals and organizations to entire nations, with far-reaching consequences (Mosca, 2018; Gidney and Ekerå, 2019).

* Corresponding author.
E-mail addresses: yaser.baseri@umontreal.ca (Y. Baseri), vikas.chouhan@unb.ca (V. Chouhan), ghorbani@unb.ca (A. Ghorbani).

https://doi.org/10.1016/j.cose.2026.104917
Received 29 August 2025; Received in revised form 1 March 2026; Accepted 6 April 2026
Available online 8 April 2026
0167-4048/Crown Copyright © 2026 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1.2. Contributions

This paper makes significant contributions to quantum-safe infrastructure security by combining comprehensive analysis with practical frameworks:

1. **Dual-Phase Infrastructure Security Analysis**: We present a comprehensive systematic examination of cybersecurity implications across both *pre-migration (classical cryptography)* and *post-migration (post-quantum cryptography)* phases, analyzing threats across nine critical infrastructure layers: applications, data, runtime, middleware, operating systems, virtualization, hardware, storage, and networks.
2. **Comprehensive Attack Vector Taxonomy with Countermeasures**: We identify and classify *numerous attack vectors* spanning pre- and post-migration phases, and provide detailed countermeasures with clear responsibility assignments for vendors, implementers, and users. Structured STRIDE-based tables map vulnerabilities to exploits and mitigation strategies, offering a practical reference for infrastructure operators.
3. **Strategic Migration Framework**: We deliver actionable guidance for quantum-safe transitions, including *algorithm selection criteria, integration pathways, and infrastructure-aware deployment strategies* that balance security robustness against operational constraints and interoperability requirements.
4. **Empirical Benchmarking of Post-Quantum Standards**: We conduct a comprehensive performance characterization of *NIST-standardized PQC algorithms* (ML-KEM, ML-DSA, SLH-DSA, Falcon, HQC), evaluating *computational latency, communication overhead, and scalability*. These benchmarks provide a quantitative foundation for algorithm selection in infrastructure deployments.

Collectively, these contributions establish both the theoretical foundations and practical tools required to guide infrastructure operators, cloud providers, and policymakers in preparing for secure, scalable, and standards-aligned quantum-safe transitions.

1.3. Organization

This paper is organized as follows: Section 2 provides a background on the intersection of quantum computing and cybersecurity, with a focus on quantum-resistant cryptographic methods. Section 3 evaluates the cybersecurity risks associated with quantum computing, encompassing both classical and emerging PQC, and strategies for developing quantum-safe infrastructures. Section 4 discusses the pre-migration phase, examining the impact of quantum computing on cybersecurity in critical infrastructures like cloud computing and the challenges it poses to traditional cryptography. Section 5 delves into the post-migration phase, analyzing how digital security is adapting through quantum-resistant cryptography and its implications for key infrastructures, particularly cloud environments. The paper concludes in Section 6 with a summary and future research directions, offering consolidated insights into the role of quantum computing in cybersecurity.

2. Related works

This section comprehensively reviews the advancements in quantum computing, the evolution of PQC, and their cumulative impact on digital infrastructure and cybersecurity. It emphasizes the transition of quantum computing from theoretical concepts to applications that directly challenge traditional cryptographic protocols. The section also highlights gaps in current research, particularly in addressing the range and complexity of quantum-induced cyber threats to both present and future infrastructure, including cloud ecosystems.

2.1. Recent trends and future directions in quantum-resistant security

Recent progress in quantum computing raise concerns about its potential threat to conventional cryptographic systems. Mashatan (2022) emphasizes the critical need for a quantum-resistant roadmap in cybersecurity, including the anticipation of timelines for certifying quantum-resistant standards and preparation against potential quantum attacks. Barker et al. (2021) propose a comprehensive framework emphasizing a systematic shift to PQC, highlighting the need for robust testing, well-planned integration timelines, and a thorough strategy for achieving quantum-safe enterprise systems in response to the challenges posed by quantum computing. Shapna Akter (2023) provide a comprehensive survey of Quantum Cryptography for Enhanced Network Security, shedding light on the research landscape and future directions in this domain. These studies collectively represent a concerted effort to navigate the evolving cybersecurity landscape in the quantum era, aiming to pave the way for robust solutions in quantum-resistant security.

2.2. Evolution of quantum computing and cryptography

Quantum computing has swiftly transitioned from theoretical exploration to practical applications, with significant implications in the field of cryptography. The groundbreaking work by Shor (1994) revealed the vulnerability of conventional cryptographic protocols, such as RSA and ECC, against quantum computing attacks, specifically how quantum algorithms could exploit mathematical shortcuts for breaking these systems. This breakthrough has catalyzed the development of PQC, which is focused on designing algorithms secure against both classical and quantum computational threats. Recent advancements in quantum computing technology, such as those by Google and IBM, illustrate an accelerated approach towards practical applications, underscoring the urgency of PQC development (Arute et al., 2019; Castelvecchi, 2017; Bernstein and Lange, 2017).

2.3. Development and standardization of PQC

In response to quantum threats, research and standardization in PQC have accelerated. Organizations like the National Institute of Standards and Technology (NIST) are evaluating a variety of cryptographic methods, with lattice-base, hash-based, code-based, and isogeny-based cryptography each offering unique strengths and potential use cases. For instance, lattice-based cryptography is gaining attention due to its efficiency and resistance to quantum attacks. NIST's ongoing efforts to standardize quantum-resistant algorithms are shaping the future landscape of secure digital communication (Alagic et al., 2025; NIST, 2022; Mosca, 2018; National Institute of Standards and Technology (NIST), 2023e, 2022). The contributions by Kumar and Garhwal (2021) in exploring new algorithmic paradigms have further enriched this domain, including developments in algorithm agility and adaptability.

2.4. Impact of quantum computing on infrastructure security

The rapid evolution of quantum computing poses significant challenges to digital infrastructure, necessitating a comprehensive understanding of potential vulnerabilities introduced by quantum technologies. Beyond the algorithmic threats discussed previously, the structural impact on networked systems is profound. Lindsay (2020) analyzes this potential impact on cryptographic protocols, providing insights into the strategic alignment necessary to fortify security in the quantum era. Furthermore, Mangla et al. (2023) focus on specific vulnerabilities in 5G networks and anticipated security challenges for future 6G networks, highlighting the need to fortify security amid the evolving telecommunications landscape.

2.5. Mitigating quantum threats in digital infrastructures

Addressing the multifaceted challenges posed by quantum advancements, pivotal works offer diverse perspectives. Mosca and Piani's Quantum Threat Timeline Report 2024 is expected to provide a chronological assessment of the quantum threat landscape (Mosca and Piani, 2024). Faruk et al. (2022) explore quantum computing's dual nature as both a potential threat and a solution in the cybersecurity domain.

2.6. Strategies of major cloud providers in securing infrastructure against quantum threats

Quantum computing presents groundbreaking opportunities and significant cybersecurity challenges, impacting cloud infrastructure. Major cloud providers such as Amazon Web Services (AWS) (Amazon Web Services, 2023), Microsoft Azure (Microsoft Azure, 2023), and Google Cloud Platform (GCP) (Google Cloud Platform, 2023), are actively developing strategies to address these challenges. These providers focus on PQC (Bernstein and Lange, 2017) to secure against quantum computing threats. They adopt hybrid cryptographic solutions (D et al., 2025), combining current encryption methods with quantum-resistant algorithms (Mosca, 2018), and emphasize cryptographic agility (Rosenberg, 2019).

Providers are balancing the trade-offs between security and computational performance, as some quantum-resistant algorithms can be less efficient (Mosca, 2018). Transitioning to PQC involves substantial system updates, demanding careful integration to maintain system integrity and backward compatibility (Quantum-Readiness Working Group (QRWG) of the Canadian Forum for Digital Infrastructure Resilience (CFDIR), 2023). Their efforts encompass all layers of infrastructure: developing quantum-safe applications (The White House, 2022), protecting data with quantum-resistant encryption, securing runtime environments (National Institute of Standards and Technology, 2024a), updating middleware protocols, integrating quantum-safe features in operating systems, fortifying virtualization layers (TechTarget, 2021), implementing quantum-resistant hardware security modules (Crypto4A, 2023), enhancing storage encryption (Entrust, 2023), and adopting quantum-safe networking protocols (IBM, 2023; Baseri et al., 2024). These strategies ensure a comprehensive examination of quantum threats across the entire stack, with a focus on collaboration with hardware vendors, cryptographic communities, and researchers (Gyongyosi and Imre, 2019). Compliance with global quantum-safe standards, like those spearheaded by NIST (National Institute of Standards and Technology (NIST), 2023a), is also emphasized. Providers update their threat models to encompass potential quantum computer attacks, reinforcing their security posture (Gidney and Ekerå, 2021).

In conclusion, AWS, Azure, and GCP are addressing the challenges posed by quantum computing (Gyongyosi and Imre, 2019). Their strategies focus on implementing quantum-resistant technologies, collaborative research, standardization, and proactive security measures, highlighting their commitment to safeguarding infrastructure in the quantum era. However, the evolving nature of quantum technologies means that these strategies will require ongoing updates and vigilance.

2.7. Gaps in current research and our focus

While there has been significant progress in PQC and understanding the broader implications of quantum computing, there remains a critical gap in detailed analyses of quantum-induced cyber threats, particularly those affecting existing and future infrastructures and cloud environments. Previous research has mostly centered on general aspects of PQC and quantum computing's implications, often missing the finer details of these emerging challenges. Our research aims to fill this gap, delving into the nuanced threats posed by quantum advancements and formulating novel strategies to protect infrastructures against them. This work ventures into a vital yet scarcely explored domain, fortifying defenses in the rapidly evolving quantum computing and cybersecurity landscape.

3. Cryptographic standards and quantum computing: Cyber impact and risk assessment

This paper presents an elaborate security framework designed to systematically tackle the broad range of cybersecurity threats precipitated by the advent of quantum computing. These threats pertain to both current and forthcoming infrastructure, including cloud-based environments. Our framework introduces a detailed security blueprint, focusing on nine critical infrastructural elements: applications, data, runtime, middleware, operating systems, virtualization, hardware, storage, and networks. This framework aims to address the vulnerabilities and risks associated with these components in the face of emerging quantum computing capabilities, ensuring robust protection for both existing and future cloud and infrastructure systems.

Before we dive into our discussion, it is crucial to understand the seriousness of the quantum computing threat Baseri et al. (2025a). The vulnerability of cryptographic systems is at the core of the risks to infrastructure and cloud security. Therefore, we need to assess how quantum computing impacts both traditional and PQC, which is considered a potential solution for quantum-safe cryptographic systems. Evaluating these risks is essential for companies to effectively prioritize threats by considering the potential consequences and the likelihood of vulnerabilities being exploited. In the context of transitioning to quantum-safe solutions, we use a risk assessment methodology guided by the recommendations of the National Institute of Standards and Technology (NIST).

Our security framework is specifically designed to counter the emerging threats introduced by quantum computing in the field of current and future cryptography. In Fig. 2, you can see an overview of selected cryptographic standards provided by NIST (National Institute of Standards and Technology, 2020a, 2001b, 2020b, 2019b, 2001a, 2019a, 2018). With the impending arrival of quantum computers with significant capabilities, their impact on both public and symmetric cryptographic systems is unavoidable. Even the new quantum-safe cryptographic methods being considered for standardization by NIST are not entirely immune to vulnerabilities. In this section, we assess the security risks associated with the evolving threats posed by quantum computing to existing and potential cryptographic solutions.

3.1. Classic cryptographic standards and quantum computing: Assessing cyber risks

Cryptographic standards, the cornerstone of security in diverse applications and communications, primarily employ classical algorithms, both asymmetric and symmetric. These algorithms face a growing threat from quantum computing, which could break or weaken them significantly. The hard problems underpinning modern asymmetric cryptography, the cornerstone of secure communication, may no longer pose a substantial challenge for quantum attackers equipped with sufficiently powerful quantum computers. Utilizing established quantum algorithms, such as Shor's algorithm, these attackers can potentially breach the security of widely used asymmetric cryptographic systems. Similarly, symmetric cryptographic systems' security is also at risk. Quantum algorithms like Grover's algorithm and the Brassard et al.'s algorithm can undermine the security of symmetric cryptographic algorithms and communication protocols. Consequently, this section delves into the vulnerabilities and strengths of classical cryptographic algorithms as we approach the need for migration to potentially quantum-safe cryptographic alternatives.

3.1.1. Quantifying quantum threats

To understand the risks associated with quantum migration, it is imperative to predict the emergence of quantum computers and the resultant risks to classical cryptosystems. Our analysis examines the timeline for quantum computers to appear within the next 5 to

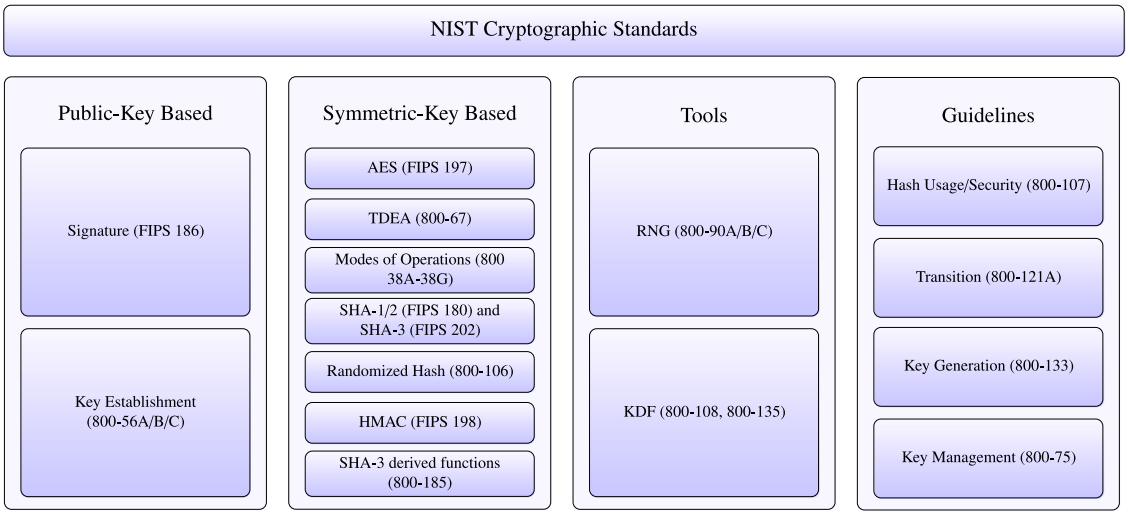


Fig. 2. NIST cryptographic standards.

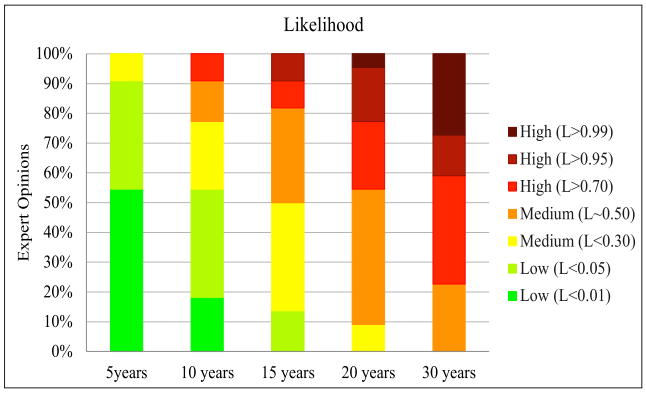


Fig. 3. Cumulative expert opinions related to quantum threat to classic cryptography.

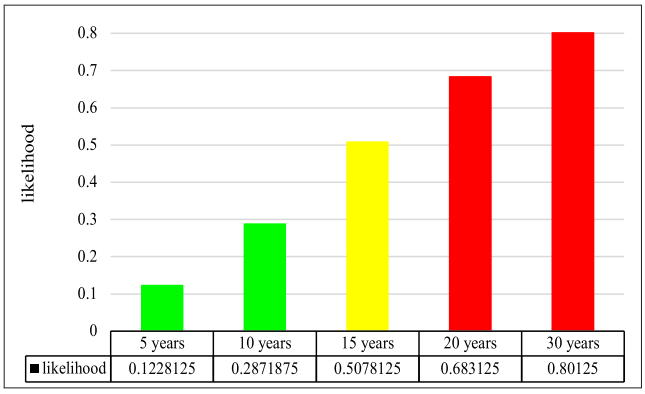


Fig. 4. Expected likelihood of quantum threat for classic cryptography within 30 years.

30 years. This analysis is built on a cumulative likelihood of significant quantum threats to classical cryptosystems.

Fig. 3 summarizes this evolution, incorporating insights from multiple quantum experts regarding the quantum threat timeline (Mosca and Piani, 2024). The "quantum threat" is defined as the probability of breaking RSA-2048 within 24 h using a quantum machine. These assessments can be extended to evaluate the likelihood of breaking other cryptographic algorithms based on their quantum security level, as presented in Fig. 4.

3.1.2. Expected likelihood of quantum threat

To assess the "expected likelihood of the quantum threat for classical cryptosystems" over various periods (5, 10, 15, 20, and 30 years), we aggregate predictions made by different experts who participated in the poll (Mosca and Piani, 2024). For each period, such as 5 years, we calculate the expected likelihood by multiplying the agreed-upon likelihoods of predictions for that period by the probability of those predictions, then summing them up.

$$E_{\text{period}_j}[\text{likelihood}] = \sum_{\omega_i \subseteq [0,1]} \text{likelihood}_{\text{period}_j}(\omega_i) \times \Pr_{\text{period}_j}(\omega_i)$$

Here, ω_i represents subsets of $[0,1]$, with their union equaling $[0,1]$. $\Pr_{\text{period}_j}(\omega_i)$ for each period period_j is determined by the fraction of expert opinions that agreed on prediction ω_i for that period, relative to the total number of predictions.

Our likelihood assessment categorizes quantum threat likelihood into three levels: low, medium, and high. As depicted in Fig. 4, the expected likelihood of a quantum threat within 10 years is low, within 15 years is medium, and beyond 20 years is high. Our evaluation adopts a medium likelihood within 15 years for the quantum threat to classical cryptosystems. This assumption can be adapted for other timeframes.

3.1.3. Quantum impact assessment

To conduct a classic algorithmic level risk assessment, we evaluate the impact of quantum threats on different classic cryptographic algorithms. The impact is determined based on the quantum security strength of each classic algorithm, as illustrated in Fig. 5. An impact is considered high if the algorithm's quantum strength is less than 64 bits, low if it is greater than or equal to 128 bits, and medium if it falls between these values. The final risk assessment combines both the likelihood and impact and is presented in Table 1.

3.1.4. Quantum impact assessment

To conduct a classic algorithmic level risk assessment, we evaluate the impact of quantum threats on different classic cryptographic algorithms as per the NIST suggested criteria (National Institute of Standards and Technology (NIST), 2025). The impact is determined based on the quantum security strength of each classic algorithm. We define quantum security strength as the effective logarithmic security level (in bits) against the best known quantum attacks, specifically

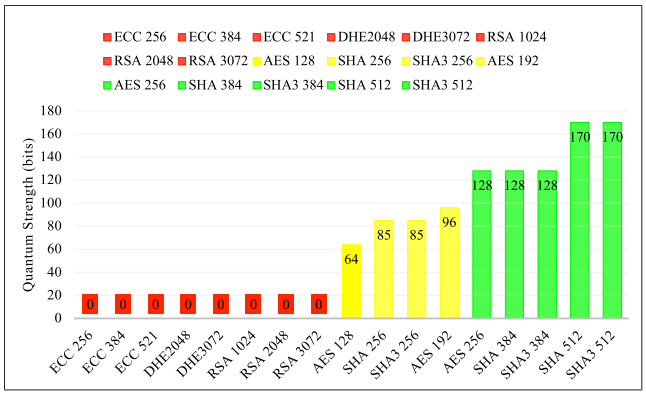


Fig. 5. Expected impact of quantum threat for classic cryptography.

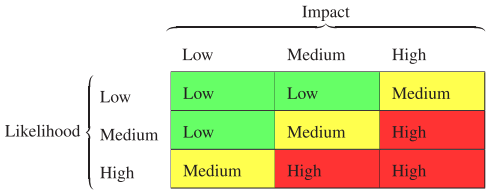


Fig. 6. Qualitative risk assessment based on likelihood and impact levels.

accounting for the quadratic speedup from Grover's algorithm for symmetric schemes and the exponential speedup from Shor's algorithm for asymmetric schemes. As illustrated in Fig. 5, an impact is considered high if the algorithm's quantum strength is less than 64 bits, low if it is greater than or equal to 128 bits, and medium if it falls between these values. The final risk assessment combines both the likelihood and impact and is presented in Table 1.

3.1.5. Risk evaluation

Table 1 offers a comprehensive insight into our findings regarding the assessment of conventional cryptographic algorithms before transitioning to quantum-safe cryptographic solutions. It provides an in-depth exploration of classical and quantum security attributes of these algorithms, their inherent vulnerabilities, and the emerging quantum threats. Furthermore, the table highlights potential quantum-resistant remedies and the attendant risks that quantum adversaries might exploit. The evaluation of these associated risks rests on a qualitative risk assessment that considers both likelihood and impact, as elucidated in Fig. 6. This holistic analysis serves as a valuable tool for assessing the security robustness of existing cryptographic algorithms and for devising effective mitigation strategies within the realm of classical cryptography.

3.2. Quantum-safe cryptographic candidates for standardization: Navigating quantum computing risks

As discussed earlier, the advent of quantum computing (QC) is poised to revolutionize the landscape of cryptographic algorithm attacks. While the impact of QC on symmetric cryptographic algorithms remains relatively contained, achieved through the adoption of longer keys or extended hash function outputs, public key cryptographic algorithms face a serious threat. This necessitates the replacement of current public key cryptographic algorithms and standards.

3.2.1. Transition to quantum-safe cryptographic algorithms

To safeguard against the emerging QC threat to widely-used public key cryptographic algorithms, the adoption of quantum-safe alternatives is imperative. The National Institute of Standards and Technology (NIST) has launched an initiative to standardize such algorithms through a competition aimed at identifying robust post-quantum cryptographic solutions for Key Encapsulation Mechanisms/Encryption (KEM/ENC) and digital Signatures (SIG). Several categories of post-quantum cryptographic algorithms have emerged, including lattice-based (Bos et al., 2018; Ducas et al., 2018; Fouque et al., 2018), code-based (Bernstein et al., 2017; Melchor et al., 2018; Aragon et al., 2017), hash-based (Bernstein et al., 2019), and isogeny-based (Jao and Feo, 2011) approaches. NIST has formalized these efforts by publishing post-quantum standards: FIPS 203 (National Institute of Standards and Technology (NIST), 2023b) (lattice-based key encapsulation mechanism), FIPS 204 (National Institute of Standards and Technology (NIST), 2023c) (lattice-based digital signature algorithm), and FIPS 205 (National Institute of Standards and Technology (NIST), 2023d) (hash-based digital signature algorithm). Additionally, NIST has selected HQC, a code-based key encapsulation mechanism, for standardization, with a draft FIPS expected in 2026 (Alagic et al., 2025). These standards, along with HQC selected for future standardization, are listed in Table 2, providing a regulatory and technical foundation for enterprises migrating to quantum-safe cryptographic infrastructures.

While NIST standardization provides quantum-resistant algorithms, two critical challenges remain for practical deployment. First, performance characteristics, including computational overhead, communication bandwidth, and scalability, must be quantified to assess real-world feasibility across diverse infrastructure contexts. Second, these algorithms remain vulnerable to implementation attacks, particularly side-channel attacks that exploit physical information leakage during cryptographic operations. Understanding both performance implications and residual security vulnerabilities is essential for informed migration strategies. The subsequent analysis systematically examines these two fundamental challenges.

3.2.2. Performance characterization of NIST standardized algorithms

Empirical performance analysis of NIST-standardized post-quantum cryptographic algorithms is essential for informed deployment decisions. This characterization examines computational latency, communication overhead, and scalability across key encapsulation mechanisms and digital signature algorithms at multiple security levels (L1/L3/L5 corresponding to AES-128/192/256), establishing quantitative foundations for evaluating quantum-safe transitions in real-world infrastructures.

Experimental methodology. Performance benchmarks utilized `liboqs v0.9.0` on Intel Core i7-10750H processors (6-core/12-thread, 2.6 GHz) with 16 GB RAM running Ubuntu 20.04. All measurements represent mean values across 1000 iterations with 95% confidence intervals.

Key encapsulation mechanisms. Table 3 summarizes computational latencies for NIST-standardized KEMs across security levels equivalent to AES-128/192/256 (L1/L3/L5). ML-KEM demonstrates consistent sub-60 μs latency across operations, optimal for latency-sensitive applications. HQC exhibits 5–16× higher overhead, reflecting complexity differences between lattice-based and code-based constructions.

Table 4 presents communication overhead characteristics. ML-KEM maintains predictable linear scaling with manageable bandwidth impact, while HQC's code-based structure produces significantly larger artifacts. Overhead factors relative to classical ECDH P-256 show ML-KEM introduces moderate 12–25× key size increases, whereas HQC imposes approximately 105–340× overhead, potentially limiting applicability in bandwidth-sensitive deployments.

Digital signature algorithms. Table 5 summarizes computational latencies for NIST-standardized signatures across L1/L3/L5 security levels. ML-DSA provides consistent sub-millisecond performance suitable for general-purpose deployments. Falcon exhibits expensive key generation (100–1000× slower than other operations) but efficient verification, ideal for certificate validation systems with infrequent key generation

Table 1
Classic cryptographic standards and quantum computing: Assessing cyber risks.

Crypto type	Algorithms	Variants	Key length (bits)	Strengths (bits)		Vulnerabilities	Quantum threats (STRIDE)	L	I	R	Recommended QC-resistant solutions
				Classic	Quantum						
Asymmetric	ECC (Turner and Brown, 2010)	ECC 256	256	128	0	Broken by Shor's Algorithm (Shor, 1994).	For digital signature: • Spoofing: Shor's Algorithm allows forging of digital signatures. • Tampering: Integrity checks can be bypassed due to signature forgery. • Repudiation: Valid signatures can be forged, denying the origin of the message. For KEM/ENC: • Information Disclosure: KEM/ENC algorithms can be broken, revealing encrypted data.	M	H	H	Algorithms presented in Table 2.
		ECC 384	384	256	0			M	H	H	
		ECC 521	521	256	0			M	H	H	
	FFDHE (Gillmor, 2016)	DHE2048	2048	112	0			M	H	H	
		DHE3072	3072	128	0			M	H	H	
	RSA (Moriarty et al., 2016)	RSA 1024	1024	80	0			M	H	H	
		RSA 2048	2048	112	0			M	H	H	
		RSA 3072	3072	128	0			M	H	H	
Symmetric	AES (Schaad, 2003)	AES 128	128	128	64	Weakened by Grover's Algorithm (Grover, 1996).	• Information Disclosure: Grover's algorithm reduces the effective key length, making brute-force attacks feasible.	M	M	M	Larger key sizes are needed.
		AES 192	192	192	96			M	M	M	
		AES 256	256	256	128			M	L	L	
	SHA2 (Eastlake 3rd and Hansen, 2011)	SHA 256	–	128	85	Weakened by Brassard et al.'s Algorithm (Brassard et al., 1997).	• Spoofing: Fake hash values can be created. • Tampering: Data integrity can be compromised by finding collisions.	M	M	M	Larger hash values are needed.
		SHA 384	–	192	128			M	L	L	
		SHA 512	–	256	170			M	L	L	
	SHA3 (Eastlake 3rd and Hansen, 2011)	SHA3 256	–	128	85			M	M	M	
		SHA3 384	–	192	128			M	L	L	
		SHA3 512	–	256	170			M	L	L	

Table 2
Quantum-safe cryptographic algorithms for standardization by NIST: Assessing cyber risks.

Algorithm	Type	Standardized As	FIPS Number	Available Attacks and Countermeasures[a]								Possible Countermeasures	L	I	R
				CA	TA	FA	SPA	APA	EM	TMP	CB				
CRYSTALS-Kyber	Lattice-based KEM	ML-KEM (Module-Lattice-Based Key-Encapsulation Mechanism)	FIPS 203 (National Institute of Standards and Technology (NIST), 2023b)			✓✓	✓✓	✓✓̸	✓✓	✓	✓✓	• **To defend against FA:** Mask the decryption process (Oder et al., 2016; Ravi et al., 2020a), or check the secret and error components of the LWE instances (Ravi et al., 2019b). • **To prevent SPA:** Mask the input (Hamburg et al., 2021), randomize the order of executed operations (Hamburg et al., 2021). • **To avoid APA:** Mask the Number Theoretic Transform (NTT) (Pessl and Primas, 2019) (no countermeasures in place for the attacks mentioned in (Dubrova et al., 2022)). • **To evade EM:** Mask the ECC procedures (Ravi et al., 2020b; Oder et al., 2016), FO transform operations (Ravi et al., 2020b), secure the secret key (Xu et al., 2021), discard ciphertexts (Xu et al., 2021), or randomly split secrets (Xu et al., 2021). • **For avoiding TMP**, no specific countermeasures mentioned for the attacks described in Ravi et al. (2021). • **To block CB:** Store the secret in the time domain instead of the frequency domain (Albrecht et al., 2018).	H	M	H
CRYSTALS-Dilithium	Lattice-based Signature	ML-DSA (Module-Lattice-Based Digital Signature Algorithm)	FIPS 204 (National Institute of Standards and Technology (NIST), 2023c)			✓✓		✓✓	✓✓			• **To defend against FA:** (a) Check the secret and error components of the LWE instances (Ravi et al., 2019b), (b) Employ generic countermeasures such as double computation, verification-after-sign, and add additional randomness (Bruinderink and Pessl, 2018). • **To avoid APA:** Implement masking techniques, including linear secret sharing (Migliore et al., 2019) and use Boolean and arithmetic masking (Marzougui et al., 2022). • **To evade EM:** (a) Re-order operations and embed the vulnerable addition operation within the signing procedure (Ravi et al., 2019a), (b) Utilize it-slicing design for Number Theoretic Transform (NTT) (Singh et al., 2022).	M	M	M
SPHINCS+	Hash-based Signature	SLH-DSA (Stateless Hash-Based Digital Signature Algorithm)	FIPS 205 (National Institute of Standards and Technology (NIST), 2023d)			✓✓		✓✓				• **To defend against FA:** (a) Redundant signature computation (Castelnovi et al., 2018), (b) derive few-time signature indices from public values (Castelnovi et al., 2018), (c) establish inter-layer links in the hyper-tree for fault detection (Castelnovi et al., 2018), (d) use sub-tree recomputation with node swaps and enhanced hash function for fault detection (Genêt et al., 2018), (e) generate and store one-time signatures for efficient reuse (Genêt et al., 2018), (f) verify vulnerable instructions by recomputing on diverse hardware modules (Genêt et al., 2018). • **To prevent APA:** Hide the order of the Mix procedures (Kannwischer et al., 2018).	M	M	M
Falcon	Lattice-based Signature	FN-DSA (FFT NTRU-Lattice-Based Digital Signature Algorithm)	Pending FIPS Certification (FIPS 206) (National Institute of Standards and Technology, 2025)		✓✓	✓✓	✓✓		✓✓			• **To avoid TA:** Enhance random shuffling through the Blind-Vector algorithm (McCarthy et al., 2019), or implement sample discarding (McCarthy et al., 2019). • **To defend against FA:** Duplicate computation of the signature (McCarthy et al., 2019), immediately verify after signing (McCarthy et al., 2019), and perform a zero-check on the sampled vector (McCarthy et al., 2019). • **To prevent SPA:** Effectively reduce the Hamming weight gap. • **To evade EM:** Conceal by maintaining a constant power consumption (Karabulut and Aysu, 2021), or mask by randomizing the intermediate values (Karabulut and Aysu, 2021).	M	M	M
HQC	Code-based KEM/ENC	To Be Standardized (Alagic et al., 2025)	Draft Expected in 2026 (Alagic et al., 2025)		✓✓̸	✓✓̸	✓		✓✓			• **To avoid TA:** Increase the initial byte generation, eliminate additional random data generation calls, and implement constant-time random number generation techniques (Guo et al., 2022). Also, construct a constant-time decoding algorithm (Wafo-Tapa et al., 2020) (no countermeasures are in place for the attacks mentioned in (Guo et al., 2020)). • **To defend against FA:** Employ countermeasures such as default failing, assembly-level instruction duplication, and introducing random delays in the system (Xagawa et al., 2021) (no countermeasures in place for the attacks mentioned in Cayrel et al. (2020)). • **For avoiding TMP**, there are no countermeasures in place for the attacks mentioned in Schamberger et al. (2020). • **To evade EM:** Implement masking using linear secret sharing (Goy et al., 2022).	H	M	H

[a] Attacks and Countermeasures for Quantum-Safe Cryptographic Algorithms Standardized/ Considered for Standardization by NIST. Attacks Are Classified as: d by Cryptanalysis Attacks (CA), Timing Attacks (TA), Fault Attacks (FA), Simple Power Analysis (SPA), Advanced (Correlation/Differential) Power Analysis (APA), Electromagnetic Attacks (EM), Template Attacks (TMP), and Cold-Boot Attacks (CB). (✓: Attack Is Feasible With No Countermeasure in Place; ✓✓: Attack Can Be Mitigated or Prevented by Effective Countermeasures; ✓✓̸: Attack Is Viable, but Countermeasures Are Only Partially Available.)

Table 3
KEM computational performance (milliseconds).

Algorithm	Security level	Key generation	Encapsulation	Decapsulation
ML-KEM-512	L1	0.032	0.032	0.022
ML-KEM-768	L3	0.045	0.046	0.041
ML-KEM-1024	L5	0.052	0.053	0.046
HQC-128	L1	0.120	0.200	0.220
HQC-192	L3	0.220	0.380	0.430
HQC-256	L5	0.450	0.700	0.750

Table 4
KEM artifact dimensions and overhead analysis.

Algorithm	Public key (bytes)	Ciphertext (bytes)	Overhead factor[a]
ML-KEM-512	800	768	12.3×
ML-KEM-768	1184	1088	17.6×
ML-KEM-1024	1568	1568	24.5×
HQC-128	2249	4497	105.3×
HQC-192	4522	9042	212.0×
HQC-256	7245	14,485	339.8×

[a] Relative to ECDH P-256 (64-byte public key).

Table 5
Signature algorithm performance analysis (milliseconds).

Algorithm	Security level	Key generation	Signing	Verification
ML-DSA-44	L1	0.039	0.129	0.040
ML-DSA-65	L3	0.053	0.136	0.056
ML-DSA-87	L5	0.083	0.165	0.082
Falcon-512	L1	12.686	0.525	0.110
Falcon-1024	L5	34.212	1.003	0.199
SLH-DSA-128f	L1	1.155	28.111	3.093
SLH-DSA-128s	L1	66.406	497.387	1.133

Table 6
Signature artifact sizes and efficiency analysis.

Algorithm	Public key (bytes)	Signature (bytes)	Size efficiency
ML-DSA-44	1312	2420	Moderate
ML-DSA-65	1952	3293	Moderate
ML-DSA-87	2592	4595	Moderate
Falcon-512	897	690	Compact
Falcon-1024	1793	1330	Compact
SLH-DSA-128f	32	17,088	Space–Time Trade-off
SLH-DSA-128s	32	7856	Space–Time Trade-off

and frequent verification. SLH-DSA offers parameter flexibility prioritizing speed (f-variants, ∼28 ms signing) or compactness (s-variants, ∼497 ms signing with smaller signatures).

Table 6 presents communication overhead. Falcon achieves optimal size efficiency with sub-1.4 KB signatures. SLH-DSA demonstrates extreme key compactness (32–64 bytes) while producing larger signatures, enabling flexible deployment based on bandwidth constraints.

Infrastructure implications and scaling analysis. PQC algorithms exhibit diverse scaling patterns from L1 to L5: ML-KEM demonstrates linear scaling (1.6× increase in computational cost and artifact size), HQC exhibits polynomial behavior (3.4× computational increase, 3.2× size increase), while signature algorithms follow logarithmic scaling with stable performance ratios. Infrastructure implications are substantial: lattice-based schemes require 3–5× more computational resources while code-based approaches incur 15–30× higher costs, with extreme cases exceeding 1000× classical requirements. Storage requirements increase 10–780× for keys and signatures, necessitating capacity planning for both long-term storage and ephemeral key management. Network bandwidth consumption increases approximately 12–340× for typical cryptographic operations, potentially degrading performance in bandwidth-constrained or latency-sensitive environments (IoT, blockchain consensus networks) (Baseri et al., 2025b).

Table 7 compares classical schemes (ECDH/ECDSA) with NIST-standardized PQC algorithms (ML-KEM/ML-DSA). PQC introduces modest computational overhead: key generation shows 1–2× increases, while encryption/signing and verification exhibit 1–3× slowdowns. However, communication and storage present primary bottlenecks: public keys (800–2592 bytes) are 12–40× larger than classical keys, while signatures/ciphertexts (690–4595 bytes) show 11–72× overhead.

These findings emphasize infrastructure-aware algorithm selection in PQC migration. While computational overhead proves tolerable in high-performance environments, storage and network limitations become critical bottlenecks in constrained devices or decentralized systems, necessitating context-specific scalability assessments balancing security levels, system capabilities, and application requirements.

Algorithm selection framework. Empirical analysis supports application-specific algorithm selection: ML-KEM and ML-DSA provide optimal security-performance-bandwidth balance for general-purpose enterprise and cloud workloads. Falcon offers superior verification performance (<200 ms at L5) despite expensive key generation, ideal for certificate validation and blockchain consensus with infrequent key generation (Baseri et al., 2025b). SLH-DSA enables stateless operation with configurable size-speed trade-offs for high-assurance environments prioritizing security over performance. ML-KEM variants provide optimal compromise for bandwidth-constrained environments (IoT, satellite) with 12–25× overhead versus classical schemes. Lattice-based algorithms (ML-KEM, ML-DSA, Falcon) benefit significantly from Single Instruction, Multiple Data (SIMD) optimization and specialized hardware, attractive for high-performance computing.

Implementation considerations. The substantial increase in cryptographic artifact sizes documented above presents significant implementation challenges when integrating PQC into existing infrastructure. Systems designed with fixed-size buffers for classical cryptographic primitives (typically 64–256 bytes for ECDH/ECDSA) must now accommodate artifacts approximately 12–340× larger. When legacy applications lack proper bounds checking or dynamic memory allocation, these oversized artifacts can trigger buffer-overflow vulnerabilities during key exchange, signature verification, or certificate processing operations. Critical deployment considerations include memory management (2–10× higher RAM requirements necessitating allocation strategies and potential hardware upgrades), cache-performance optimization (larger keys impacting CPU cache efficiency requiring memory access pattern optimization), and backward compatibility through hybrid deployments combining classical and post-quantum algorithms during transitions. In practical deployments, these overheads have direct operational impact beyond bandwidth-constrained edge environments. On high-throughput TLS, SSH, IPsec, and S/MIME servers, expanded cryptographic message sizes increase per-connection memory allocation, while higher signature verification and key encapsulation costs may elevate CPU utilization, reducing concurrent connection capacity or increasing handshake latency under heavy load (Baseri et al., 2024). In multi-tenant cloud environments, these effects can amplify resource contention and scaling costs for services with frequent short-lived connections. Accordingly, PQC migration requires capacity planning that accounts for system-level performance constraints alongside cryptographic robustness. Section 5 analyzes how these implementation constraints and operational overheads may create exploitable attack vectors across infrastructure layers, while Table 9 documents specific exploitation scenarios and countermeasures.

3.2.3. Side-channel vulnerabilities and implementation security

NIST's Post-Quantum Cryptography (PQC) competition endeavors to establish new cryptographic standards that can withstand QC attacks. However, it is essential to recognize that even these algorithms may still be susceptible to other types of attacks, such as side-channel attacks.

A side-channel attack exploits information leakage during the execution of a cryptographic algorithm, including power consumption,

Table 7
Performance comparison with classical cryptography.

Metric	Classical (ECDH/ECDSA)	ML-KEM/ML-DSA	Overhead factor
Key Generation	<0.1 ms	0.03–0.08 ms	1–2×
Encryption/Signing	<0.1 ms	0.03–0.17 ms	1–3×
Verification	<0.1 ms	0.04–0.13 ms	1–2×
Public Key Size	64 bytes	800–2592 bytes	12–40×
Signature/Ciphertext	64 bytes	690–4595 bytes	11–72×

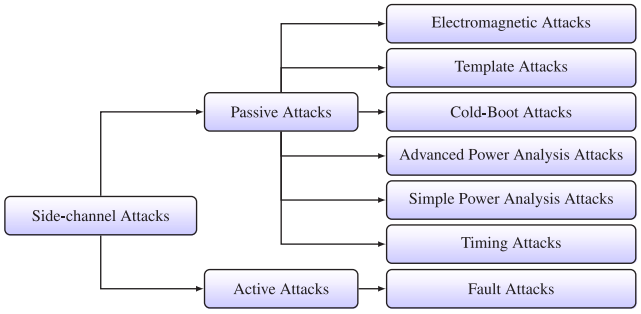


Fig. 7. Taxonomy of attacks for quantum-safe cryptographic algorithms for standardization by NIST.

electromagnetic radiation, or timing information, as shown in Fig. 7. By analyzing this leaked information, an attacker may extract cryptographic secrets, such as private or ephemeral keys. Cryptanalysis attacks, on the other hand, aim to break the encryption or signature schemes by identifying structural weaknesses in the algorithm. Significant instances of side-channel and cryptanalysis attacks on NIST's standardized PQC algorithms have been reported. Notably, the evaluation process is ongoing, and more attacks may emerge in the future. This section examines these attacks, potential countermeasures, and the associated threats to PQC algorithms considered by NIST as quantum-safe cryptographic candidates.

3.2.4. Quantum attack vectors for quantum-safe cryptography

A quantum attacker may persistently attempt to compromise PQC by exploiting various vulnerabilities, including side channels or mathematical analysis. Fig. 7 provides a taxonomy of these vulnerabilities that a quantum attacker could exploit to breach the security of PQC algorithms considered for NIST standardization. In light of this, an attacker can execute side-channel attacks or mathematical analysis attacks. Consequently, we assess the risks associated with each attack on the candidates listed in Table 2. Our evaluation hinges on a qualitative risk assessment, which factors in both likelihood and impact, as visualized in Fig. 6.

3.2.5. Likelihood and impact evaluation

In the likelihood assessment, we examine various factors such as exploitability (via physical access, network, or the internet), available countermeasures (detailed in Table 2), and the criteria provided in Appendix G of NIST-SP 800-30 (National Institute of Standards and Technology (NIST), 2012). We categorize likelihood levels as follows:

1. If a known exploit is available and can be launched from the Internet or a network, the likelihood is considered high when no countermeasures are in place; otherwise, it is assessed as medium.
2. When a known exploit exists, there are no countermeasures, and physical access to the target system is required, the likelihood is rated as medium.
3. In cases where no known exploit exists (or exists along with an effective countermeasure), and a malicious user necessitates administrative or elevated privileges to execute an attack on the target system, the likelihood is determined to be low.

For impact assessment, we refer to the criteria outlined in Appendix H of NIST-SP 800-30 (National Institute of Standards and Technology (NIST), 2012). Threats stemming from quantum attackers have the potential to significantly impact user satisfaction, expose personal data and sensitive information, or damage an organization's reputation. Therefore, the impact level is considered medium.

In our research, we provide a comprehensive analysis of the probability and impact levels, along with the associated risks for various post-migration algorithms. We summarize our findings in Table 2, which includes details on vulnerabilities, potential attacks, countermeasures, and post-quantum threats associated with different post-migration algorithms.

4. Cybersecurity implications of quantum computing on digital infrastructure in pre-migration era

Digital infrastructure, crucial to contemporary society, is structured across nine vital service layers: applications, data, runtime, middleware, operating systems, virtualization, hardware, storage, and networks. These layers work in unison to support global Internet-based services. Our research examines the impact of classical cryptography on these layers, identifying and addressing the threats posed by the advent of quantum computing to both existing and upcoming digital ecosystems, including cloud platforms.

Focusing on what we term the Pre-Migration Phase, it is essential to analyze the current landscape. This phase zeroes in on the cyber impact of quantum computing on infrastructures that currently depend on classical cryptography. With the progression of quantum computing technologies, existing cryptographic standards, which are pivotal for the protection of infrastructure and confidential data, are increasingly at risk of being breached. The delay in adopting quantum-resistant algorithms may expose organizations to a spectrum of quantum-enabled cyberattacks. We explore how advances in quantum computing, including developments in quantum engineering and cryptography, jeopardize classical cryptographic systems, underlining the profound risks these advancements pose on conventional encryption techniques.

4.1. Cyber impact of quantum computing in the pre-migration phase

The advent of quantum computing presents significant security challenges to traditional cryptographic systems. Many current cryptographic algorithms, crucial for safeguarding infrastructure and confidential data, are at risk of being rendered ineffective against quantum computer attacks. Organizations lagging in the adoption of quantum-resistant cryptographic methods are exposed to several potential quantum-enabled threats, including (a) cryptographic breaches, (b) identity theft, (c) financial fraud, and (d) data tampering.

(a) Cryptographic Breaches: Quantum computers have the capability to decrypt standard encryption methodologies such as RSA, Diffie–Hellman, and elliptic curve cryptography. This breach could lead to unauthorized access to critical information, encompassing private communications, passwords, and financial transactions.

(b) Identity Theft: The exploitation of digital signatures by quantum computers can facilitate identity theft. Impersonation of legitimate users or entities could result in unauthorized access to high-security systems or sensitive data, impacting governmental or military operations.

(c) Financial Fraud: The ability of quantum computers to undermine cryptographic security in financial transactions could lead to the misappropriation of funds, unauthorized transfers, or manipulation of financial records.

(d) Data Tampering: Quantum computing might enable alterations in digital data storage by forging the digital signatures used for integrity verification, leading to potential manipulations in medical records, financial statements, or electoral databases.

(e) Cyber Espionage: Nation-states or powerful entities harnessing quantum computing capabilities could engage in advanced espionage activities, targeting confidential and strategic data such as industrial trade secrets or national security information.

To mitigate these vulnerabilities, organizations are advised to proactively transition to quantum-resistant cryptographic technologies. In addition to this cryptographic upgrade, implementing robust access controls and advanced threat detection and response mechanisms are essential strategies for reinforcing cybersecurity in the quantum era.

4.2. Quantum computing attack vectors in the pre-migration phase

Classical systems are fundamentally threatened by quantum computing due to its potential to solve specific mathematical problems that underpin modern cryptography. As a result, a future quantum attacker will possess the ability to break or weaken the most prominent encryption algorithms in use today. Therefore, this section surveys existing security challenges present in the infrastructure before migration to the PQC standard and presents the threats, vulnerabilities, attack vectors, and cryptography types of existing classical systems to evaluate the risk that quantum attackers can cause. This classification can assist security analysts in discovering mitigation measures and identifying security issues more quickly. The threats analyzed below arise from two distinct mechanisms. First, *direct quantum cryptanalytic exploitation*, in which Cryptographically Relevant Quantum Computerss (CRQCs) break classical cryptographic primitives (e.g., Shor's algorithm against RSA/ECC and Grover's reduction of symmetric security margins). Second, *classical exploitation following cryptographic failure*, in which the quantum-induced break of classical primitives allows conventional attack techniques to propagate through shared trust relationships across dependent layers, exposing attack surfaces that would otherwise remain protected. The former requires direct quantum computational capability; the latter leverages classical techniques whose execution is made possible by that quantum break. The attack vectors that can be exploited by quantum machines are discussed below.

4.2.1. Code injection

In the context of quantum computing, attackers might leverage the advanced capabilities of quantum algorithms, such as Shor's algorithm for rapid factorization and Grover's algorithm for efficient database search, to undermine cryptographic security. These quantum algorithms can solve the mathematical problems underpinning modern asymmetric cryptography, enabling attackers to recover private signing keys and forge digital signatures. This vulnerability might allow the insertion of malicious code into software updates or trusted communications. Potential threats include Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020), Insecure Code (OWASP, 2023; Hoffman, 2020), Keyloggers (CrowdStrike, 2024), Viruses (Sharma, 2024; Souppaya and Scarfone, 2013), Trojan Horses (Khan et al., 2017; Souppaya and Scarfone, 2013), Worms (Rawal et al., 2023; Souppaya and Scarfone, 2013), Rootkits (Rawal et al., 2023), Trap Doors (Cheng et al., 2025), Crimeware (Khan et al., 2017), Fileless Malware (CrowdStrike, 2024; Singh and Tripathy, 2025), Data Theft (System, 2021), and System Sabotage (System, 2021). Such attacks could compromise the integrity, confidentiality, and availability of systems and data.

4.2.2. Hypervisor exploitation

In virtualized environments, a significant threat emerges from quantum attackers targeting the hypervisor. Leveraging the advanced capabilities of quantum algorithms, these attackers might successfully bypass established cryptographic defenses to inject malicious code. This intrusion can result in elevated privileges, granting the attackers overarching control over the virtualized framework (Compastié et al., 2020). The consequences are severe, including data interception or manipulation and the unauthorized exfiltration of sensitive information. Furthermore, such breaches can disrupt the functionality of all dependent virtual machines, increasing the risk of a full-scale network breakdown. These vulnerabilities might also be exploited to orchestrate extensive denial-of-service (DoS) attacks, further amplifying their destructive impact.

4.2.3. VM migration and quantum risks

In the process of Virtual Machine (VM) migration, sensitive data is generally encrypted to ensure security while in transit (Oberheide et al., 2008). Nevertheless, the advent of quantum computing presents a significant threat, as it equips attackers with the ability to rapidly decrypt this information. This vulnerability could lead to unauthorized access to confidential data and potentially compromise critical services, including cloud-based operations and data center functionalities.

4.2.4. OS kernel exploitation

Quantum attackers could target the core of a computer's operating system — the kernel. By leveraging algorithms like Shor's and Grover's, they can undermine the cryptographic safeguards protecting kernel integrity and authentication. A successful attack here could lead to a complete system takeover, data leaks, and the introduction of deep system vulnerabilities (Compastié et al., 2020).

4.2.5. Firmware exploitation

Firmware, frequently underestimated in cybersecurity strategies, emerges as a pivotal point of vulnerability against quantum-enabled adversaries. Such attackers could leverage advanced quantum algorithms, notably Shor's and Grover's, to undermine cryptographic safeguards of firmware updates. This vulnerability permits the unauthorized insertion of malicious firmware across a spectrum of devices, extending from IoT appliances to vital infrastructure systems. The resultant risks span espionage, strategic sabotage, and potentially widespread operational disruptions (Cuozzo, 2016).

4.2.6. Symmetric cryptographic key recovery

Leveraging quantum algorithms like Grover's, quantum attackers can significantly reduce the time required to brute-force symmetric cryptographic keys. This poses a grave threat to data confidentiality and can lead to unauthorized access and data breaches (Brassard et al., 1997; Grover, 1996; Grover et al., 2020).

4.2.7. Unauthorized data exfiltration

The advent of quantum computing introduces significant security challenges, especially concerning the safety of encrypted data in virtual machines (VMs) and during data transmission. Quantum algorithms like Shor's and Grover's threaten to undermine traditional cryptographic defenses. These quantum algorithms enable attackers to potentially decrypt and access confidential data within VMs and encrypted communications, bypassing existing encryption techniques. Such breaches have profound implications for individual privacy, corporate confidentiality, and national security. The main tactics for this type of data exfiltration encompass Inter-VM Communication Introspection (Compastié et al., 2020), Decrypting Encrypted Network Traffic (Brassard et al., 1997; Grover, 1996; Shor, 1994), Man-In-The-Middle (MITM) Attacks (Chakraborty and Kalaimannan, 2017), Enhanced Cryptanalysis (Brassard et al., 1997; Grover, 1996; Shor, 1994), Cross-VM Side-channel Cache Attacks utilizing Cryptanalysis (Gruss et al., 2019; Esfahani et al., 2021), and Software-Induced Fault Attacks (Krautter et al., 2018), among others.

4.2.8. Storage server/disk encryption breach

Quantum attackers are capable of breaching storage encryption by illicitly decrypting keys using sophisticated algorithms, such as Shor's and Grover's. This vulnerability endangers not just the confidentiality but also the integrity of stored data. Such intrusions could lead to unauthorized access, alteration, or corruption of vital information, thereby posing a serious threat to the privacy and reliability of the data (Gort et al., 2022; Jangjou and Sohrabi, 2022).

4.2.9. POST request collision

Quantum-enhanced collision attacks causing server overloads represent a *pre-migration* application-layer risk in which attackers overwhelm servers with POST requests that are computationally expensive to process. These attacks exploit application-layer cost asymmetries by concentrating traffic on operations that incur significant processing overhead, potentially leading to server overloads, crashes, or severe performance degradation.

In this setting, attackers may issue large volumes of POST requests with dynamically generated, hash-intensive payloads, increasing the demand for hash computation and validation and heightening the risk of resource exhaustion through algorithmic complexity effects (Smith and Doe, 2019). With access to quantum-era computational capabilities, a future quantum attacker may further amplify this effect by generating and adapting such high-cost payloads at scale, intensifying contention at pre-rejection processing stages.

Although modern systems deploy rate limiting and intrusion prevention mechanisms, POST requests often incur non-trivial processing overhead (e.g., TLS termination, request parsing, and hash-based validation) before such controls are applied. As a result, quantum-enhanced amplification of attacker-side throughput may increase the likelihood of server overloads, particularly in heterogeneous or resource-constrained infrastructure environments.

4.2.10. Quantum-enhanced encryption exploitation

Quantum computing does not provide attackers with fundamentally stronger encryption capabilities; however, it introduces new risks related to key management, recovery, and long-term data accessibility. In pre-migration environments that rely on classical public-key cryptography (e.g., RSA, ECC, and Diffie–Hellman) for key exchange, backup recovery, or key escrow mechanisms, a future quantum adversary may undermine the cryptographic assurances required for secure data restoration. In particular, Shor's algorithm enables CRQCs to efficiently solve integer factorization and discrete logarithm problems, exposing key management infrastructure to retrospective cryptanalysis (harvest now, decrypt later). In this context, ransomware attacks may become more damaging not because attackers encrypt data more effectively (symmetric algorithms such as AES-256 remain quantum-resistant), but because defenders may lose the ability to securely recover encryption keys or verify key provenance from long-term encrypted backups once quantum attacks against classical cryptographic systems become feasible. This shift increases the likelihood of permanent data loss following ransomware incidents, particularly in infrastructures that have not transitioned to quantum-resistant key management mechanisms (Craciun et al., 2019; Kilber et al., 2021).

4.2.11. Access points (AP) breaches

Wi-Fi networks and other network access points secured with legacy protocols like WPA or WEP are at risk. These protocols depend on cryptographic algorithms vulnerable to quantum computing attacks. Specifically, Shor's algorithm can break cryptosystems based on public-key cryptography, which is a foundation of many current security protocols, while Grover's algorithm poses a threat to systems using symmetric key cryptography. Brassard et al.'s contributions in quantum computing further indicate vulnerabilities in these older security protocols (Brassard et al., 1997). Such quantum computing advancements could enable attackers to gain unauthorized access by decrypting encrypted communications, leading to interception or network-based attacks (Lounis and Zulkernine, 2020).

4.3. Security challenges in each infrastructure layer

Based on the outlined attack vectors, it is clear that various threats across different infrastructure layers are emerging, highlighting the need for a comprehensive reassessment. These emerging threats appear distinctly across diverse infrastructure layers, challenging traditional security frameworks and requiring a paradigm shift in both defensive strategies and cryptographic standards. Each layer, from data storage to network transmission, now faces unique vulnerabilities. This situation necessitates a transformation in our security paradigm, which is particularly important in the emerging era of quantum computing. Table 8 presents a comprehensive overview of pre-migration security challenges across various infrastructure layers. In the subsequent sections, we will examine the specific security challenges faced by each infrastructure layer, discuss these challenges in detail, and propose targeted strategies to effectively address them, ensuring a robust defense against these emerging threats.

4.3.1. Application layer

Quantum computing introduces significant risks to Application Layer security, particularly in the areas of data protection and user authentication. The potential of quantum computing to break current encryption methods poses a serious threat to the confidentiality and integrity of application data. Furthermore, authentication processes that rely on existing cryptographic standards are at risk of becoming obsolete. To combat these threats, it is imperative to adopt quantum-resistant algorithms and enhance security protocols to protect application data and prevent unauthorized access or alterations. Vulnerabilities at the Application Layer include weak encryption algorithms, insecure code, and flawed permission models, among others. These vulnerabilities can be exploited through various means, such as brute force attacks (Merhav and Cohen, 2019), exploitation of application software (Compastié et al., 2020; Hoffman, 2020), and assertion repudiation (Grassi et al., 2020; Aldeco Perez, 2012). Potential attack vectors include quantum-enabled decryption of virtual machines (VMs) during migration once classical encryption schemes are weakened, as well as downstream application-layer exploits that become feasible when cryptographic trust mechanisms are compromised. In particular, quantum attacks against classical digital signature schemes may undermine the integrity of software update verification, thereby enabling attackers to inject malicious code that is accepted as trusted. Similarly, the compromise of cryptographic authentication primitives may permit impersonation or unauthorized access once signature- or hash-based verification assumptions no longer hold. To mitigate these risks, the countermeasures suggested involve transitioning to lattice-based or other post-quantum cryptographic standards, strengthening key management practices, adopting secure coding standards, code audits, and implementing post-quantum secure signatures. Additionally, deploying quantum-resistant authentication mechanisms, using hardware security modules for key storage, implementing quantum-safe encryption for data-at-rest and in-transit, and utilizing secure key exchange protocols are recommended. The responsibility for implementing these countermeasures lies with various stakeholders, including software vendors, security architects, application developers, service providers, end-users, service providers, network administrators, infrastructure operators, and network engineers. These stakeholders must collaborate to ensure the Application Layer is fortified against emerging quantum computing threats.

4.3.2. Data layer

Advancements in quantum computing, particularly algorithms like Shor's, Grover's, and those by Brassard et al. are poised to subvert traditional cryptographic paradigms that secure cloud-based data storage. The remarkable speed at which quantum computing can decrypt data poses a significant risk to both the confidentiality and integrity of information stored in the cloud (Jangjou and Sohrabi, 2022; Brassard

et al., 1997; Grover, 1996; Shor, 1994). This emerging threat necessitates a critical reassessment of Key Management Systems (KMS) and underscores the need for cryptographic systems that are resistant to quantum attacks. To address these vulnerabilities, it is imperative to integrate quantum-secure cryptographic primitives to protect against both data breaches and unauthorized alterations of data. The industry must emphasize the urgency of transitioning to PQC to maintain robust security standards. Countermeasures such as transitioning to quantum-resistant encryption standards, promoting cryptographic best practices, and implementing regular integrity checks are vital. Additionally, enhancing update mechanisms with quantum-resistant signatures and securing distribution networks with robust authentication protocols are necessary steps to safeguard against these quantum threats. The responsibility for addressing these challenges is shared among various stakeholders, including industry standards bodies, security architects, data storage providers, security engineers, software vendors, system administrators, cryptographers, and software developers. Each plays a crucial role in ensuring the security of data in the face of quantum computing advancements.

4.3.3. Runtime layer

Quantum threats at the Runtime Layer pose significant risks, such as the exposure of sensitive in-memory data and the compromise of execution environments like Java Virtual Machines and .NET runtimes. These advanced quantum attacks could potentially bypass traditional security measures, enabling unauthorized access to critical data, including personal user information and proprietary algorithms during their execution phase. To counteract these threats, it is essential to implement quantum-resistant encryption standards and employ hybrid encryption models (D et al., 2025). Additionally, transitioning to quantum-resistant encryption protocols and establishing quantum-secure channels are crucial steps. Regular penetration testing, security updates, and security audits are also vital to strengthen the defenses against quantum decryption capabilities. Responsibility involves a clear dichotomy: vendors must engineer quantum-resistant solutions, while security architects and CISOs must manage deployment despite widespread expertise shortages. As detailed in Section 5.5.4, the industry faces acute skills gaps in PQC algorithm design (including lattice-based, code-based, and hash-based schemes), side-channel mitigation techniques, and performance optimization. Consequently, effective migration necessitates capacity building through vendor-led training, specialized third-party auditing, and collaborative initiatives (e.g., the Post-Quantum Cryptography Coalition (Post-Quantum Cryptography Coalition, 2025)) to bridge the research-to-operations divide.

4.3.4. Middleware layer

This layer serves as a crucial communication and data-processing hub for various software services. Its reliance on cryptographic protocols, including those for message queuing, transaction management, and authentication, exposes vulnerabilities to threats such as Cross-VM Side-channel Cache Attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020) utilizing Cryptanalysis techniques (Brassard et al., 1997; Grover, 1996; Shor, 1994). Advances in quantum computing pose a substantial threat, with the potential to decrypt or alter middleware communications, leading to data leaks, unauthorized changes to transactions, and compromised inter-application messages. To mitigate these risks, it is imperative to adopt quantum-resistant cryptographic measures to preserve the integrity and confidentiality of data within the middleware layer. Countermeasures, including implementing stateful hash-based signatures for code integrity, quantum-safe encrypted channels (established using post-quantum key exchange protocols such as ML-KEM-based TLS 1.3 or hybrid classical-PQC schemes (IETF TLS Working Group, 2024; Astrizi and Custodio, 2024)), and quantum-resistant public key infrastructures, can be effective. The responsibility for these implementations lies with both vendors, who must provide the necessary tools and updates, and implementers, who must integrate and maintain these security measures.

4.3.5. Operating system layer

The security implications for the Operating System (OS) Layer in the face of quantum computing are profound, with the potential for kernel-level cryptographic breaches significantly increased (Compastié et al., 2020). Such breaches could result in unauthorized privilege escalation or even total system takeover. The advent of quantum computing emphasizes the urgency for OS security enhancements, particularly in the realm of kernel protection and the adoption of quantum-resistant cryptographic measures. To mitigate these risks, it is imperative to fortify the OS security framework and advance core system defenses to counteract the sophisticated exploitation of OS vulnerabilities by quantum capabilities. Countermeasures include the implementation of Quantum-Secure Signature Schemes and the transition to Quantum-Resistant Certificates to prevent spoofing attacks. For tampering threats, Secure Kernel Development Techniques and Quantum-Resistant Hardware Security Modules (HSMs) are recommended (Crypto4A, 2023). To address information disclosure, integrating hardware-enforced security measures like Trusted Platform Modules (TPMs) and quantum-resistant hardware features, along with regular kernel integrity checks, are vital. For repudiation, enhancing audit and logging mechanisms, utilizing multi-factor authentication, and implementing quantum-resistant cryptographic solutions for signature verification are key. Lastly, to prevent the elevation of privilege, it is crucial to implement hardware-enforced security measures unaffected by quantum decryption capabilities and to conduct regular kernel integrity checks using cryptographic hashes based on post-quantum algorithms. These strategies underscore the collective responsibility of OS developers, system administrators, software vendors, security teams, hardware providers, IT security teams, end-users, cryptographers, security architects, and hardware manufacturers in safeguarding against quantum computing threats.

4.3.6. Virtualization layer

In the Virtualization Layer, exposure to quantum computing threats is multifaceted, with significant concerns including the targeting of hypervisors and unauthorized access to virtualized infrastructures. These vulnerabilities could lead to breaches in VM data confidentiality (e.g., data exfiltration during VM migration (Oberheide et al., 2008)). Advanced quantum computing attacks exploiting cryptographic weaknesses could heavily burden computational and memory resources, resulting in performance degradation and the potential for Denial of Service (DoS) in cloud environments. The phenomenon of VM escape is particularly alarming as it endangers the isolation and integrity of virtualized environments. To mitigate these threats, it is imperative to upgrade cryptographic algorithms resilient to quantum computing techniques, capable of withstanding attacks from Grover's algorithm. This involves transitioning to larger key sizes and adopting post-quantum cryptographic standards. Additionally, enhancing hypervisor security is crucial, achievable through hardware-enforced isolation and the use of intrusion detection systems to monitor anomalous activities. For VMs, implementing strict access control, segregation of critical VMs, and continuous monitoring can help detect and respond to side-channel attacks. Furthermore, secure protocols for VM migration, coupled with end-to-end encryption and regular security audits of migration processes, are essential countermeasures to protect against data exfiltration during VM transfers. These measures require a collaborative effort, with vendors responsible for development and innovation, while implementers must ensure the deployment, configuration, and execution of these security enhancements.

4.3.7. Hardware layer

The advent of quantum computing poses significant security risks to the hardware layer, particularly in cloud server processors and memory components. These risks are primarily due to vulnerabilities that can be exploited by advanced quantum algorithms, such as Shor's and Grover's, leading to breaches in encrypted data. The table highlights specific threats like quantum key-breaking, which can be exploited

through brute force attacks (Lu, 2019), and firmware integrity issues (Cuozzo, 2016) that can lead to unauthorized access and control. To mitigate these threats, it is crucial to implement quantum-resistant cryptographic algorithms and enhance encryption with quantum-safe solutions. Additionally, secure boot processes, firmware signing, and verification, as well as immutable logs and event tracking systems, are recommended countermeasures. The responsibility for these countermeasures primarily lies with vendors, who must ensure the deployment of secure hardware trust anchors to protect data confidentiality and prevent theft.

4.3.8. Storage layer

Quantum computing significantly threatens storage layer security, especially through information disclosure vulnerabilities. Attackers can exploit classical cryptosystems, leading to unauthorized data access and exfiltration (e.g., with quantum-enhanced ransomware making encrypted data nearly impossible to decrypt (Craciun et al., 2019; Kilber et al., 2021)). The rise of quantum algorithms like Shor's and Grover's poses a real risk to data integrity, as they can break current encryption methods. To counteract these threats, it is imperative to adopt quantum-resistant encryption techniques and conduct regular security audits. Enhanced symmetric encryption with increased key sizes and the implementation of cryptographic agility are also vital countermeasures. These strategies are vital for protecting the confidentiality and integrity of data at rest against quantum threats.

4.3.9. Network layer

In the era of quantum computing, the network layer finds itself in a new battlefield of cybersecurity. Existing encryptions, as shown in Fig. 5, and protocols (Quantum-Readiness Working Group (QRWG) of the Canadian Forum for Digital Infrastructure Resilience (CFDIR), 2023), which are standardized and commonly used today, are potential targets for future quantum-powered adversaries. These adversaries could exploit these weaknesses to decrypt network traffic, thereby putting private communications at risk. To defend against these advanced threats, a shift towards cryptographic standards that can withstand quantum attacks is imperative, along with the strengthening of overall network defenses. This involves the adoption of longer cryptographic keys, regular revision of encryption methods, and the implementation of transmission protocols designed to resist quantum-based cryptanalysis. Both vendors and implementers bear the responsibility of incorporating these advanced security measures to integrate cutting-edge security solutions. This integration aims to protect data during its journey through cloud-based services and applications. In essence, fortifying the network layer against quantum threats necessitates an anticipatory stance, involving the development and deployment of encryption capable of defying quantum-level attacks and sophisticated security methodologies. Constant vigilance through routine security checks and the timely application of updates and patches is indispensable to ensure the ongoing confidentiality and integrity of network interactions in the cloud computing landscape.

Table 8 presents a comprehensive overview of pre-migration security challenges across various infrastructure layers. This inclusive table serves as a quick reference guide, encapsulating key insights derived from our analysis. The table encompasses the following components: (1) *Layer*, identifying the specific infrastructure layer examined (e.g., Data Layer, Application Layer, etc.). (2) *STRIDE Threat*, enumerating STRIDE threats pertinent to each vulnerability. (3) *Vulnerabilities*, elucidating specific vulnerabilities associated with each layer susceptible to exploitation by identified threats. (4) *Exploited by*, detailing the threats responsible for the vulnerabilities. (5) *Attack Vector*, furnishing examples of how attackers could exploit these vulnerabilities (e.g., buffer overflow attacks, side-channel attacks, etc.). (6) *Countermeasures*, presenting a compilation of proposed solutions and practices to mitigate identified threats and vulnerabilities. (7) *Responsibility*, indicating whether mitigation falls under the purview of vendors, implementers, users, or a combination thereof.

4.4. Cross-layer attack propagation in pre-migration infrastructure

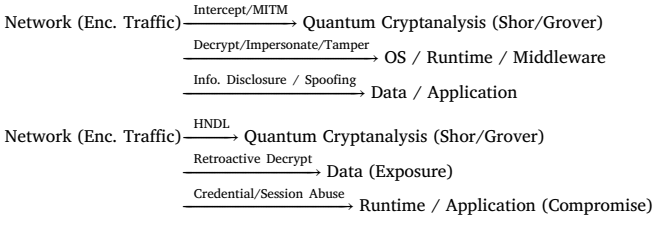
While the preceding analysis examined threats within individual layers, this section investigates how classical-cryptography failures cascade across layers through dependency relationships. In pre-migration systems, a break of core primitives can trigger *systematic* propagation without requiring per-layer exploitation: once RSA/ECC-based mechanisms are broken or symmetric key sizes rendered insufficient, dependent services lose their cryptographic guarantees through shared trust relationships. This differs from post-migration risk, where failures are probabilistic, multi-stage, and dominated by implementation weaknesses. The driver is *mathematical collapse*: Shor's algorithm reduces RSA/ECC to polynomial time (Shor, 1994), while Grover's algorithm halves the effective bit strength of symmetric keys, making undersized constructions inadequate. Grounded in Table 8, five main cross-layer cascades arise: (1) *Secure-Channel Cryptanalysis* via intercept, MITM, and HNDL-based retroactive decryption; (2) *Update-Signature Forgery* via forged signed updates and code injection; (3) *Firmware Integrity Compromise* via firmware authenticity failure and hypervisor/VM escape; (4) *Storage Cryptanalysis and Key Recovery* via classical key recovery and bulk decryption; and (5) *VM Mobility/Migration Exposure* via quantum cryptanalysis of migration channels. Collectively, these illustrate how a break in shared trust anchors (secure channels; signed updates/firmware; encrypted storage) can propagate systematically across the entire nine-layer stack.

4.4.1. Secure-channel cryptanalysis cascade

Table 8 highlights network-layer exposure through Tampering and Information Disclosure (MITM, packet interception over encrypted traffic), with corresponding risks in Runtime/Middleware/OS and Data/Application once cryptographic protection fails. Two operationally distinct sub-mechanisms arise: (a) active channel compromise, exploiting real-time CRQC decryption capability, and (b) Harvest-Now, Decrypt-Later (HNDL), deferring decryption until CRQC becomes available.

(a) Active channel compromise. An attacker intercepts an encrypted channel handshake (TLS, SSH, IPsec/IKE, VPN tunnels) and applies CRQC-enabled cryptanalysis to break the classical public-key authentication or key-agreement mechanism (RSA/ECDSA signatures or DH/ECDH-based exchange), enabling MITM decryption. Modified HTTP responses, API calls, tunneled traffic, or management-plane communication are re-encrypted and forwarded; injected payloads execute at the Application layer while recovered tokens and credentials enable impersonation, privileged API access, and lateral movement into Middleware or host services.

(b) HNDL. The *temporal* arm archives ciphertext today (TLS sessions, SSH tunnels, VPN flows, and encrypted API traffic) for retroactive decryption (Mosca, 2018), creating a cumulative exposure surface that grows with each day of delayed migration. Archived sessions may reveal historical data, long-lived credentials, API keys, or embedded tokens, enabling impersonation, middleware access, and pivoting to currently operational systems via recovered authentication material.



4.4.2. Update-signature forgery cascade

Table 8 identifies code injection via *forged signatures during software updates* across the stack. CRQC-enabled cryptanalysis (Shor's algorithm against RSA/ECC-based code-signing mechanisms) allows attackers to

Table 8
Pre-migration security challenges for each infrastructure layer.

Layer	STRIDE threat	Vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
Application	Spoofing	Weak Encryption Algorithms	VM Mobility (Compastié et al., 2020; Huang et al., 2012), Brute Force Attack (Merhav and Cohen, 2019)	Quantum algorithms like Shor's and Grover's could be used to decrypt VMs during migration or brute force cryptographic keys	Transition to lattice-based or other NIST-approved post-quantum cryptographic standards, strengthen key management practices	Software vendors, Security architects
	Tampering	Insecure Code	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection with forged signature during software update, exploiting vulnerabilities in the update process	Adoption of secure coding standards, Code audits, Implement post-quantum secure signatures	Application developers, Security teams
	Repudiation	Weak Authentication Mechanisms	Assertion Repudiation including Hash Attack, Data Driven Attacks and Imperson-ation/Masquerade Attack (Grassi et al., 2020; Aldeco Perez, 2012)	Quantum algorithms can forge signatures or hashes used in authentication	Deployment of quantum-resistant authentication mechanisms, Use of hardware security modules for key storage	Service providers, End-users
	Info. Disclosure	Insufficient Data Protection	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020), Inter-VM Communication Introspection (Compastié et al., 2020), Cryptanalysis Attacks (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Code injection during updates, MITM attacks on VMs, cryptographic breaches	Implement quantum-safe encryption for data-at-rest and in-transit, utilize secure key exchange protocols	Service providers, Network administrators
	Denial of Service	Vulnerable Network Interfaces	POST Request Collision (Smith and Doe, 2019)	Quantum-enhanced collision attacks causing server overloads	Load balancing, Deployment of quantum-resistant hash functions, real-time traffic monitoring	Infrastructure operators, Network engineers
	Elevation of Privilege	Flawed Permission Models	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection leading to unauthorized privilege escalation	Enforce robust access control policies, Regular privilege audits, Zero trust architecture implementation	System administrators, IT departments
Data	Spoofing	Susceptible encryption under quantum attacks	Brute Force Attack (Gort et al., 2022)	Quantum algorithms reducing symmetric key security	Transition to quantum-resistant encryption standards; promote best practices in cryptographic best practices.	Industry Standards Bodies, Security Architects.
	Tampering	Modification of encrypted data storage	Data Modification Attack (Jangjou and Sohrabi, 2022)	Storage servers encryption breach by quantum attacker	Regular integrity checks; integrate quantum-resistant cryptographic hashes and encryption.	Data Storage Providers, Security Engineers.
	Info. Disclosure	Insecure software update mechanisms	Injecting Keylogger (CrowdStrike, 2024)	Code injection via quantum-forged signatures	Enhance update mechanisms with quantum-resistant signatures; secure distribution networks with robust authentication protocols.	Software Vendors, System Administrators.
		Classical cryptosystems breakable by quantum algorithms	Cryptanalysis Attacks (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Quantum decryption via Shor's, Grover's and Brassard et al.'s algorithms	Early adoption of NIST-approved post-quantum algorithms; update encryption protocols.	Cryptographers, Software Developers
	Denial of Service	Quantum-enhanced collision attacks on hash functions	Hash DoS (Masdari and Jalali, 2016)	POST request collision using quantum computing	Implement rate limiting; adopt post-quantum hash functions; use advanced intrusion detection systems.	Network Administrators, Cybersecurity Teams
	Elevation of Privilege	Weak cryptographic access controls	Data Theft (System, 2021)	Quantum algorithms enabling unauthorized access	Employ multi-factor authentication; establish strict access control policies using post-quantum secure methods.	IT Security Policy Makers, End-Users
		Compromised cryptographic mechanisms	Unauthorized Data Access (Jangjou and Sohrabi, 2022)	Quantum cryptanalysis for access control breach	Upgrade to quantum-resistant access control systems; continuous security training for users.	Access Control Vendors, Corporate Training Departments

(*continued on next page*)

Table 8 (*continued*).

Layer	STRIDE threat	Vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
Runtime	Spoofing	Susceptibility to quantum decryption	VM Mobility (Compastié et al., 2020; Huang et al., 2012)	Decryption of an encrypted VM while moving it from one physical host to another	Implement quantum-resistant encryption standards; Employ hybrid encryption models	Vendors, Security Architects
		Weak encryption methods	Brute Force Attack (Tirado et al., 2018)	Via Grover's algorithm to brute-force symmetric cryptographic keys	Early adoption of post-quantum cryptographic algorithms; Regular security updates	Implementers, IT Security Teams
	Tampering	Insecure update mechanisms	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection with forged signature during software update	Deployment of quantum-safe digital signature schemes; Use of secure and authenticated update channels	Vendors, Developers
	Info. Disclosure	Inadequate encryption	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection leading to data leaks	Transition to quantum-resistant encryption protocols; Use of secure communication channels	Vendors, Network Administrators
		Vulnerable VM communication	Inter-VM Communication Introspection (Compastié et al., 2020)	MITM attacks exploiting quantum decryption capabilities	Establishment of quantum-secure channels; Regular penetration testing	Implementers, Security Teams
		Weak cryptosystems	Cryptanalysis Attacks (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Data exfiltration using quantum algorithms to break encryption	Upgrading to quantum-resistant algorithms; Conducting regular security audits	Vendors, CISOs
	Elevation of Privilege	Compromised update integrity	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Forged updates granting elevated access.	Secure bootstrapping and update processes; Multi-factor authentication systems	Implementers, Users, IT Administrators
Middleware	Spoofing	VM Identity Forgery	VM Mobility (Compastié et al., 2020; Huang et al., 2012)	Decryption of an encrypted VM during migration	Implementation of quantum-resistant cryptographic protocols for secure VM migration	Vendors to provide tools; Implementers to deploy
		Cryptographic Key Cracking	Brute Force Attack (Bates et al., 2012)	Quantum algorithms like Grover's	Adoption of quantum-resistant public key infrastructures (PKIs)	Vendors to update products; Implementers to integrate
	Tampering	Code	C&C Channel Exploitation (Compastié et al., 2020)	MITM attacks with forged signatures	Usage of stateful hash-based signatures for code and communications	Vendors to develop; Implementers to apply
		Hypervisor Integrity	Hypervisor Exploitation (Compastié et al., 2020)	Direct attacks on the hypervisor	Deployment of verified boot processes and attestation mechanisms	Vendors to innovate; Implementers to maintain
	Repudiation	Non-repudiation Failure	C&C Channel Exploitation (Compastié et al., 2020)	Forging signatures on C&C channels	Immutable logging mechanisms with quantum-resistant signatures	Implementers to enforce; Users to monitor
	Info. Disclosure	Inter-VM Communication Eavesdropping	Cross-VM Side-channel Cache Attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020)	Cryptanalysis with quantum algorithms	Isolation enhancements with quantum-safe encrypted channels	Vendors to enhance; Implementers to configure
	Denial of Service	Hypervisor Overload	Hypervisor Exploitation (Compastié et al., 2020)	Quantum DoS attacks via hypervisor exploitation	Redundant, distributed architectures and quantum-aware rate limiting	Vendors to design; Implementers to orchestrate
	Elevation of Privilege	Unauthorized Access	Hypervisor Exploitation (Compastié et al., 2020)	Gaining higher privileges via hypervisor exploitation	Multifactor authentication systems with quantum-resistant protocols	Implementers to enforce; Users to adopt
OS	Spoofing	Quantum Algorithm Exploits	Injecting Virus (Sharma, 2024)	Code injection with forged signature during software update	Implementation of Quantum-Secure Signature Schemes, Transition to Quantum-Resistant Certificates	OS Developers, System Administrators
		Cryptographic Breakdown	Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Code injection with forged signature during software update	Adoption of PQC Libraries, Secure Software Development Lifecycle (SDLC) Practices	Software Vendors, Security Teams
			Injecting Trap Door (Cheng et al., 2025)	Malicious code, containing a trap door, which is injected into software with a forged signature during software update.	Enhanced Software Supply Chain Security, Continuous Integrity Monitoring	System Administrators, Security Auditors
		Symmetric Key Exposure	Brute Force Attack (Tirado et al., 2018)	Exploitation of symmetric key cryptography via Grover's algorithm	Migration to Larger Symmetric Key Sizes, Quantum Key Distribution (QKD) Integration	Cryptographers, Security Architects

(*continued on next page*)

Table 8 (*continued*).

Layer	STRIDE threat	Vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
OS	Tampering	Kernel Integrity Compromise	OS Kernel Exploitation (Compastié et al., 2020)	Exploitation through breaking hardware security modules (HSM) using quantum algorithms.	Secure Kernel Development Techniques, Quantum-Resistant HSMs	OS Vendors, Hardware Providers
		Malware Insertion	Injecting Virus (Sharma, 2024)	Code injection with forged signature during software update	Advanced Endpoint Detection and Response (EDR) Tools, Regular Security Updates	IT Security Teams, End Users
			Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Code injection with forged signature during software update.	Mandatory Access Control (MAC), Executable Space Protection	System Administrators, Application Developers
			Injecting Trojan Horse (Khan et al., 2017)	Code injection with forged signature during software update.	Real-Time Threat Intelligence, Heuristic Analysis Techniques	IT Security Analysts, End Users
			Crimeware (Khan et al., 2017)	Code injection with forged signature during software update.	Comprehensive Cybersecurity Training, Security Operations Center (SOC)	IT Staff, SOC Analysts
	Repudiation	Non-repudiation Failure	Injecting Virus (Sharma, 2024), Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Code injection with forged signature during software update	Implement quantum-resistant cryptographic solutions for signature verification; Enhance audit and logging mechanisms to track software changes and updates; Utilize multi-factor authentication to reinforce non-repudiation.	Vendors to provide quantum-safe signature schemes; Implementers to ensure proper configuration and monitoring; Users to adhere to security practices.
	Info. Disclosure	Quantum Cryptanalytic Attacks	OS Kernel Exploitation (Compastié et al., 2020)	Exploited through breaking HSM via Shor's, Grover's, and Brassard et al.'s algorithms by quantum attackers.	Implementing hardware-enforced security measures that remain unaffected by quantum decryption capabilities, such as Trusted Platform Modules (TPMs) for secure boot; Integrating quantum-resistant hardware security features; Regular kernel integrity checks using cryptographic hashes based on post-quantum algorithms.	OS Vendors, Hardware Manufacturers
		Weaknesses in Encrypted Data Handling	Inter-VM Communication Introspection (Compastié et al., 2020)	Man-in-the-Middle (MITM) and data exfiltration of encrypted content by quantum-capable adversaries.	Transition to Quantum-Secure Channels, Real-time monitoring and anomaly detection systems.	Implementers to enforce new protocols; Security teams to monitor and respond to incidents.
		Malicious Code Insertion	Injecting Insecure Code (OWASP, 2023; Hoffman, 2020), Keylogger (CrowdStrike, 2024), Trojan Horse (CrowdStrike, 2024), Fileless Malware (CrowdStrike, 2024; Singh and Tripathy, 2025), Worm (Rawal et al., 2023)	Code injection with forged signature during software update	Implement quantum-resistant cryptographic algorithms for code signing, advanced threat detection systems with heuristic analysis, secure software update mechanisms, employ SSDLC, and conduct regular developer training on quantum-related threats.	Developers adhere to SSDLC with quantum-resistant measures, vendors provide quantum-resistant code signing tools, and users verify signatures with quantum-resistant verification methods.
		Cross-VM Side-channel Attacks	Side-channel Cache Attacks in combination with cryptanalysis (Gruss et al., 2019; Esfahani et al., 2021)	Data exfiltration via Shor's, Grover's, and Brassard et al.'s algorithms, leveraging side-channel vulnerabilities.	Strengthening VM isolation, employing noise and blinding techniques in cryptographic processes, and regular security audits for virtualized environments.	Virtualization Platform Providers, System Administrators.
	Denial of Service	Malicious code inserted via signature forgery	Injecting Virus (Sharma, 2024)	Code injection with forged signature during software updates	Quantum-secure code signing protocols; Advanced heuristic-based antivirus solutions; Secure software update mechanisms	Application Developers, End Users
		Quantum-Enhanced Encryption	Ransomware (Craciun et al., 2019; Kilber et al., 2021)	Triggered when more powerful quantum attackers have more power to encrypt contents in a more complex way.	Backup systems, Anti-ransomware solutions	Users, Implementers

(*continued on next page*)

Table 8 (*continued*).

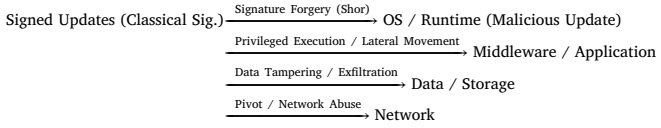
Layer	STRIDE threat	Vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
	Elevation of Privilege	Kernel integrity compromised by quantum algorithms	OS Kernel Exploitation (Compastié et al., 2020)	Bypassing hardware security modules (HSMs) via quantum algorithms (Shor's, Grover's and Brassard et al.'s algorithms)	Implementing hardware-enforced security measures that remain unaffected by quantum decryption capabilities, such as Trusted Platform Modules (TPMs) for secure boot; Integrating quantum-resistant hardware security features; Regular kernel integrity checks using cryptographic hashes based on post-quantum algorithms.	OS Vendors, Hardware Manufacturers
		Malicious code inserted via signature forgery	Injecting Virus (Sharma, 2024), Insecure Code (OWASP, 2023; Hoffman, 2020), Trap Door (Cheng et al., 2025)	Code injection with forged signature during software updates	Implement quantum-resistant cryptographic algorithms for code signing, advanced threat detection systems with heuristic analysis, secure software update mechanisms, employ SSDLC and conduct regular developer training on quantum-related threats.	Developers adhere to SSDLC with quantum-resistant measures, vendors provide quantum-resistant code signing tools, and users verify signatures with quantum-resistant verification methods.
Virtualization	Spoofing	Cryptographic key vulnerability	Brute Force Attack (Grover et al., 2020)	Via Grover's algorithm to brute-force a key	Upgrade to cryptographic algorithms with larger key sizes, and encourage early adoption of post-quantum cryptographic standards.	Vendors to develop, Implementers to deploy
	Info. Disclosure	Hypervisor vulnerabilities	Side-channel cache attacks (Irazoqui et al., 2015) and cryptanalysis attack (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Data exfiltration via quantum algorithms	Introduce hardware-enforced isolation, leverage intrusion detection systems to monitor anomalous activities, and apply quantum-resistant algorithms for sensitive operations.	Vendors to innovate, Implementers to install and maintain
		Cross-VM attacks	Cross-VM Side-channel Cache Attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020) in combination with Cryptanalysis (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Data exfiltration via quantum algorithms	Segregate critical VMs, implement strict access control and continuous monitoring to detect and respond to side-channel attacks.	Implementers to configure, IT security to oversee
		VM migration risks	VM Migration (Oberheide et al., 2008)	Data exfiltration during VM migration	Employ end-to-end encryption for VM migration data, utilize secure migration protocols, and conduct regular security audits on migration processes.	IT security to audit, Implementers to execute
Hardware	Spoofing	Quantum key-breaking	Brute Force Attack (Lu, 2019)	Exploiting quantum algorithms to break cryptographic keys	Implement quantum-resistant cryptographic algorithms	Vendors
	Tampering	Firmware integrity	Firmware Exploitation (Cuozzo, 2016)	Forged firmware updates with malicious code	Secure boot, firmware signing, and verification	Vendors
	Repudiation	Traceability of changes	Firmware Exploitation (Cuozzo, 2016)	Unauthorized firmware updates to deny operations	Immutable logs and event tracking systems	Implementers
	Info. Disclosure	Encrypted data breach	Cryptanalysis Attacks (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Advanced cryptanalysis leveraging quantum algorithms	Enhance encryption with quantum-safe solutions	Vendors
	Elevation of Privilege	Unauthorized access control	Firmware Exploitation (Cuozzo, 2016)	Injecting malicious firmware to gain higher privileges	Access control mechanisms, multi-factor authentication	Users
Storage	Spoofing	Symmetric cryptographic key vulnerability	Brute Force Attack (Shen et al., 2020)	Via Grover's algorithm to brute-force an n-bit symmetric key	Enhanced symmetric encryption with longer key sizes, Implementation of cryptographic agility	Vendors, Implementers
	Info. Disclosure	Vulnerabilities in classic cryptosystems	Cryptanalysis Attacks (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Data exfiltration via breaking classic cryptosystems	Early adoption of quantum-resistant encryption methods, Regular security audits	Vendors, Implementers
	Denial of Service	Susceptibility to quantum-complex encryption	Ransomware (Craciun et al., 2019; Kilber et al., 2021)	Quantum-enhanced ransomware attacks	Enhanced intrusion detection systems, Frequent and secure data backups, Ransomware-specific defenses	Users, Implementers

(*continued on next page*)

Table 8 (*continued*).

Layer	STRIDE threat	Vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
	Elevation of Privilege	Forgery of digital signatures	Data Theft (System, 2021)	Code injection with forged signatures during updates	Implementation of transitional digital signature standards, Multi-factor authentication for update processes	Vendors, Implementers
Network	Spoofing	Vulnerable symmetric encryption	Brute Force Attack (Idhom et al., 2020)	Via Grover's algorithm to brute-force a symmetric cryptographic key	Early implementation of longer key lengths; Monitoring and updating cryptographic standards based on the latest quantum computing advancements	Vendors and Implementers
	Tampering	Insecure Wi-Fi Protocols	Wi-fi Wardriving (Lounis and Zulkernine, 2020)	Via Shor's and Grover's algorithms to break classic cryptosystems	Upgrade to WPA3 with quantum-resistant features; Regular updates and patches to Wi-Fi protocols	Vendors and Users
		Physical AP vulnerabilities	Physical AP Attacks (Chakraborty and Kalaimannan, 2017; Lounis and Zulkernine, 2020)	Via quantum algorithms to compromise APs	Regular security assessments of APs	Implementers and Users
		Susceptible cryptographic protocols	Cryptanalysis Attacks (Zhang et al., 2018)	Via quantum algorithms to break cryptosystems	Development and integration of quantum-resistant cryptographic protocols; Regular testing and validation	Vendors
		Signature forgery risk	MITM (Chakraborty and Kalaimannan, 2017)	Forged signature on messages by quantum attacker	Digital signature standards resistant to quantum attacks	Vendors
	Repudiation	Lack of non-repudiation controls	Message Repudiation Attack (Aldeco Perez, 2012)	Forged signatures using quantum algorithms	Integration of robust, quantum-resistant non-repudiation mechanisms in messaging and communication systems	Vendors and Implementers
	Info. Disclosure	Encryption weaknesses	Packet Sniffing (Over Encrypted Packets) (Shibly and De Soto, 2020)	Data exfiltration by breaking cryptosystems with quantum computing	Adoption of advanced, quantum-resistant encryption techniques; Continuous monitoring and updating of encryption protocols	Vendors and Users
		Vulnerable data transmission	MITM (Aldeco Perez, 2012)	Data interception using advanced quantum algorithms	Implementation of secure, quantum-resistant transmission protocols; Regular security assessments of transmission channels	Vendors
		Weak cryptanalysis defenses	Cryptanalysis Attacks (Brassard et al., 1997; Grover, 1996; Shor, 1994)	Cryptanalysis Attacks & Data theft through advanced quantum cryptanalysis	Conducting regular security audits; Updating cryptographic algorithms to quantum-resistant versions	Vendors and Implementers
	Elevation of Privilege	Compromised update mechanisms	System Sabotage (System, 2021)	Code injection with forged signature during software update	Establishing secure and authenticated software update protocols; Continuous monitoring of software update processes for anomalies	Vendors and Implementers

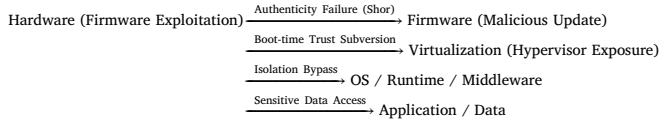
forge update authenticity; a single malicious update at the OS or Runtime layer executes with elevated privileges and propagates through shared trust relationships, enabling persistence, lateral movement into Middleware, and tampering or exfiltration of Data-layer assets and Storage backups, without requiring independent authentication bypass at each layer.



This directly supports the malware insertion vectors in Table 8 for the OS layer (viruses, trojan horses, fileless malware, worms, trap doors, crimeware), delivered through otherwise authenticated update channels.

4.4.3. Firmware integrity compromise cascade

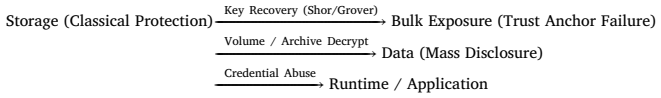
Table 8 identifies firmware integrity failures at the Hardware layer (Tampering: forged firmware updates; Information Disclosure: encrypted data breach) and virtualization-layer exposure via hypervisor dependencies. CRQC-enabled attacks against RSA/ECC-based firmware authenticity mechanisms may enable forged firmware updates; the resulting malicious firmware operates below OS visibility, undermining secure boot and virtualization trust assumptions, facilitating isolation bypass, hypervisor compromise, and upward propagation into OS, Runtime, and Middleware layers. This exposes sensitive application data or in-memory cryptographic material and establishes persistence that survives OS reinstallation.



4.4.4. Storage cryptanalysis and key recovery cascade

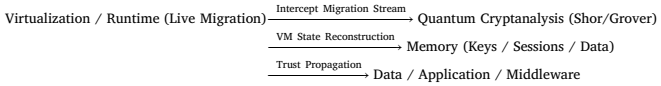
Table 8 attributes Storage-layer exposure to classical cryptosystem weaknesses (Information Disclosure via cryptanalysis) and reduced symmetric security margins under Grover's algorithm. In deployments using hierarchical key management, CRQC-enabled attacks against RSA/ECC-based wrapping mechanisms, or symmetric KEKs of insufficient size, may render bulk data decryptable once protecting key material is recovered. The failure is often *atomic* at the trust-anchor level: entire storage volumes, database tablespaces, or backup archives

become simultaneously exposed, and recovered credentials may further enable runtime access or compromise of active systems.



4.4.5. VM mobility/migration exposure cascade

Table 8 identifies VM mobility and migration as a recurring exposure surface. Live migration transmits VM state (memory pages, session data, and cryptographic material) over encrypted management channels. CRQC-enabled cryptanalysis against classical RSA/ECC-based protection can recover session secrets and enable offline reconstruction of VM memory state, exposing embedded credentials and keys. Because migration traffic often traverses dedicated management planes with distinct monitoring controls, compromise may propagate into trusted peer systems and orchestration infrastructure.



Collectively, these cascades form a *quantum cliff-edge* in the pre-migration stack. Unlike post-migration attacks (probabilistic, multi-stage, and dependent on chaining implementation flaws), pre-migration failures are mathematically driven and structurally coupled. Once classical secure channels, signed updates/firmware, or storage protection are broken, dependent layers lose their cryptographic guarantees and become systematically exposed through shared trust relationships. VM migration represents a critical exposure surface that accelerates this propagation when migration channel confidentiality depends on breakable classical mechanisms.

5. Cybersecurity implications of quantum computing on digital infrastructure in post-migration era

Building upon the insights discussed in the preceding section, this part of our study shifts focus towards the Post-Migration Phase, where quantum-resistant cryptography begins to reshape the digital security landscape. We thoroughly examine the effects of post-quantum cryptographic systems on the infrastructural layers, aiming to pinpoint and mitigate the potential threats that quantum computing could introduce to both existing and future digital ecosystems, including cloud computing platforms. In this advanced stage of transition towards a quantum-safe infrastructure, we delve into the specific Cyber Impacts that arise when infrastructures shift to quantum-resistant algorithms, like those in post-quantum cryptographic systems. Our analysis is particularly critical at this moment, as moving to these advanced algorithms is vital to defend against the enhanced computational capabilities of quantum computers, which are predominantly aimed at breaking encryption. However, it is crucial to understand that the impact of quantum computing is not limited to cryptographic threats alone. We then navigate through the landscape of quantum computing threats in the Post-Migration Era, scrutinizing the effectiveness, adaptability, and resilience of post-quantum cryptographic systems. Despite their robust design against quantum and classical computational threats, vulnerabilities remain. Quantum attackers are continually probing for weaknesses, leveraging sophisticated quantum technologies in techniques like cryptanalysis, side-channel attacks, code injection, and more. This section emphasizes the need for ongoing vigilance and regular reassessment of vulnerabilities within infrastructures, ensuring consistent security even after the adoption of quantum-resistant cryptographic solutions.

5.1. Cyber impact of quantum computing in the post-migration phase

The shift to quantum-safe cryptographic algorithms marks a pivotal step in safeguarding organizations from the advanced computational power of quantum computers. This transition, primarily aimed at preventing quantum computers' ability to break traditional encryption, also brings to the fore new cybersecurity challenges that go beyond mere decryption threats. Key among these are issues related to (a) increased key sizes and network traffic, (b) implementation complexity, (c) performance overheads, and (d) adapting network security devices.

- **(a) Increased Key Sizes and Network Traffic:** The adoption of PQC typically results in larger cryptographic keys and ciphertexts. This increase can lead to more fragmented network traffic, posing challenges for systems with limited capabilities in managing and reassembling fragmented data. Consequently, the attack surface for threats associated with data fragmentation broadens.
- **(b) Implementation Complexity:** PQC integration adds layers of complexity to existing systems, potentially birthing new security vulnerabilities. These vulnerabilities are particularly pronounced in the management of fragmented network packets, necessitating proactive measures for robust security maintenance in the quantum era.
- **(c) Performance Overheads:** PQC algorithms can degrade performance, especially in high-traffic environments. This degradation can strain network infrastructure, heightening susceptibility to attacks exploiting fragmentation vulnerabilities.
- **(d) Adapting Network Security Devices:** The increasing volume of PQC-encrypted traffic necessitates the advancement of network security appliances, including firewalls and intrusion detection systems, to effectively process and inspect this new type of traffic. During the transition phase, there may be temporary vulnerabilities that could exacerbate the effects of current fragmentation-based attacks and potentially lead to the emergence of new cyber threats.

While the shift to PQC is essential in mitigating threats from quantum computing, it introduces a range of new cybersecurity challenges. These include performance issues in high-traffic environments, increased vulnerability to DoS attacks due to heightened computational demands, and potential weaknesses in cryptographic protocols exploitable by quantum techniques. Moreover, quantum-safe cryptography, though resistant to decryption, does not inherently safeguard against social engineering attacks like phishing. The transition to PQC also raises risks such as buffer overflow attacks in older systems not designed for larger cryptographic keys and ciphertexts, which could lead to advanced malware injections or the forging of cryptographic signatures. Therefore, adopting PQC is a critical but not all-encompassing solution. A comprehensive security strategy is imperative, one that fortifies systems against a variety of sophisticated threats that could emerge in the post-quantum landscape. The following section will explore various attack vectors that might arise in infrastructures transitioning to PQC.

5.2. Quantum computing attack vectors in the post-migration phase

Post-quantum cryptographic systems utilize quantum-resistant algorithms, safeguarding against attacks from both quantum and classical computers. These systems are designed to integrate seamlessly with existing communication protocols and networks after migration. Despite this, the cryptographic community continues rigorous security evaluation to stress-test PQC standards. Researchers and adversarial modelers assess potential vulnerabilities using theoretical quantum frameworks, classical simulations, and implementation analysis, examining quantum algorithmic weaknesses, side-channel attacks (detailed in Table 2), cryptanalysis, and code injection to anticipate future threat vectors. This section reviews security challenges present in infrastructure following the transition to PQC standards (summarized in Table 9) and

explores the associated threats, vulnerabilities, attack vectors, and cryptography types. This information aims to aid security analysts in quickly identifying and mitigating security issues. Post-migration attack vectors fall into two categories: *PQC-targeted attacks*, which directly target the mathematical structure or physical implementation of PQC algorithms (e.g., structural cryptanalysis, side-channel and fault exploitation as summarized in Table 2), and *deployment-amplified attacks*, including buffer overflows, code injection, and resource exhaustion (i.e., classical threats that become significantly more dangerous in PQC deployments due to larger cryptographic artifacts, increased computational overhead, and implementation immaturity). Below, we discuss potential attack vectors in systems secured by PQC.

5.2.1. Side-channels

Quantum attackers are capable of compromising cryptographic algorithms by exploiting implementation weaknesses. Side-channel attacks may stem from the co-location of virtual machines (VMs) or hosts, shared network infrastructure, and other scenarios involving shared resources. Key vulnerabilities include OS Kernel Exploitation (Compastié et al., 2020), Hypervisor Exploitation (Compastié et al., 2020), VM Monitoring (Compastié et al., 2020), Inter-VM Communication Inspection (Compastié et al., 2020), as well as Cross-VM (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020), Micro-architectural Cache (Shen et al., 2021), and Fault attacks (Blömer et al., 2014). These vulnerabilities can lead to the disclosure of sensitive information, data tampering, and challenges in ensuring non-repudiation.

5.2.2. Code injection

Code injection, while a classical attack, becomes more exploitable in PQC deployments due to significantly larger cryptographic artifacts (public keys 12–40× larger, signatures/ciphertexts 11–72× larger as discussed in Section 3.2.2). Legacy systems with fixed buffers designed for classical keys (64–256 bytes) encounter overflow conditions when processing multi-kilobyte PQC artifacts. Adversaries exploiting these memory safety vulnerabilities can inject malicious code into Application Software (Compastié et al., 2020; Hoffman, 2020), Hypervisors (Compastié et al., 2020), and Firmware (Cuozzo, 2016), enabling Keyloggers (CrowdStrike, 2024), Viruses (Sharma, 2024), Rootkits (Rawal et al., 2023), VM Escapes (Compastié et al., 2020), and System Sabotage (System, 2021). Control-flow hijacking via buffer overflows emphasizes the need for memory-safe programming, bounds checking, and proper validation of oversized PQC keys. These attacks require only classical computing resources.

5.2.3. File system exploitation

In the context of PQC and the advancements in quantum computing, file systems and hard drives face heightened risks. These vulnerabilities often stem from network daemons, email clients, or web browsers, which can serve as paths for quantum attackers. Utilizing buffer overflow vulnerabilities (Masdari and Jalali, 2016; Compastié et al., 2020), attackers are able to inject wiper malware into systems (CrowdStrike, 2024). This form of malware is particularly insidious, targeting crucial data for deletion or corruption, leading to potentially severe and irreparable system disruptions.

5.2.4. OS kernel exploitation

This method typically involves gaining system control through methods like side-channel attacks, pose risks such as tampering, information disclosure, privilege elevation, rootkit installation, memory corruption, unauthorized data access, and Denial-of-Service (DoS) attacks. While PQC aims to provide robust encryption in the era of quantum computing, its integration into OS kernels needs careful attention. The increased computational and memory requirements of some PQC algorithms could inadvertently introduce new vulnerabilities or exacerbate existing ones, such as buffer overflows (Masdari and Jalali, 2016; Compastié et al., 2020) or timing side-channel attacks (Mushtaq et al., 2018). These vulnerabilities could allow attackers to bypass security controls, modify system processes, access secure data, and gain administrative privileges. Kernel exploits are particularly dangerous due to their high level of access, with the potential to completely undermine system integrity and confidentiality (Compastié et al., 2020). Therefore, in addition to adopting PQC, operating systems require rigorous testing and validation of their cryptographic implementations, along with a continued focus on securing against both traditional and quantum-enhanced threats to kernel security.

5.2.5. Hypervisor exploitation

As organizations transition to PQC, new potential vulnerabilities may emerge in the realm of virtualization, particularly within hypervisor environments. Attackers could exploit these vulnerabilities, such as those in hypervisor interfaces and configurations (Compastié et al., 2020), possibly using techniques like buffer overflows (Masdari and Jalali, 2016; Compastié et al., 2020), exacerbated by the larger key sizes and complex data structures of PQC algorithms. Such exploits could lead to the injection of malicious code into the hypervisor or VMs (OWASP, 2023; Hoffman, 2020). One critical consequence of this could be VM Hopping or Guest Jumping, where attackers gain unauthorized access to other VMs on the same host, breaching isolation mechanisms (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012). This type of attack not only undermines the integrity and confidentiality of the targeted VMs but also poses a significant threat to the entire virtualized infrastructure. Therefore, as hypervisors are updated to support PQC, it is crucial to ensure that their security measures are equally advanced to handle the unique challenges presented by PQC, including thorough testing and hardening of the hypervisor against both conventional and quantum-inspired exploitation techniques.

5.2.6. VM/hypervisor service exploitation

The shift from classical to PQC introduces new dynamics in managing virtual environments. With PQC, larger key sizes and potentially more complex cryptographic operations can increase the processing load on VMs and Hypervisors. This increased load could lead to vulnerabilities in scenarios such as VM/Hypervisor Denial-of-Service (DoS) attacks, where the system is overwhelmed by processing excessive encrypted traffic or cryptographic operations (Compastié et al., 2020; Huang et al., 2012; Masdari and Jalali, 2016). Such a scenario might lead to VM/Hypervisor or Cloud-Internal Denial of Service, wherein critical resources (like CPU time and memory) are excessively consumed, potentially causing system crashes or significant performance degradation. Moreover, the integration of PQC algorithms into VMs and hypervisors must be done with careful consideration to avoid creating new vulnerabilities. For instance, the computational and memory overhead associated with PQC might be exploited in VM Poaching attacks (Almutairy et al., 2019; Kedia et al., 2013), where an attacker aims to monopolize the resources of shared VMs, leading to service degradation for other users. It is crucial to monitor and optimally allocate resources in virtualized environments to ensure that the additional demands of PQC do not compromise overall system stability and security. Effective resource management strategies and regular security assessments are key in adapting to the heightened requirements of PQC in virtualized systems.

5.2.7. VM migration exploitation

VM migration (Oberheide et al., 2008), the process of transferring a virtual machine between host systems, often involves moving large volumes of encrypted data. Employing PQC algorithms as encryption methods in VM migration processes offers heightened security. However, these algorithms often come with increased computational overhead, potentially impacting the performance and latency of VM migrations. Beyond merely selecting robust PQC methods, there is a

need for a well-considered strategy that accounts for the practical implications of their implementation. This strategy must safeguard against quantum-specific threats, such as side-channel attacks tailored to PQC, while also ensuring that VM migrations remain operationally efficient. Thus, the transition to PQC in the realm of VM migration demands a holistic approach, one that prioritizes both quantum-resilient security and practical performance in the evolving cybersecurity landscape.

5.2.8. Oversized cryptography exploitation

The adoption of PQC introduces new challenges due to the inherently larger key sizes and, in some cases, larger ciphertexts or digital signatures compared to classical cryptographic algorithms. When these larger cryptographic elements are embedded within messages or data packets, the resulting oversized payloads can strain the processing capabilities of web services or network infrastructure. This strain may lead to increased latency or, in more severe cases, result in a Denial-of-Service (DoS) attack as systems struggle to handle the voluminous or complex encrypted data (Gruschka and Luttenberger, 2006; Masdari and Jalali, 2016). Careful implementation and optimization of PQC algorithms are therefore necessary to balance the security benefits of quantum resistance against the practical implications of increased data sizes, ensuring that network and service performance is not adversely impacted.

5.2.9. Program stack exploitation

In the context of transitioning to PQC, there are additional considerations for the security of the program stack, particularly regarding how stack memory and buffer handling might adapt to the larger key sizes and potentially different performance characteristics of PQC algorithms. While PQC is aimed at securing communications against quantum computer-based threats, its implementation might inadvertently affect how the software manages memory and handles errors, potentially impacting the stack's vulnerability to exploits like Return Oriented Programming (ROP) (Wang and Backer, 2016; Poulios et al., 2015). ROP is an advanced technique that allows attackers to manipulate the program stack to redirect software control flow and execute arbitrary code, usually by taking advantage of buffer overflows and other memory corruption vulnerabilities. The integration of PQC could alter the attack surface for such techniques if not properly managed. It is crucial for developers to be aware of these changes and proactively address potential new vulnerabilities in the program stack that could arise from PQC implementation, ensuring both cryptographic robustness and overall application security against techniques like ROP.

5.2.10. Cryptanalysis

The implementation of PQC algorithms is not immune to cryptanalytic attacks (Zhang et al., 2018; Otmani et al., 2010). Quantum attackers might exploit implementation flaws or leverage quantum algorithms to devise new types of cryptanalytic attacks, which are not yet fully understood in the current cryptographic landscape. This includes exploiting side-channel vulnerabilities, as detailed in Table 2, along with other attack vectors that could be more effectively executed using quantum computers. For instance, Grover's algorithm could be used to enhance brute-force attacks on symmetric cryptographic algorithms, necessitating the doubling of key sizes to maintain equivalent levels of security. Consequently, cryptanalysis in the quantum era demands not only algorithmic changes but also an increased emphasis on implementation robustness, algorithmic resilience, and the continual assessment of cryptographic strength against emerging quantum techniques. Vigilance in the design and deployment of cryptographic systems becomes paramount in ensuring the confidentiality and integrity of information in a post-quantum world.

5.2.11. Infrastructure resource exhaustion

The implementation of PQC algorithms could significantly impact infrastructure resources. PQC algorithms, designed to be secure against quantum computer-based attacks, often require more computational power and memory than many traditional cryptographic algorithms. This increase in resource demands could lead to scenarios where system infrastructure, including servers and networks, faces higher loads, particularly during intensive encryption/decryption operations (Gruschka and Luttenberger, 2006; Masdari and Jalali, 2016). Such a situation might not only degrade system performance but also increase vulnerability to conventional attack strategies like Denial-of-Service (DoS), where an attacker deliberately overwhelms a system with excessive requests (Compastié et al., 2020; Huang et al., 2012; Masdari and Jalali, 2016). The risk is particularly acute in environments with legacy systems or limited computational resources, which might struggle to cope with the added demands of PQC. Thus, transitioning to PQC necessitates a careful evaluation and potential upgrade of existing infrastructure to ensure it can handle the new cryptographic workload without compromising performance or security.

5.3. Security challenges in each infrastructure layer in the post-migration phase

The attack vectors identified in the preceding section manifest as distinct vulnerabilities across the entire infrastructure stack. This evolving post-migration landscape demands a comprehensive reassessment of security strategies for each layer. Table 9 provides a systematic overview of the security challenges encountered in each infrastructure layer following post-quantum migration. The subsequent sections deliver a granular, layer-by-layer analysis of these challenges, examining specific vulnerabilities and proposing targeted mitigation strategies to strengthen infrastructure resilience.

5.3.1. Application layer

The Application Layer faces significant PQC challenges, particularly in key and ciphertext management. These issues increase the risk of buffer overflow vulnerabilities (Masdari and Jalali, 2016; Compastié et al., 2020), which are exploited by various attack vectors, including side-channel attacks (Gruss et al., 2019; Esfahani et al., 2021) that leverage VM co-residence (Compastié et al., 2020). Malicious entities can exploit these vulnerabilities to launch a range of security threats, such as APTs, keyloggers, ransomware, and polymorphic malware, leading to severe consequences such as application control flow manipulation and unauthorized data access. To mitigate these risks, secure coding practices are essential, including bounds checking and code audits to prevent side-channel vulnerabilities. Robust VM isolation and strict resource allocation policies are crucial to protect against shared resource exploitation. Employing memory-safe programming languages, keeping software up-to-date, and deploying advanced intrusion detection systems are critical to prevent tampering and unauthorized access. Additionally, comprehensive defense strategies should include anti-malware solutions, network firewalls, and the promotion of secure user behaviors to safeguard against vulnerable network endpoints. For information disclosure threats, rigorous input validation, secure memory handling, frequent software patching, and intrusion detection systems are vital. Enhancing security with encrypted inter-VM communications, antivirus software, and application whitelisting provides further protection against spyware and malware. To address denial of service attacks, optimized input validation, effective memory management, and a resilient network infrastructure are necessary to handle oversized cryptographic payloads. Regular security audits, prompt application of updates, and advanced encryption protocols are imperative to prevent privilege escalation incidents by addressing potential side-channel vulnerabilities. The responsibility for implementing these security measures lies with vendors and implementers, and to some extent, users themselves.

5.3.2. Data layer

This layer is susceptible to information disclosure threats, particularly from quantum-capable adversaries who may utilize advanced cryptanalytic techniques to exploit vulnerabilities such as side-channel attacks and buffer overflows. These vulnerabilities can be exploited through various means, including the injection of spyware (Rawal et al., 2023; CrowdStrike, 2024) or keyloggers (CrowdStrike, 2024), as well as mathematical analysis attacks, such as optical attacks (Roy et al., 2022; Ravi et al., 2024; Joy Persial et al., 2011). The use of post-quantum cryptographic algorithms, while enhancing security, also introduces challenges due to their larger key and ciphertext sizes. These increased sizes can lead to a higher likelihood of buffer overflow attacks, which not only threaten data confidentiality and integrity but also present challenges for cryptographic agility and backward compatibility. To mitigate these risks, it is essential to implement quantum-resistant algorithms and employ memory protection techniques. Regular security audits and timely patching are also crucial. The responsibility for addressing these concerns is shared among vendors, who must provide secure hardware and software; implementers, who are responsible for secure system configuration; and users, who should adhere to best practices.

5.3.3. Runtime layer

At the runtime layer, adversaries may exploit vulnerabilities such as out-of-bounds memory access or memory corruption to facilitate attacks like Return Oriented Programming (ROP) and Jump-Oriented Programming (JOP) (Wang and Backer, 2016; Poulios et al., 2015). These sophisticated techniques manipulate the application's control flow by utilizing sequences of existing code in the runtime memory, potentially leading to unauthorized actions or the bypassing of security mechanisms. To counter such threats, it is crucial to implement control-flow integrity measures, stack canaries, and to use memory-safe programming languages. Regular patch management and runtime protection mechanisms are also vital in mitigating these risks. Responsibility for these countermeasures lies with both implementers and vendors, who must ensure the integrity and security of the runtime environment.

5.3.4. Middleware layer

This Layer, serving as a critical bridge between applications and operating systems, is susceptible to a range of security threats. Notably, the misuse of oversized cryptographic payloads poses a significant risk. Such payloads can lead to buffer overflow vulnerabilities, which, if exploited, may result in denial of service attacks, service impairment, or even arbitrary code execution (Masdari and Jalali, 2016; Compastié et al., 2020; Almutairy et al., 2019; Kedia et al., 2013). These incidents can undermine the trustworthiness and reliability of middleware services. To mitigate these risks, it is essential to implement input validation and adopt safe libraries and memory management practices. Furthermore, securing hypervisor configurations and interfaces against buffer overflow exploitation is crucial. Regular security audits, updates, and the use of hardware that mitigates side-channel risks can also play a vital role in protecting the middleware layer. Implementing robust isolation mechanisms, updating hypervisor security policies, and deploying advanced intrusion detection systems are additional countermeasures that can help safeguard against unauthorized data access or manipulation, resource exhaustion, and elevation of privilege attacks. The responsibility for these countermeasures falls on both vendors and implementers, highlighting the need for a collaborative approach to middleware security.

5.3.5. Operating systems layer

The Operating Systems (OS) layer is a critical component that, if compromised, can lead to significant security breaches. This layer is vulnerable to a variety of security threats, including spoofing, tampering, information disclosure, denial of service (DoS), and elevation of privilege. These threats exploit various vulnerabilities, such as kernel-level security breaches (Compastié et al., 2020), buffer overflow (Masdari and Jalali, 2016; Compastié et al., 2020), and side-channel attacks (Gruss et al., 2019; Esfahani et al., 2021), which can lead to unauthorized actions like forging signatures, data alteration, and system overloads. To mitigate spoofing threats, it is essential to conduct thorough testing and validation of cryptographic implementations and to keep kernel protection mechanisms up to date. Tampering, which involves unauthorized data access or modification, can be countered by regular security audits and implementing buffer overflow protections like stack canaries (Zhou et al., 2022), Address Space Layout Randomization (ASLR) (Nicula and Zota, 2019), and Data Execution Prevention (DEP) (Microsoft, 2023). Information disclosure risks, such as kernel memory exposure or hypervisor monitoring vulnerabilities, can be addressed through isolation techniques, enhanced access control, encryption, and hypervisor security hardening. DoS attacks, which aim to consume excessive resources or induce system overload, can be mitigated by scalable infrastructure, load balancing, resource management, and robust access control measures. Elevation of privilege occurs when attackers exploit hypervisor vulnerabilities or out-of-bounds memory access to gain unauthorized root-level execution. Preventative measures include hypervisor security hardening, memory protection mechanisms, and intrusion detection systems. The responsibility for implementing these countermeasures is shared among vendors, implementers, and users. Secure coding practices, regular patching, cybersecurity training, and the use of antivirus software are fundamental. Additional measures such as immutable logs, blockchain technology for integrity, and enhanced algorithm validation are recommended to prevent repudiation. Collaboration with law enforcement and proactive threat hunting are also crucial in mitigating code injection vulnerabilities. Through these collective efforts, the integrity and security of the OS layer can be maintained against emerging threats.

5.3.6. Virtualization layer

In the virtualization layer, security challenges such as information disclosure and denial of service are prevalent. Hypervisors and virtual machine monitors are vulnerable to side-channel attacks, including cache attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020), which can lead to information leakage. These vulnerabilities are exploited through various attack vectors, such as micro-architectural side-channel attacks (Shen et al., 2021) that can disclose sensitive information. To mitigate these threats, it is essential to implement strict isolation policies, utilize hardware security modules, and conduct regular security audits to detect and patch vulnerabilities. Additionally, resource exhaustion can be caused by cloud-internal denial of service attacks, where host resources are overwhelmed by heavy encrypted traffic from malicious VMs. Countermeasures include setting resource allocation limits, employing traffic shaping policies, and using anomaly detection systems to prevent resource exhaustion. Optimized VM migration protocols are also necessary to handle PQC encryption loads efficiently and maintain service continuity during heavy traffic periods. The responsibility for these countermeasures lies with both vendors and implementers, while users are also responsible for some aspects of denial of service mitigation.

5.3.7. Hardware layer

In the hardware layer, security is compromised by sophisticated side-channel attacks that exploit vulnerabilities in specific components such as cryptographic accelerators and shared caches within multi-core processors. These attacks are particularly concerning in shared or cloud-based infrastructures, as they can jeopardize the security of

multiple hardware modules. The vulnerabilities include firmware weaknesses (Cuozzo, 2016) and susceptibility to side-channel attacks (Schellenberg et al., 2018; Sayakkara et al., 2019; Açikkapi et al., 2019; Genkin et al., 2017; Le et al., 2019; Breier et al., 2020), which can be exploited through techniques like power-monitoring, electromagnetic, and acoustic cryptanalysis, as well as differential fault analysis attacks. Countermeasures to these threats involve implementing firmware integrity checks, utilizing hardware-based security modules, and applying side-channel attack mitigation techniques. Additionally, side-channel-resistant hardware design, isolation techniques, shielding, and noise generation can be employed to prevent information disclosure. To address the threat of tampering, secure logging, and non-repudiation mechanisms are recommended. Resource management and DoS mitigation protocols are necessary to prevent denial of service attacks, while the least privilege principle, user access control, and secure boot processes can help prevent the elevation of privilege. The responsibility for implementing these countermeasures falls on both vendors and implementers, with implementers and users also being accountable for certain aspects such as secure logging and resource management.

5.3.8. Storage layer

In the storage layer, the integrity and availability of data are at risk due to potential buffer overflow vulnerabilities that can be exploited by sophisticated wiper (CrowdStrike, 2024) and ransomware attacks (Craciun et al., 2019; Kilber et al., 2021). These vulnerabilities can lead to the compromise of various storage systems, including direct-attached storage, network-attached storage, and storage area networks. To mitigate these threats, it is essential to implement buffer overflow protection mechanisms and ensure regular software updates. These countermeasures fall under the responsibility of both vendors and implementers to safeguard against attacks that could damage file systems, RAID configurations, and storage protocols.

5.3.9. Network layer

The network layer is susceptible to a range of vulnerabilities, including those arising from cryptanalysis (Zhang et al., 2018), side-channel leakage (Lounis and Zulkernine, 2020), fault injection (Blömer et al., 2014), and fragmentation (Jafarigiv et al., 2020; Müller et al., 2020; Beernink, 2022). These vulnerabilities can lead to tampering, information disclosure, and denial of service. Sophisticated techniques such as cryptanalysis, side-channel attacks manipulating system processes or data, fault attacks leaking sensitive information, and fragmentation causing service disruptions are commonly employed in these threats. The risks are particularly pronounced during heavy VM migrations, where oversized encrypted messages can cause re-transmission issues, and the potential for decryption of in-transit data due to weaknesses in network security protocols is a concern. To counter these threats, it is essential to implement regular algorithm updates, enhanced encryption methods, isolation and noise generation techniques, fault detection and correction mechanisms, secure data erasure protocols, traffic management, and buffer size adjustments. Both vendors and implementers are responsible for these countermeasures, with users also playing a role in ensuring encryption at rest and employing memory-safe programming and access controls to prevent system configuration exploits and sabotage. To counteract these risks, a layered and resilient security approach is essential, incorporating continuous adaptation and quantum-resistant cryptographic integration, alongside robust key management and security architecture practices. This proactive defense should also utilize advanced techniques such as quantum key distribution (QKD), hardware-based entropy sources for random number generation, and continuous security monitoring to detect anomalies suggestive of quantum-level attacks.

Table 9 presents a comprehensive synthesis of our analysis regarding post-migration security challenges across various infrastructure layers. Following the structure of Table 8, this reference distills our extensive review into essential points. It categorizes the content into seven critical columns: (1) *Layer*, which identifies the scrutinized infrastructure component; (2) *STRIDE Threat*, listing applicable STRIDE threats; (3) *Vulnerabilities*, describing specific exploitable weaknesses; (4) *Exploited by*, linking the threats to vulnerabilities; (5) *Attack Vector*, exemplifying exploitation methods; (6) *Countermeasures*, suggesting practical mitigation strategies; and (7) *Responsibility*, designating stakeholder accountability for implementing these countermeasures.

5.4. Cross-layer attack propagation in post-migration infrastructure

While the preceding analysis examined post-migration threats within individual layers, this section examines how vulnerabilities cascade *across* layers through dependency relationships. Post-migration cascades differ fundamentally from pre-migration failures: rather than mathematical collapse of shared cryptographic trust anchors, they arise from implementation-specific weaknesses (computational overhead, artifact size inflation, and memory safety gaps) that require multi-stage chaining across dependency boundaries to propagate. Grounded in Table 9, there are three main cross-layer cascades: (1) *Side-Channel Leakage* via hardware physical leakage propagating through VM isolation boundaries to application compromise; (2) *Resource Exhaustion* via PQC artifact fragmentation cascading to application denial-of-service; and (3) *Memory Safety and Artifact Mismatch* via oversized storage artifacts triggering buffer overflows and data isolation breaches.

5.4.1. Side-channel cascades: Hardware to application

Extended execution times from PQC's elevated computational complexity amplify physical information leakage, creating observation windows that enable vertical propagation through isolation boundaries:

$$\text{Hardware (Power/EM Leakage)} \xrightarrow{\text{Correlate}} \text{Virtualization (VM Hopping)} \xrightarrow{\text{Lateral}} \text{Runtime (Kernel Exploit)} \xrightarrow{\text{Inject}} \text{Application (ROP/JOP)}$$

This cascade is *persistent but probabilistic*: each PQC operation provides fresh attack opportunities, but successful key extraction requires statistical correlation over thousands of iterations. Once hardware-layer keys leak to the hypervisor level, VM isolation boundaries collapse, enabling lateral propagation to middleware and vertical escalation through OS kernel exploits to runtime environments. Secondary paths include: (i) cache-timing attacks (Hardware →Middleware) exploiting increased memory access patterns, (ii) VM escape (Virtualization→OS) enabling host kernel compromise, (iii) direct privilege escalation (OS→Application) bypassing runtime isolation, and (iv) hypervisor exploitation (Middleware →Data) accessing memory across tenant boundaries.

5.4.2. Resource exhaustion cascades: Network to application

Oversized PQC artifacts fragment across packets when transmitted over standard MTU limits, creating computational stress that propagates upward:

$$\text{Network (Fragmentation)} \xrightarrow{\text{Reassembly}} \text{Middleware (Connection Overhead)} \xrightarrow{\text{Timeout}} \text{Application (DoS)}$$

This cascade occurs *without cryptographic compromise* as the failure mode is operational rather than algorithmic. In high-throughput scenarios (blockchain consensus, IoT telemetry aggregation, real-time financial trading), legitimate cryptographically-protected traffic becomes an unintentional DoS vector. Secondary cascades include: (i) traffic overload (Network→Data) saturating storage I/O during certificate validation, (ii) runtime traffic stress (Network→Runtime) affecting container scheduling under fragmented packet load, and (iii) computational exhaustion (Hardware →OS) from sustained PQC operations monopolizing CPU cycles.

5.4.3. Memory safety cascades: Storage to application

Fixed-size buffers designed for classical signatures encounter artifacts up to 72× larger, triggering memory corruption that propagates through multiple paths:

$$\text{Storage (Oversized)} \xrightarrow{\text{Persist}} \text{Application (Buffer Overflow)}$$
$$\xrightarrow{\text{Stack Corruption}} \text{Runtime (Control-flow Hijack)}$$
$$\xrightarrow{\text{Memory Leak}} \text{Data (Isolation Breach)}$$

These failures stem from *data structure mismatches* between classical and post-quantum artifact formats, a migration hazard independent of cryptographic strength. The primary cascade begins with buffer overflows at the application layer, which enable stack corruption that hijacks runtime control flow, ultimately leaking sensitive data through memory disclosure. Additionally, application-layer SQL injection exploits can reverse-propagate to modify data-layer records. Secondary cascades include: (i) credential store compromise (Storage→Runtime) where oversized encrypted blobs trigger parsing vulnerabilities, and (ii) data remanence attacks (Storage→Data) where inadequate sanitization of oversized key material enables forensic recovery.

These cascades create a cross-layer systemic stress gradient across the infrastructure stack. While achieving algorithmic security against quantum computers, PQC deployment introduces operational vulnerabilities through computational complexity (Hardware, Virtualization, OS), artifact size inflation (Network, Storage), and implementation immaturity (Middleware, Runtime, Application, Data). Unlike deterministic pre-migration cryptographic collapses, post-migration exploitation typically requires multi-stage chaining across dependency boundaries. Consequently, the threat model must expand beyond cryptanalysis to include constant-time implementation validation, side-channel hardening, infrastructure capacity planning for artifact-induced overhead, and comprehensive memory-safety verification.

5.5. Real-world deployment challenges

The transition from classical cryptography to post-quantum cryptographic (PQC) systems represents a complex infrastructure transformation. While the preceding subsections examined technical security implications, this section addresses practical deployment challenges in production environments. Addressing these hurdles, spanning hardware acceleration, hybrid integration, and performance monitoring, is essential to balance security robustness with operational constraints.

5.5.1. PQC hardware acceleration challenges

The computational demands of PQC algorithms significantly exceed those of classical primitives, necessitating hardware acceleration that is currently hindered by architectural limitations.

Current hardware limitations. Modern cryptographic accelerators and HSMs, designed for RSA and ECC, lack native support for the Number Theoretic Transform (NTT) and quasi-cyclic code operations central to lattice-based and code-based PQC schemes (Wang et al., 2020; Karl et al., 2024). As demonstrated in the performance analysis (Section 3.2.2), optimized software implementations of ML-KEM and ML-DSA still incur 3–5× higher computational costs than classical algorithms, creating prohibitive penalties for high-throughput applications like TLS termination. Furthermore, PQC algorithms introduce a 17–25× increase in artifact sizes, ML-KEM-768 requires 1184-byte public keys versus 64 bytes for ECDH P-256 (National Institute of Standards and Technology (NIST), 2023b), which strains existing memory architectures. This increased footprint degrades cache performance, particularly in multi-core systems where shared Last-Level Cache (LLC) resources face contention from concurrent PQC contexts (Shen et al., 2018).

Single instruction, multiple data (SIMD) and vector processing optimization. While lattice-based algorithms benefit from SIMD vectorization, achieving the potential 2–4× speedup requires algorithm-aware implementations aligned with AVX2 or AVX-512 register widths (Fitzgibbon and Ottaviani, 2024). However, these gains are platform-dependent; embedded ARM Cortex-M4 processors often lack vector capabilities entirely, forcing reliance on slower scalar operations (Kaps et al., 2019). Similarly, while GPUs can accelerate high-throughput batch operations, they offer limited benefits for latency-sensitive single-operation flows or power-constrained SoC environments where dedicated GPU resources are unavailable (Li et al., 2023).

Specialized hardware development. Developing dedicated PQC accelerators involves navigating obsolescence risks, complexity, and integration hurdles. Hardware designs initiated during standardization face "algorithm evolution risk", as seen when CRYSTALS-Kyber parameters changed for ML-KEM, rendering early designs obsolete (National Institute of Standards and Technology (NIST), 2023b). Production systems also require cryptographic agility, necessitating complex multi-algorithm support that increases silicon area and power consumption (Karabulut and Aysu, 2024). Furthermore, implementing hardware-level side-channel resistance, such as masking and constant-time execution against the attacks detailed in Table 2, substantially increases design complexity (Ravi et al., 2024). Finally, retrofitting these accelerators into legacy infrastructure is often infeasible due to interface and capacity mismatches in existing HSMs (Karl et al., 2024; Wang et al., 2020).

Resource constraints in embedded systems. Embedded and IoT deployments face acute performance gaps, with systems often constrained to <100 MHz clocks and minimal RAM. Benchmarking indicates ML-KEM-512 key generation can exceed 500 ms on Cortex-M4 platforms compared to 32 ms on server-grade processors, violating latency requirements for industrial control loops (Kannwischer et al., 2019). Additionally, PQC operations can reduce battery life by 30%–50% in energy-constrained devices (Septien-Hernandez et al., 2022). A critical dilemma arises in ROM-based devices: premature implementation risks locking in vulnerable algorithms (as demonstrated by the SIKE break (Castryck and Decru, 2023)), while delaying implementation renders devices permanently quantum-vulnerable due to non-updateable firmware.

5.5.2. Hybrid key exchange implementation

Hybrid schemes combining classical and PQC algorithms provide a necessary security fallback (D et al., 2025; Stebila and Mosca, 2020) but introduce substantial protocol and operational complexities.

Protocol complexity and integration. Hybrid key exchange in TLS 1.3 necessitates handling significantly larger payloads and complex negotiation logic. Concatenating classical (e.g., X25519) and PQC (e.g., ML-KEM-768) keys results in ClientHello messages exceeding 1200 bytes, approaching the IPv6 MSS of 1440 bytes and risking packet fragmentation or rejection by rigid middleboxes (Montenegro et al., 2024). Negotiation complexity increases as clients must balance bandwidth waste against round-trip latency (HelloRetryRequest) when offering hybrid versus standalone shares (Meta Engineering, 2024). Authentication migration remains distinct from confidentiality; while hybrid key exchange secures the session, maintaining classical signatures offers backward compatibility but leaves authentication vulnerable, whereas migrating to PQC signatures incurs significant certificate size overheads (Astrizi and Custodio, 2024). Finally, implementations must carefully handle failure modes to prevent downgrade attacks if one component of the hybrid pair fails (IETF TLS Working Group, 2024).

Table 9
Post-migration security challenges for each infrastructure layer.

Layer	STRIDE threat	Exploited vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
Application	Tampering	Buffer overflow vulnerabilities	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection exploiting memory buffer limits and side-channel attacks as detailed in Table 2.	Use memory-safe programming languages, maintain up-to-date software patches, and deploy advanced intrusion detection systems	Implementers, Vendors
		Hypervisor's security gaps	VM Hopping/Guest Jumping (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012)	Exploitation of hypervisor vulnerabilities through interface or configuration weaknesses (e.g., buffer overflow and code injection).	Conduct rigorous hypervisor security testing and enforce strict access controls	Vendors, Implementers
		Insufficient isolation mechanisms	VM Escape to VM (Compastié et al., 2020)	Exploiting buffer overflow vulnerabilities for unauthorized memory access	Enhance secure coding standards, monitor for anomalies, and verify isolation protocols	Implementers
		Insecure network endpoints	Wiper Malware Injection (CrowdStrike, 2024)	Utilization of buffer overflow vulnerabilities through network daemons, email clients, or web browsers	Deploy comprehensive anti-malware solutions, network firewalls, and promote secure user practices	Users, Vendors
	Repudiation	Insecure cryptographic implementations leading to stack-based or heap-based buffer overflows	VM Monitoring from VM (Compastié et al., 2020)	Side-channel attacks exploiting VM co-residence on shared resources	Implement secure coding practices such as bounds checking and regular code audits to prevent side-channel vulnerabilities	Vendors, Implementers
		Co-residence of VMs leading to shared resource exploitation	VM Monitoring from Host (Compastié et al., 2020)	Side-channel attacks exploiting shared resources in co-hosted VMs	Employ robust VM isolation mechanisms and strict resource allocation policies	Implementers, Users
	Info. Disclosure	Buffer overflows, Side-channel attacks, Insecure code execution	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020), VM Monitoring from VM (Compastié et al., 2020), VM Monitoring from Host (Compastié et al., 2020)	Code injection, Side-channel attacks	Input validation, Secure memory management, Regular software updates, Intrusion detection systems	Vendors, Implementers
			Inter-VM Communication Introspection (Compastié et al., 2020),Injecting Spyware (Rawal et al., 2023)	Buffer overflow, Side-channel attacks	Encryption of inter-VM communications, Antivirus software, Application whitelisting	Implementers
			Injecting Worm (Rawal et al., 2023), Injecting Rootkit (Rawal et al., 2023)	Exploiting control flow, Memory corruption	Control flow integrity, Memory-safe languages	Implementers
			Side-channel/Math. Analysis Attacks (Chen et al., 2010)	Cryptographic attacks, Math analysis	Post-quantum algorithms, Secure cryptographic implementations	Vendors, Academia, Research community
	Denial of Service	Oversized cryptographic payloads	Oversized Cryptography (Gruschka and Luttenberger, 2006; Masdari and Jalali, 2016)	Strained processing capabilities due to encrypted or digitally signed oversized payloads	Optimized input validation, efficient memory management, robust network infrastructure	Implementers
	Elevation of Privilege	Side-Channel Vulnerabilities	Data Theft (System, 2021)	Exploiting buffer overflow or side-channel vulnerabilities for code injection	Security audits, updates, and enhanced encryption protocols	Vendors, Implementers
Data	Info. Disclosure	Vulnerability to side-channel attacks and buffer overflows	Injecting Spyware (Rawal et al., 2023; CrowdStrike, 2024), Injecting Keylogger (CrowdStrike, 2024), Side-channel/Math. Analysis Attacks including Optical Attack (Roy et al., 2022; Joy Persial et al., 2011)	Code injection via buffer overflow, Information leakage through side-channel attacks	Implementing quantum-resistant algorithms, Employing memory protection techniques, Regular security audits and patching	Vendors (for providing secure hardware/software), Implementers (for secure system configuration), Users (for following best practices)
	Denial of Service	Increased computational load due to oversized cryptographic elements	Oversized Cryptography (Gruschka and Luttenberger, 2006; Masdari and Jalali, 2016)	Oversized payload causing system strain	Optimize cryptographic algorithms to reduce size and computational overhead	Implementers

(*continued on next page*)

Table 9 (*continued*).

Layer	STRIDE threat	Exploited vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
	Elevation of Privilege	Buffer overflow vulnerabilities	Data Theft (System, 2021)	Code injection via buffer overflow	Buffer overflow protections, secure coding practices	Vendors and Implementers
Runtime	Spoofing	Out-of-bounds memory access	Command Injections (Kim and Kwon, 2022; Hoffman, 2020)	Buffer overflow attacks	Implement input validation, use memory-safe languages, enforce least privilege	Implementers, Vendors
	Tampering	Memory corruption vulnerabilities	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection (via malware or out-of-bounds memory access vulnerability through buffer overflow) or side-channel attacks as mentioned in Table 2.	Use of memory-safe programming languages, regular patch management, and employing runtime protection mechanisms	Implementers, Vendors
		Out-of-bounds memory access	VM Escape to VM (Compastié et al., 2020)	Exploiting buffer overflow or code injection that executed at the root level	Enhanced VM monitoring, isolation techniques, and vulnerability scanning	Vendors, Implementers
		Program stack integrity compromise	Return Oriented Programming (ROP) (Wang and Backer, 2016; Poulios et al., 2015)	Buffer overflow leading to program stack control and corruption	Implementation of control-flow integrity measures and stack canaries	Implementers, Users
		Injection flaws	Command Injections (Kim and Kwon, 2022; Hoffman, 2020)	Exploiting out-of-bounds memory access vulnerabilities through buffer overflow	Input validation, parameterized queries, and use of least privilege principle	Implementers, Users
		Inadequate input validation	Command Injections (Kim and Kwon, 2022; Hoffman, 2020)	Buffer overflow due to out-of-bounds memory access.	Input validation and use of parameterized queries.	Users/Implementers
	Repudiation	Inadequate Logging	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection via buffer overflow or side-channel attacks	Implementing comprehensive logging and monitoring systems; Utilizing secure coding practices to prevent buffer overflows	Vendors and Implementers
	Info. Disclosure	Side-Channel Leaks	VM Monitoring from VM (Compastié et al., 2020)	Side-channel attacks exploiting the co-residency of VMs, leading to unauthorized monitoring and data leakage.	Implement robust logging and monitoring systems; Use of hardware-level isolation features.	Vendors, Implementers
			VM Monitoring from Host (Compastié et al., 2020)	Exploitation of side-channels mentioned in Table 2 to gain sensitive information from VMs.	Side-channel mitigation techniques; Secure coding practices; Regular security patches.	Vendors, Implementers
			Side-channel/Math. Analysis Attacks including Timing attack (Mushtaq et al., 2018)	Overloading systems through resource-intensive operations or attacks.	Resource usage monitoring, Load balancing, Redundancy planning	Implementers, Vendors
		Vulnerabilities in memory handling, unsafe buffer limits, out-of-bounds memory access	Command Injections (Kim and Kwon, 2022; Hoffman, 2020)	Out-of-bounds memory access	Regular security updates; Memory protection techniques	Vendors and Users
			Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection through buffer overflow or side-channel attacks (Table 2).	Implement memory safety practices and secure coding standards.	Vendors, Implementers
		Interception of Inter-VM Communication	Inter-VM Communication Introspection (Compastié et al., 2020)	Exploitation of shared networking infrastructure (Table 2).	Strong isolation and encryption mechanisms.	Implementers
	Denial of Service	Inefficient processing of encrypted traffic	VM Denial of Service (Compastié et al., 2020; Huang et al., 2012)	Excessive encrypted traffic causing resource exhaustion	Resource allocation strategies, DoS mitigation solutions	Implementers, Vendors
	Elevation of Privilege	Buffer overflow vulnerabilities	Application Software Exploitation (Compastié et al., 2020; Hoffman, 2020)	Code injection via malware or side-channel attacks (Table 2)	Memory protection mechanisms, Secure coding practices	Implementers, Vendors
		Memory access vulnerabilities, Root-level code execution	VM Escape to VM (Compastié et al., 2020)	Out-of-bounds access via buffer overflow or code injection at root level	Input validation, Stack protection techniques, Use of least privilege principle, code execution controls	Implementers
		Command injection susceptibility	Command Injections (Kim and Kwon, 2022; Hoffman, 2020)	Buffer overflow leading to unauthorized command execution	Boundary checks, Sanitization of input data	Implementers, Users

(*continued on next page*)

Table 9 (*continued*).

Layer	STRIDE threat	Exploited vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
		Hypervisor's weakness, interface or configuration vulnerability	VM Hopping/Guest Jumping (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012)	Exploiting hypervisor vulnerabilities (e.g., via buffer overflow and code injection)	Hypervisor security hardening, intrusion detection systems, and strict access control	Vendors, Implementers
		Insecure command execution	Command Injections (Kim and Kwon, 2022; Hoffman, 2020)	Out-of-bounds memory access via buffer overflow	Input validation, secure coding practices	Users
Middleware	Tampering	Buffer overflow vulnerabilities	Hypervisor Exploitation (Compastié et al., 2020)	Out-of-bounds memory access via buffer overflow, root-level code execution causing data leakage (host/VM to VM)	Implement input validation, use of safe libraries and memory management practices	Implementers
	Repudiation	Buffer overflow attacks leading to unauthorized data access or manipulation.	Hypervisor Exploitation (Compastié et al., 2020)	(a) Out-of-bounds memory access vulnerability via buffer overflow or injection of code, which can be executed at the root level and cause data leakage from host/VM to VM or (b) Side-channel attacks as mentioned in Table 2.	Implementation of rigorous input validation, adoption of memory-safe programming practices, and deployment of intrusion detection systems capable of identifying such exploitation attempts.	Vendors and Implementers
	Info. Disclosure	Side-channel attacks	VM Escape to Host (Compastié et al., 2020)	Exploiting side-channel vulnerabilities for data leakage (host to VM)	Utilize hardware that mitigates side-channel risks, enforce strict access controls	Vendors, Implementers
		Insecure inter-VM communications	Hypervisor Exploitation (Compastié et al., 2020)	(a) Buffer overflow or code injection executed at the root level causing data leakage from host/VM to VM. (b) Side-channel attacks as mentioned in Table 2.	Implement robust isolation mechanisms, update hypervisor security policies, and deploy advanced intrusion detection systems.	Vendors and Implementers
		Side-channel vulnerabilities	Cross-VM Side-channel Attacks including Cache Attacks and Math. Analysis Attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020)	Information leakage through side-channel/math. analysis attacks.	Regularly update and patch hypervisors, use noise-generating techniques to obscure side-channels, and employ strict access controls.	Vendors and Implementers
		Micro-architectural vulnerabilities	Micro-architectural Cache Side-channel Attacks such as Covert-channel Attacks, Rowhammer, and Transient Execution Attacks (Shen et al., 2021; Lou et al., 2021)	Exploitation of micro-architectural weaknesses leading to data leakage.	Harden security against specific attacks (e.g., rowhammer), use hardware that resists known vulnerabilities and perform continuous monitoring.	Vendors and Implementers
	Denial of Service	Resource exhaustion	Hypervisor Denial of Service (Masdari and Jalali, 2016; Compastié et al., 2020)	Overloaded hypervisor due to large encrypted traffic	Load balancing, traffic analysis, DoS protection mechanisms	Implementers, Users
		Hypervisor configuration flaws	VM Hopping/Guest Jumping (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012)	Exploiting hypervisor interface or configuration via buffer overflow	Secure configuration, regular security audits, hypervisor updates	Vendors, Implementers
		Overloaded resource allocation	VM Poaching (Almutairy et al., 2019; Kedia et al., 2013)	Draining hypervisor resources causing crashes	Resource usage monitoring, infrastructure scaling	Implementers
	Elevation of Privilege	Unauthorized access to root level	VM Escape to Host (Compastié et al., 2020)	Buffer overflow leading to root level access from VM	Strict access controls, vulnerability scanning, privilege restrictions	Implementers, Users
OS	Spoofing	Kernel-level security breaches	Injecting Virus (Sharma, 2024)	Code injection via buffer overflow or side-channel attacks potentially allowing to forge signatures.	Implementing rigorous testing and validation of cryptographic implementations, updating kernel protection mechanisms	Vendors and implementers
	Tampering	Unauthorized data access or modification, Code injection vulnerability	Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Exploiting vulnerabilities like buffer overflow or side-channel attacks to modify system processes	Regular security audits, application of buffer overflow protections like stack canaries, ASLR, DEP	Implementers and users
		Kernel manipulation, Buffer overflow vulnerabilities	OS Kernel Exploitation (Compastié et al., 2020)	Side-channel attacks, Forged signatures via side-channel vulnerabilities	Enhanced kernel protection, Secure coding practices, Regular security patches	Vendors, Implementers

(*continued on next page*)

Table 9 (*continued*).

Layer	STRIDE threat	Exploited vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
		Hypervisor's weakness	VM Hopping/Guest Jumping (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012)	Exploiting interface or configuration vulnerabilities	Strong isolation policies, secure hypervisor configurations	Vendors
		Out-of-bounds memory access	VM Escape to VM (Compastié et al., 2020), VM Escape to Host (Compastié et al., 2020)	Buffer overflow or injection of code leading to data leakage	Memory protection mechanisms, secure coding practices	Implementers
		Code injection vulnerability	Injecting Virus (Sharma, 2024), Trojan Horse (Khan et al., 2017), Spyware (Khan et al., 2017; CrowdStrike, 2024), Keylogger (CrowdStrike, 2024), Crimeware (Khan et al., 2017)	Buffer overflow or side-channel attacks forging signatures	Law enforcement collaboration, cybersecurity training, signature-based detection, behavior monitoring	Users
	Repudiation	Lack of proper logging mechanisms	Cryptanalysis Attacks (Otmani et al., 2010)	Advanced cryptanalysis exploiting PQC algorithm weaknesses	Immutable logs, Blockchain for integrity, Enhanced algorithm validation	Implementers, Users
	Info. Disclosure	Kernel memory exposure	OS Kernel Exploitation (Compastié et al., 2020)	Kernel manipulation via side-channel attacks (i.e., the ones mentioned in Table 2).	Implementing isolation techniques, enhanced access control, and encryption.	Vendors/Users
		Hypervisor monitoring vulnerabilities	VM Monitoring from Host (Compastié et al., 2020)	Side-channel attacks mentioned in Table 2.	Hypervisor security hardening, strict VM isolation protocols.	Vendors
		Inadequate access controls	VM Monitoring from VM (Compastié et al., 2020)	Side-channel attacks; VM co-residence	Enhanced access control mechanisms; Secure VM isolation	Implementers; Vendors
		Inter-VM communication eavesdropping	Inter-VM Communication Introspection (Compastié et al., 2020)	Side-channel attacks via co-residence of VMs or hosts, common networking infrastructure.	Secure communication channels, enhanced VM monitoring.	Implementers
		Insufficient process isolation	VM Monitoring from Host (Compastié et al., 2020)	Side-channel attacks	Use of secure and isolated hypervisors; Hardware-based security modules	Implementers; Vendors
		Leakage of sensitive information	Side-channel attacks including cache attacks (Gruss et al., 2019; Esfahani et al., 2021) and software-initiated fault attacks (Krautter et al., 2018)	Information leaked through side-channel/mathematical analysis attacks.	Side-channel attack mitigation techniques, secure coding.	Vendors
		Out-of-bounds memory access	Injecting Malware (Virus, Spyware, Keylogger) (Sharma, 2024; CrowdStrike, 2024)	Exploiting buffer overflow vulnerabilities, Side-channel attacks	Input validation, secure coding practices, regular software updates, application whitelisting, memory-safe languages, use of ASLR and DEP	Implementers, Vendors
		Unauthorized data access and leakage	Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Code injection via buffer overflow or side-channel attacks.	Use of hardware security modules, secure boot processes.	Vendors
		Sensitive data exposure through malicious software	Injecting Trojan Horse (CrowdStrike, 2024)	Code injection via buffer overflow or side-channel attacks.	Antivirus software, intrusion detection systems.	Users
		System compromise leading to data leakage	Injecting Fileless Malware (CrowdStrike, 2024; Singh and Tripathy, 2025)	Hijacking control flow of applications via ROP/JOP.	Behavior-based detection systems, system integrity checks.	Implementers
		Propagation of malware with data extraction capabilities	Injecting Worm (Rawal et al., 2023)	Hijacking application control flow via buffer-overflow vulnerability.	Network segmentation, proactive threat hunting.	Implementers
		Side-channel Vulnerabilities	Cross-VM Side-channel Cache Attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020)	Leveraging shared cache resources for data leakage	Isolation techniques, Cache partitioning	Vendors
	Denial of Service	Excessive resource consumption by PQC algorithms	infrastructure Resource Exhaustion	Overwhelming cryptographic engines with quantum operations	Scalable infrastructure, Load balancing, Resource management	Vendors, Implementers

(*continued on next page*)

Table 9 (*continued*).

Layer	STRIDE threat	Exploited vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
OS		Buffer Overflow Vulnerability	Buffer Overflow (Masdari and Jalali, 2016; Compastié et al., 2020)	Out-of-bounds memory access vulnerability causing system crashes or performance degradation.	Implement stack canaries, ASLR, DEP; Regular memory checks; Code auditing; Input validation	Vendors, Implementers
		VM Resource Exhaustion	VM Denial of Service (Masdari and Jalali, 2016; Compastié et al., 2020; Huang et al., 2012)	VMs exhausted by huge encrypted traffic, leading to service disruption.	Enhanced VM monitoring; Resource usage limits; Traffic shaping	Implementers, Users
		Hypervisor Resource Exhaustion	Hypervisor Denial of Service (Masdari and Jalali, 2016; Compastié et al., 2020)	Hypervisors exhausted by excessive encrypted traffic, leading to service disruption.	Robust access control; Enhanced hypervisor security; Traffic shaping	Vendors, Implementers
		Malware Induced System Overload	Injecting Virus (Sharma, 2024)	Malware through code injection overloading system resources.	Antivirus and anti-malware solutions; System hardening; Regular updates	Vendors, Implementers, Users
		System Compromise via Insecure Code	Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Injection of insecure code leading to unauthorized actions or system overload.	Secure coding practices; Memory-safe programming; Dynamic analysis tools	Implementers, Users
	Elevation of Privilege	Hypervisor vulnerabilities, Interface or configuration issues	VM Hopping/Guest Jumping (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012)	Exploiting hypervisor weaknesses (e.g., buffer overflow, code injection)	Hypervisor security hardening, Use of HSMs, Strong isolation policies	Vendors, Implementers
		Kernel manipulation vulnerabilities	OS Kernel Exploitation (Compastié et al., 2020)	Side-channel attacks	Enhanced kernel isolation, Secure coding practices, Regular patching	Implementers
		Hypervisor vulnerabilities	VM Hopping/Guest Jumping (Compastié et al., 2020; Almutairy et al., 2019; Huang et al., 2012)	Exploiting buffer overflow and code injection vulnerabilities	Hypervisor security updates, Intrusion detection systems	Vendors, Implementers
		Out-of-bounds memory access	VM Escape to VM (Compastié et al., 2020)	Buffer overflow or code injection attacks	Memory protection mechanisms, Access controls	Vendors, Implementers
		Unauthorized root-level execution	VM Escape to Host (Compastié et al., 2020)	Buffer overflow or code injection leading to data leakage	VM monitoring tools, Isolation techniques	Vendors, Implementers
		Code execution via vulnerabilities	Injecting Virus (Sharma, 2024)	Code injection through side-channels	Antivirus software, Security-hardened OS configurations	Users, Implementers
		Insecure code execution paths	Injecting Insecure Code (OWASP, 2023; Hoffman, 2020)	Code injection through buffer overflow	Secure code reviews, Static and dynamic analysis tools	Implementers
Virtualization	Info. Disclosure	Side-channel susceptibility	Side-channel Attacks including Cache Attack (Irazoqui et al., 2015)	Information leakage via side channels	Implementation of strict isolation policies and hardware security modules	Vendors
		Susceptibility to side-channel attacks	Cross-VM Side-channel Cache Attacks (Shahzad and Litchfield, 2015; Anwar et al., 2017; Saeed et al., 2020)	Information leaked through side-channel/math. analysis attacks (i.e., the ones mentioned in Table 2).	Implementation of strict isolation mechanisms and regular security audits to detect and patch vulnerabilities.	Vendors, Implementers
		Micro-architectural vulnerabilities	Micro-architectural Cache Attacks (Shen et al., 2021)	Information leaked through micro-architectural side-channel attacks.	Regular patching and use of side-channel resistant algorithms, enhanced monitoring of micro-architectural behavior and anomaly detection systems.	Vendors, Implementers
	Denial of Service	Resource exhaustion	Cloud-Internal DoS (CIDoS) (Masdari and Jalali, 2016)	Host resources exhausted by heavy encrypted traffic from malicious VMs.	Resource allocation limits and traffic shaping policies to prevent resource exhaustion and anomaly detection.	Implementers, Users
		VM migration traffic	VM Migration (Oberheide et al., 2008)	Heavy traffic during encrypted VM migration causing service delays.	Optimized VM migration protocols to handle PQC encryption loads efficiently and load balancing.	Vendors, Implementers

(*continued on next page*)

Table 9 (*continued*).

Layer	STRIDE threat	Exploited vulnerabilities	Exploited by	Attack vector	Countermeasures	Responsibility
Hardware	Tampering	Firmware vulnerabilities, Side-channel attack susceptibility	Firmware Exploitation (Cuozzo, 2016)	Code injection via buffer overflow, Side-channel attacks (Table 2)	Firmware integrity checks, Hardware-based security modules, Side-channel attack mitigation techniques	Vendors, Implementers
	Repudiation	Lack of firmware execution traceability	Firmware Exploitation (Cuozzo, 2016)	Code injection via buffer overflow, Side-channel attacks (Table 2)	Secure logging, Non-repudiation mechanisms	Implementers, Users
	Info. Disclosure	Side-channel data leakage	Side-channel Attacks (Schellenberg et al., 2018; Sayakkara et al., 2019; Açikkapi et al., 2019; Genkin et al., 2017; Le et al., 2019; Breier et al., 2020)	Power-monitoring, Electromagnetic, Acoustic cryptanalysis, Differential Fault Analysis attacks (Table 2).	Side-channel resistant hardware design, Isolation techniques, Shielding, Noise generation	Vendors, Implementers
	Denial of Service	Resource exhaustion	Energy-Oriented DoS (Masdari and Jalali, 2016)	Overburdened traffic causing resource exhaustion	Resource management, DoS mitigation protocols	Implementers, Users
	Elevation of Privilege	Firmware vulnerabilities, Side-channel attack susceptibility	Firmware Exploitation (Cuozzo, 2016)	Code injection via buffer overflow, Side-channel attacks (Table 2)	Least privilege principle, User access control, Secure boot processes	Vendors, Implementers
Storage	Info. Disclosure	Side-channel susceptibility	Side-channel Attack (Brumley, 2015)	Exploited through cache attacks and co-residency of attacker and victim.	Side-channel resistant hardware design, Isolation techniques, Shielding, Noise generation.	Vendors and Implementers
		Data remanence	Data Remanence Attack (Joshi, 2018)	Cold boot attacks performed on memory dump.	Encryption of all sensitive data at rest and secure deletion protocols.	Implementers and Users
	Denial of Service	Resource exhaustion	Energy-Oriented DoS (Masdari and Jalali, 2016)	Overburdened traffic draining infrastructure resources during encryption/decryption.	Upgrading infrastructure to handle increased loads and DDoS mitigation strategies, optimized cryptographic algorithms, load balancing.	Vendors and Implementers
	Elevation of Privilege	Buffer overflow vulnerabilities	Data Theft (System, 2021)	Code injection via out-of-bounds memory access vulnerability through buffer overflow or side-channel attacks.	Buffer overflow protection mechanisms and regular software updates.	Vendors and Implementers
Network	Tampering	Susceptibility to Cryptanalysis	Cryptanalysis Attacks (Zhang et al., 2018)	Mathematical analysis attacks mentioned in Table 2.	Regular algorithm updates, enhanced encryption methods	Vendors, Implementers
		Vulnerable to Side-channel Leakage	Side-channel Attacks (Lounis and Zulkernine, 2020)	Information leaked through side-channel attacks used to alter system processes or data.	Enhanced isolation, noise generation techniques	Vendors, Implementers
		Risk of Fault Injection	Fault Attacks (Blömer et al., 2014)	Information leakage through fault attacks	Fault detection and correction mechanisms	Vendors, Implementers
	Info. Disclosure	Risk of Data Remanence	Side-channel Attacks including Data Remanence Attack (Naik and Chatterjee, 2021)	Exploited through cold boot attacks performed on memory dump	Secure data erasure protocols, encryption at rest	Users, Implementers
	Denial of Service	Fragmentation-induced Overhead	TCP/UDP Fragmentation (Jafarigiv et al., 2020; Müller et al., 2020; Beernink, 2022)	Oversized encrypted messages causing re-transmission	Traffic management, buffer size adjustments	Vendors, Implementers
	Elevation of Privilege	System Configuration Exploits	System Sabotage (System, 2021)	Code injection via buffer overflow leading to system sabotage	Memory-safe programming, access controls	Users, Implementers

Performance overhead and optimization. Hybrid operations effectively double the cryptographic workload, adding 0.3–0.4 ms latency per exchange (Section 3.2.2) which compounds into substantial CPU load at scale (Ghinea et al., 2022; Goldberg et al., 2025; Joseph et al., 2022; D et al., 2025). However, network bandwidth often constitutes the primary bottleneck; the 12–25× increase in message sizes can dominate latency in constrained environments like satellite or cellular networks (Meta Engineering, 2024). TLS session resumption is also impacted, as NewSessionTicket messages incorporating PQC state may exceed 2 KB, triggering fragmentation issues that force organizations to consider alternative resumption strategies like session IDs (Meta Engineering, 2024).

Interoperability and standardization gaps. Standardization remains fragmented; while TLS 1.3 hybrid drafts are mature (IETF TLS Working Group, 2024), specifications for IKEv2, SSH, and hybrid signatures are less developed, risking interoperability failures (Turnip et al., 2025; Ma et al., 2021; Astrizi and Custodio, 2024). Early testing by Google and Cloudflare revealed numerous implementation bugs and parameter mismatches across libraries (Cloudflare, 2019), underscoring the need for extensive validation before production deployment.

5.5.3. Performance metrics and monitoring requirements

Traditional monitoring frameworks are insufficient for PQC; organizations require granular visibility into both cryptographic operations and end-to-end system impact.

Cryptographic operation metrics. Baselines must be established for latency, throughput, and resource consumption. Latency measurements must focus on tail percentiles (p99, p99.9), as cache effects can push ML-KEM-768 encapsulation from a 45 μs mean to over 100 μs (National Institute of Standards and Technology (NIST), 2023b). Throughput capacity must be re-evaluated for high-volume gateways (Rawal and Curry, 2024), and memory monitoring is critical given the 2–10× higher peak consumption of PQC algorithms, which can trigger exhaustion in multi-threaded environments (Banegas et al., 2022). Organizations should anticipate and plan for a 3–5× increase in CPU utilization for cryptographic workloads (Rawal and Curry, 2024).

End-to-end application impact. Monitoring must extend to user-visible metrics such as transaction latency and Time to First Byte (TTFB), which can increase by 15%–30% due to hybrid PQC overhead (Montenegro et al., 2024). SLA compliance verification is essential, particularly for geographically distributed systems where network latency exacerbates the impact of larger message sizes. Continuous resource monitoring is required to establish alerting thresholds that distinguish legitimate PQC workloads from anomalous behavior or attacks.

Benchmarking challenges. Current benchmarks often isolate algorithmic performance, failing to reflect integrated protocol behavior, multi-tenant contention, or variable network conditions (Banegas et al., 2022; Open Quantum Safe, 2025; Fitzgibbon and Ottaviani, 2024). While NIST provides reference benchmarks, organizations need data reflecting specific deployment scenarios, including protocol integration and long-term stability (Fitzgibbon and Ottaviani, 2024; Kannwischer et al., 2019; Banegas et al., 2022).

5.5.4. Organizational and operational challenges

Beyond technical implementation, PQC migration imposes significant organizational demands regarding workforce readiness, asset management, and operational resilience.

Skills gap and workforce readiness. PQC deployment requires specialized expertise in lattice-based and code-based mathematics, side-channel mitigation, and performance optimization, skills currently concentrated in academia rather than enterprise IT. The shortage of qualified personnel forces organizations to rely on slow internal training programs or vendor-managed services, the latter introducing lock-in risks and potential interoperability constraints (Canadian Centre for Cyber Security, 2025a).

Cryptographic asset discovery and inventory. Systematic migration requires comprehensive inventory of cryptographic dependencies across the infrastructure. Automated discovery tools often exhibit blind spots regarding compiled binaries, third-party components, and air-gapped systems (National Cybersecurity Center of Excellence, 2024; TNO et al., 2024). Consequently, discovery efforts frequently uncover shadow IT deployments and unexpected dependencies, extending migration timelines and necessitating continuous inventory maintenance to track evolving infrastructure configurations (Post-Quantum Cryptography Coalition, 2025; Barker et al., 2021).

Cryptographic agility and incident response. Operational resilience requires cryptographic agility, which encompasses modular implementations, policy-based algorithm selection, and runtime configuration flexibility, to adapt to future vulnerabilities without complete system redeployment (NIST, 2025). Incident response plans must specifically address cryptanalytic breaks, incorporating detection mechanisms for algorithmic compromise, emergency migration procedures, and large-scale data re-encryption capabilities to mitigate risks if algorithms such as those presented in Table 1 are compromised (Castryck and Decru, 2023; PQShield, 2024).

Regulatory complexity and compliance burden. Organizations must navigate an evolving compliance landscape characterized by fragmented adoption timelines and jurisdictionally distinct requirements. The increased evidentiary burden for hybrid cryptographic schemes, which requires demonstrations of security for both classical and post-quantum components, further complicates audit processes and compliance verification (TNO et al., 2024). A comprehensive analysis of specific mandates, certification pathways, and strategic compliance frameworks is presented in Section 5.5.5.

Phased migration strategy. Effective migration demands a phased approach governed by risk-based prioritization frameworks that identify high-vulnerability systems, particularly those susceptible to "harvest now, decrypt later" attacks (Post-Quantum Cryptography Coalition, 2025; Barenkamp, 2022). Pilot deployments are critical for validating performance assumptions, compatibility constraints, and vendor readiness prior to production rollout (NIST, 2025; Aydeger et al., 2024). Backward compatibility strategies must be carefully managed across multi-year transition periods to maintain service continuity while progressively reducing quantum vulnerability.

5.5.5. Regulatory readiness and compliance pathways

The transition to quantum-safe infrastructure requires navigating complex regulatory landscapes across multiple jurisdictions while maintaining operational continuity.

Government mandates and regulatory frameworks. National Security Memorandum 10 (NSM-10) mandates federal agencies adopt quantum-resistant cryptography by 2035, with OMB M-23-02 requiring quarterly cryptographic inventory reporting (The White House, 2022; Office of Management and Budget, 2022). NSA's CNSA 2.0 mandates ML-KEM-768/1024 for key establishment and ML-DSA-87 or FN-DSA-1024 for signatures in National Security Systems by 2030, extending to defense contractors and critical infrastructure (National Security Agency, 2022). In Canada, the Government's ITSM.40.001 roadmap similarly requires federal departments to execute PQC migration plans, mandating the transition of high-priority systems by 2031 and all remaining non-classified systems by 2035 (Canadian Centre for Cyber Security, 2025b). The EU's DORA (effective January 2025) imposes ICT risk management obligations encompassing quantum threats on financial entities (European Parliament and Council, 2022b), while NIS2 mandates quantum-safe cryptography implementation, though technical requirements remain under development (European Parliament and Council, 2022a). PCI DSS 4.0 introduces cryptographic agility and vulnerability monitoring requirements driving PQC readiness (PCI Security Standards Council, 2022). CISA provides post-quantum readiness guidance for critical infrastructure, recommending immediate cryptographic discovery despite absent binding mandates for most sectors (Cybersecurity and Infrastructure Security Agency, 2024). Financial sector "Harvest-Now, Decrypt-Later (HNDL)" exposure creates urgency, with institutions voluntarily accelerating migrations (Hansen and Nielsen, 2024).

Standards development and technical harmonization. NIST's FIPS 203, 204, and 205 (August 2024) establish technical compliance foundations with algorithm specifications (National Institute of Standards and Technology (NIST), 2023b,c,d). IETF develops protocol-specific guidance for TLS 1.3, IPsec, and SSH hybrid implementations (IETF TLS Working Group, 2024), while ETSI provides telecommunications infrastructure migration specifications (ETSI Cyber Working Group, 2024). ISO/IEC JTC 1/SC 27 WG 2 harmonizes international standards, though regional algorithm selection and parameter differences create interoperability challenges (ISO/IEC JTC 1/SC 27, 2024). Divergent timelines, such as NSA's 2030 deadline versus less urgent commercial guidance, force resource prioritization amid conflicting stakeholder demands. Geopolitical considerations and national security interests complicate multilateral harmonization (PQShield, 2024).

Certification and validation challenges. FIPS 140–3 requires updates for PQC algorithm evaluation, particularly side-channel resistance and implementation robustness (National Institute of Standards and Technology, 2024). NIST CMVP adapts testing requirements, but PQC implementation immaturity, absent side-channel countermeasures, and 12–18 month validation timelines delay certification (National Institute of Standards and Technology, 2024). Common Criteria faces Protection Profile development challenges, as lattice-based and code-based algorithm security properties differ fundamentally from classical cryptography (Common Criteria, 2024). Algorithm evolution risk, where advancing cryptanalysis potentially necessitates standard revisions (as demonstrated by the SIKE compromise (Castryck and Decru, 2023)), may invalidate certified implementations, increasing compliance costs. Hybrid cryptography regulatory treatment lacks clarity, with uncertainty whether hybrid implementations satisfy quantum-safe requirements or represent transitional states requiring pure PQC migration (NIST, 2025).

Audit, assurance, and documentation requirements. Evolving frameworks require cryptographic inventories documenting algorithm usage, key lengths, and certificate chains, which are formalized in NSM-10 for federal agencies and emerging in financial sectors (The White House, 2022; Hansen and Nielsen, 2024). Cryptographic Bill of Materials (CBOM), analogous to Software Bill of Materials (SBOM), standardizes cryptographic asset and quantum vulnerability documentation (Haag et al., 2022). Auditors require risk assessments, migration plans, and hybrid implementation test results, with hybrid schemes introducing verification requirements preventing downgrade attacks and ensuring proper failure handling (IETF TLS Working Group, 2024). Frameworks emphasize continuous monitoring over point-in-time assessments, necessitating automated cryptographic discovery, vulnerability scanning, and compliance reporting. Audit trails must demonstrate migration progress including pilot testing results, vendor certifications, and risk-based prioritization justifications (Post-Quantum Cryptography Coalition, 2025).

Supply chain and vendor management. Organizations must assess vendor PQC roadmaps, migration timelines, algorithm support, and certification plans while managing cryptographic dependencies in third-party libraries, HSMs, and TLS implementations. Procurement contracts require PQC migration commitments, cryptographic breach liability, and incident response obligations, creating cascading compliance burdens requiring ecosystem-wide quantum-readiness verification (Capgemini, 2024). Emerging third-party certification programs validate PQC implementations, with Common Criteria establishing evaluation methodologies likely becoming compliance prerequisites (Common Criteria, 2024).

Compliance pathways and strategic implementation. Organizations face distinct pathways based on sector, jurisdiction, and risk profile. Structured approaches involve regulatory gap assessment identifying requirements and prioritizing systems by data sensitivity, confidentiality lifetime, threat exposure, and deadlines (Post-Quantum Cryptography Coalition, 2025; TNO et al., 2024). High-risk entities handling classified information or critical infrastructure must prioritize early migration meeting 2030–2035 deadlines, requiring substantial capital investment and accepting newly standardized algorithm deployment risks (The White House, 2022; National Security Agency, 2022). Medium-risk organizations adopt phased approaches beginning with cryptographic discovery and hybrid deployments in non-critical systems, expanding as vendor ecosystems mature (TNO et al., 2024). Low-risk entities with limited sensitive data or short confidentiality requirements may defer migration while maintaining crypto-agility for tightening requirements or earlier quantum threats (Post-Quantum Cryptography Coalition, 2025). Pilot deployments gather performance data and compliance evidence before production rollout (Smith and Chen, 2024; Müller and Schmidt, 2024). Proactive regulator and auditor engagement through industry working groups (Post-Quantum Cryptography Coalition, sector-specific information sharing) aligns expectations, develops shared frameworks, reduces duplication, and promotes interoperability (Post-Quantum Cryptography Coalition, 2025). Cross-border operations require satisfying most stringent applicable requirements while managing multiple compliance postures. Documentation strategies encompass cryptographic assets, risk assessments, migration plans, test results, and compliance artifacts. Continuous monitoring tools enable cryptographic discovery, vulnerability management, and compliance reporting adaptation to evolving standards. Organizations integrating regulatory considerations into PQC migration strategies from inception position themselves to meet compliance deadlines while maintaining quantum-era operational resilience.

6. Conclusion

As the digital landscape evolves with the integration of PQC, we face not only a technological shift but also a strategic cybersecurity imperative. The emergence of quantum computing brings both unparalleled potential and new vulnerabilities, pressing us to rethink our cryptographic foundations. This paper has highlighted the multifaceted challenges and transformations needed across various infrastructure layers to safeguard against emerging quantum threats. The transition to PQC extends beyond merely adopting larger keys and ciphertexts; it necessitates a fundamental reshaping of our security architectures. Each infrastructure layer, from application to network management, must be rigorously examined and fortified in this new era. Addressing these vulnerabilities requires more than algorithmic updates; it demands a holistic and proactive security overhaul. Our journey towards robust PQC calls for collaborative efforts across academia, industry, and policy realms. This collective endeavor should not only focus on developing and standardizing resilient PQC algorithms but also on reevaluating our overall security postures to counteract sophisticated quantum-borne threats effectively. Future research should continue to drive algorithmic innovation, ensuring these new solutions are both secure and practically deployable. Equally important is the concurrent evolution of hardware and software systems to integrate these cryptographic advances smoothly. This progression also involves enhanced threat modeling and interdisciplinary collaboration, crucial for addressing real-world application challenges and elevating public awareness. In summary, the migration towards PQC represents a comprehensive, collaborative effort to protect our digital domains against upcoming quantum threats. It underscores the need for ongoing innovation, vigilance, and adaptation in our cybersecurity strategies. As quantum computing continues to advance, it is imperative that our defenses not only keep pace but also anticipate and preempt future vulnerabilities.

CRediT authorship contribution statement

Yaser Baseri: Writing – review & editing, Writing – original draft, Validation, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Vikas Chouhan:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Ali Ghorbani:** Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Açikkapi, M.Ş., Özkaynak, F., Özer, A.B., 2019. Side-channel analysis of chaos-based substitution box structures. IEEE Access 7, 79030–79043.

Alagic, G., Bros, M., Ciadoux, P., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Silberg, H., Smith-Tone, D., Waller, N., 2025. Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process. Tech. Rep. NIST IR 8545, National Institute of Standards and Technology, http://dx.doi.org/10.6028/NIST.IR.8545, URL https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8545.pdf.

Albrecht, M.R., Deo, A., Paterson, K.G., 2018. Cold boot attacks on ring and module LWE keys under the NTT. Cryptology ePrint Archive.

Aldeco Perez, R., 2012. Secure Provenance-Based Auditing of Personal Data Use (Ph.D. thesis). University of Southampton.

Almutairy, N.M., Al-Shqeerat, K.H., Al Hamad, H.A., 2019. A taxonomy of virtualization security issues in cloud computing environments. Indian J. Sci. Technol. 12 (3), 1–19.

Amazon Web Services, 2023. Overview of Amazon Web Services for Quantum Technologies. https://d1.awsstatic.com/whitepapers/overview-of-aws-quantum-technologies.pdf.

Anwar, S., Inayat, Z., Zolkipli, M.F., Zain, J.M., Gani, A., Anuar, N.B., Khan, M.K., Chang, V., 2017. Cross-VM cache-based side channel attacks and proposed prevention mechanisms: A survey. J. Netw. Comput. Appl. 93, 259–279.

Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Gueron, S., Guneysu, T., Melchor, C.A., et al., 2017. BIKE: bit flipping key encapsulation.

Arute, F., et al., 2019. Quantum supremacy using a programmable superconducting processor. Nature 574 (7779), 505–510.

Astrizi, T.L., Custodio, R., 2024. Seamless transition to post-quantum TLS 1.3: A hybrid approach using identity-based encryption. Sensors 24 (22), 7300.

Aydeger, A., Zeydan, E., Yadav, A.K., Hemachandra, K.T., Liyanage, M., 2024. Towards a quantum-resilient future: Strategies for transitioning to post-quantum cryptography. In: 2024 15th International Conference on Network of the Future. NoF, pp. 195–203. http://dx.doi.org/10.1109/NoF62948.2024.10741441.

Banegas, G., Zandberg, K., Baccelli, E., Herrmann, A., Smith, B., 2022. Quantum-resistant software update security on low-power networked embedded devices. In: Applied Cryptography and Network Security. Springer, Cham, pp. 872–891.

Barenkamp, M., 2022. "Steal Now, Decrypt Later" post-quantum-kryptografie & KI. Inform. Spektrum 45 (6), 349–355.

Barker, W., Souppaya, M., Newhouse, W., 2021. Migration to post-quantum cryptography. pp. 1–15, NIST National Institute of, Standards and Technology and National Cybersecurity, Center of Excellence.

Baseri, Y., Chouhan, V., Ghorbani, A., Chow, A., 2025a. Evaluation framework for quantum security risk assessment: A comprehensive strategy for quantum-safe transition. Comput. Secur. 150, 104272. http://dx.doi.org/10.1016/j.cose.2024.104272.

Baseri, Y., Chouhan, V., Hafid, A., 2024. Navigating quantum security risks in networked environments: A comprehensive study of quantum-safe network protocols. Comput. Secur. 103883.

Baseri, Y., Hafid, A., Shahsavari, Y., Makrakis, D., Khodaiemehr, H., 2025b. Blockchain security risk assessment in quantum era, migration strategies and proactive defense. IEEE Commun. Surv. Tutor. 1. http://dx.doi.org/10.1109/COMST.2025.3621113.

Bates, A., Mood, B., Pletcher, J., Pruse, H., Valafar, M., Butler, K., 2012. Detecting co-residency with active traffic analysis techniques. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop. pp. 1–12.

Beernink, G., 2022. Taking the quantum leap: Preparing DNSSEC for post quantum cryptography. http://essay.utwente.nl/89509/.

Bernstein, D.J., Buchmann, J., Dahmen, E., 2009. Introduction to post-quantum cryptography.

Bernstein, D.J., Chou, T., Lange, T., von Maurich, I., Misoczki, R., Niederhagen, R., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., et al., 2017. Classic mceliece: conservative code-based cryptography. NIST Submissions.

Bernstein, D.J., Hülsing, A., Kölbl, S., Niederhagen, R., Rijneveld, J., Schwabe, P., 2019. The sphincs+ signature framework. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2129–2146.

Bernstein, D., Lange, T., 2017. Post-quantum cryptography. Nature 549, 188–194. http://dx.doi.org/10.1038/nature23461.

Blömer, J., Günther, P., Liske, G., 2014. Tampering attacks in pairing-based cryptography. In: 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE, pp. 1–7.

Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D., 2018. CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy. EuroS&P, IEEE, pp. 353–367.

Brassard, G., Hoyer, P., Tapp, A., 1997. Quantum algorithm for the collision problem. arXiv Preprint Quant-Ph/9705002.

Breier, J., Khairallah, M., Hou, X., Liu, Y., 2020. A countermeasure against statistical ineffective fault analysis. IEEE Trans. Circuits Syst. II: Express Briefs 67 (12), 3322–3326.

Bruinderink, L.G., Pessl, P., 2018. Differential fault attacks on deterministic lattice signatures. IACR Trans. Cryptogr. Hardw. Embed. Syst. 21–43.

Brumley, B.B., 2015. Cache storage attacks. In: Cryptographers' Track At the RSA Conference. Springer, pp. 22–34.

Canadian Centre for Cyber Security, 2025a. Roadmap for the Migration to Post-Quantum Cryptography. Tech. Rep. ITSM.40.001, Government of Canada, URL https://www.cyber.gc.ca/en/guidance/roadmap-migration-post-quantum-cryptography-government-canada-itsm40001.

Canadian Centre for Cyber Security, 2025b. Roadmap for the migration to post-quantum cryptography (ITSM.40.001). https://www.cyber.gc.ca/en/guidance/roadmap-migration-post-quantum-cryptography-government-canada-itsm40001.

Capgemini, 2024. PQC migration unpacked: Four focus areas to build momentum. https://www.capgemini.com/insights/expert-perspectives/pqc-migration-unpacked/.

Castelnovi, L., Martinelli, A., Prest, T., 2018. Grafting trees: a fault attack against the SPHINCS framework. In: International Conference on Post-Quantum Cryptography. Springer, pp. 165–184.

Castelvecchi, D., 2017. Quantum computers ready to leap out of the lab in 2017. Nature 541 (7635).

Castryck, W., Decru, T., 2023. An efficient key recovery attack on SIDH. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 423–447.

Cayrel, P.-L., Colombier, B., Dragoi, V.-F., Menu, A., Bossuet, L., 2020. Message-recovery laser fault injection attack on code-based cryptosystems. IACR Cryptol. EPrint Arch. 2020, 900, IACR Cryptol. ePrint Arch.

Chakraborty, N., Kalaimannan, E., 2017. Minimum cost security measurements for attack tree based threat models in smart grid. In: 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference. UEMCON, IEEE, pp. 614–618.

Chen, S., Wang, R., Wang, X., Zhang, K., 2010. Side-channel leaks in web applications: a reality today, a challenge tomorrow. In: Proceedings of the IEEE Symposium on Security and Privacy (Oakland). IEEE Computer Society, proceedings of the ieee symposium on security and privacy (oakland) Edition.

Cheng, P., Wu, Z., Du, W., Zhao, H., Lu, W., Liu, G., 2025. Backdoor attacks and countermeasures in natural language processing models: a comprehensive security review. IEEE Transactions on Neural Networks and Learning Systems 36 (8), 13628–13648. http://dx.doi.org/10.1109/TNNLS.2025.3540303.

Cloudflare, 2019. Towards post-quantum cryptography in TLS. https://blog.cloudflare.com/towards-post-quantum-cryptography-in-tls/.

Common Criteria, 2024. Common Criteria for Information Technology Security Evaluation. Post-Quantum Cryptography Evaluation Methodologies. https://www.commoncriteriaportal.org/. (Accessed December 2025).

Compastié, M., Badonnel, R., Festor, O., He, R., 2020. From virtualization security issues to cloud protection opportunities: An in-depth analysis of system virtualization models. Comput. Secur. 97, 101905.

Craciun, V.C., Mogage, A., Simion, E., 2019. Trends in design of ransomware viruses. In: Innovative Security Solutions for Information Technology and Communications: 11th International Conference, SecITC 2018, Bucharest, Romania, November 8–9, 2018, Revised Selected Papers 11. Springer, pp. 259–272.

CrowdStrike, 2024. The 12 most common types of malware. https://www.crowdstrike.com/en-us/cybersecurity-101/malware/types-of-malware/. (Accessed December 2025).

Crypto4A, 2023. Quantum safe hardware security modules. https://crypto4a.com/hardware-security-modules/.

Cuozzo, G., 2016. Critical infrastructure cyber-attack through firmware exploitation.

Cybersecurity and Infrastructure Security Agency, 2023. Post-Quantum Cryptography Initiative. Tech. rep. CISA.

Cybersecurity and Infrastructure Security Agency, 2024. Post-Quantum Cryptography Initiative. Tech. rep. U.S. Department of Homeland Security, (Accessed December 2025).

D, F., P, M., Hale, B., 2025. Terminology for Post-Quantum Traditional Hybrid Schemes. In: Request for Comments. (9794), RFC Editor, http://dx.doi.org/10.17487/RFC9794, https://www.rfc-editor.org/info/rfc9794.

Dubrova, E., Ngo, K., Gärtner, J., 2022. Breaking a fifth-order masked implementation of CRYSTALS-kyber by copy-paste. Cryptology ePrint Archive.

Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D., 2018. Crystals-dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. 238–268.

Eastlake 3rd, D., Hansen, T., 2011. US Secure Hash Algorithms (SHA and SHA-Based HMAC and HKDF). Tech. rep.

Entrust, 2023. A comprehensive guide to quantum-resistant encryption. https://www.entrust.com/resources/learn/post-quantum-cryptography-and-encryption.

Esfahani, M., Soleimany, H., Aref, M.R., 2021. Enhanced cache attack on AES applicable on ARM-based devices with new operating systems. Comput. Netw. 198, 108407.

ETSI Cyber Working Group, 2024. Quantum-Safe Cryptography: Telecommunications and ICT Infrastructure Protection. Tech. rep. European Telecommunications Standards Institute, (Accessed December 2025).

European Parliament and Council, 2022a. Directive (EU) 2022/2555 on Measures for a High Common Level of Cybersecurity Across the Union (NIS2 Directive). Off. J. Eur. Union L 333, 80–152, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2555. (Accessed December 2025).

European Parliament and Council, 2022b. Regulation (EU) 2022/2554 on Digital Operational Resilience for the Financial Sector (DORA). Off. J. Eur. Union L 333, 1–79, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022R2554. (Effective 17 January 2025).

Faruk, M.J.H., Tahora, S., Tasnim, M., Shahriar, H., Sakib, N., 2022. A review of quantum cybersecurity: threats, risks and opportunities. In: 2022 1st International Conference on AI in Cybersecurity. ICAIC, IEEE, pp. 1–8.

Fitzgibbon, G., Ottaviani, C., 2024. Constrained device performance benchmarking with the implementation of post-quantum cryptography. Cryptography 8 (2), http://dx.doi.org/10.3390/cryptography8020021.

Fouque, P.-A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., 2018. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Submiss. NIST's Post-Quantum Cryptogr. Stand. Process. 36 (5).

Genêt, A., Kannwischer, M.J., Pelletier, H., McLauchlan, A., 2018. Practical fault injection attacks on sphincs. Cryptology ePrint Archive.

Genkin, D., Shamir, A., Tromer, E., 2017. Acoustic cryptanalysis. J. Cryptology 30 (2), 392–443.

Ghinea, D., Kaczmarczyck, F., Pullman, J., Cretin, J., Misoczki, R., Kölbl, S., Invernizzi, L., Bursztein, E., Picod, J.-M., 2022. Hybrid post-quantum signatures in hardware security keys.

Gidney, C., Ekerå, M., 2021. Quantum computing, postselection, and probabilistic polynomial-time. Phys. Rev. A 103 (3), 032414.

Gidney, C., Ekerå, M., 2019. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. arXiv preprint arXiv:1905.09749.

Gillmor, D., 2016. Negotiated finite field Diffie-Hellman ephemeral parameters for transport layer security (TLS). Tech. rep.

Goldberg, S., et al., 2025. Hybrid PQC integration across end-user, backbone, and cloud environments. In: ACM SIGSAC Conference on Computer and Communications Security.

Google Cloud Platform, 2023. Google cloud quantum computing service. https://quantumai.google/cirq/google/concepts.

Gort, M.L.P., Olliaro, M., Cortesi, A., 2022. Relational data watermarking resilience to brute force attacks in untrusted environments. Expert Syst. Appl. 118713.

Goy, G., Loiseau, A., Gaborit, P., 2022. A new key recovery side-channel attack on HQC with chosen ciphertext. In: International Conference on Post-Quantum Cryptography. Springer, pp. 353–371.

Grassi, P., Garcia, M., Fenton, J., 2020. Digital Identity Guidelines. Tech. rep. National Institute of Standards and Technology.

Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212–219.

Grover, V., et al., 2020. An efficient brute force attack handling techniques for server virtualization. In: Proceedings of the International Conference on Innovative Computing & Communications. ICICC.

Gruschka, N., Luttenberger, N., 2006. Protecting web services from dos attacks by soap message validation. In: IFIP International Information Security Conference. Springer, pp. 171–182.

Gruss, D., Kraft, E., Tiwari, T., Schwarz, M., Trachtenberg, A., Hennessey, J., Ionescu, A., Fogh, A., 2019. Page cache attacks. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 167–180.

Guo, Q., Hlauschek, C., Johansson, T., Lahr, N., Nilsson, A., Schröder, R.L., 2022. Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE. IACR Trans. Cryptogr. Hardw. Embed. Syst. 223–263.

Guo, Q., Johansson, T., Nilsson, A., 2020. A key-recovery timing attack on post-quantum primitives using the fujisaki-Okamoto transformation and its application on FrodoKEM. In: Annual International Cryptology Conference. Springer, pp. 359–386.

Gyongyosi, L., Imre, S., 2019. A survey on quantum computing technology. Comput. Sci. Rev. 31, 51–71.

Haag, W., Souppaya, M., Watrobski, P.J., 2022. Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations. Tech. Rep. NIST SP 800-161r1, National Institute of Standards and Technology, http://dx.doi.org/10.6028/NIST.SP.800-161r1, Discusses Cryptographic Bill of Materials (CBOM) concepts.

Hamburg, M., Hermelink, J., Primas, R., Samardjiska, S., Schamberger, T., Streit, S., Strieder, E., van Vredendaal, C., 2021. Chosen ciphertext k-trace attacks on masked CCA2 secure kyber. IACR Trans. Cryptogr. Hardw. Embed. Syst. 88–113.

Hansen, P., Nielsen, M., 2024. Financial Sector Post-Quantum Migration: Challenges and Timelines. J. Financ. Cryptogr. 8 (2), 145–162. http://dx.doi.org/10.1000/jfc.2024.migration, Discusses harvest-now-decrypt-later risks in financial sector.

Hoffman, A., 2020. Web Application Security: Exploitation and Countermeasures for Modern Web Applications. O'Reilly Media.

Huang, Y.-L., Chen, B., Shih, M.-W., Lai, C.-Y., 2012. Security impacts of virtualization on a network testbed. In: 2012 IEEE Sixth International Conference on Software Security and Reliability. IEEE, pp. 71–77.

IBM, 2023. Make the world quantum safe. https://www.ibm.com/quantum/quantum-safe.

Idhom, M., Wahanani, H.E., Fauzi, A., 2020. Network security system on multiple servers against brute force attacks. In: 2020 6th Information Technology International Seminar. ITIS, IEEE, pp. 258–262.

IETF TLS Working Group, 2024. Hybrid Key Exchange in TLS 1.3. Tech. Rep. draft-ietf-tls-hybrid-design-16, Internet Engineering Task Force.

Irazoqui, G., Eisenbarth, T., Sunar, B., 2015. S $ A: A shared cache attack that works across cores and defies VM sandboxing–and its application to AES. In: 2015 IEEE Symposium on Security and Privacy. IEEE, pp. 591–604.

ISACA, 2023. The Quantum Computing Threat: Risks and Responses. Tech. rep. ISACA.

ISO/IEC JTC 1/SC 27, 2024. ISO/IEC 23837-1: Security Techniques – Post-Quantum Cryptography. Tech. rep. International Organization for Standardization, Under Development. (Accessed December 2025).

Jafarigiv, D., Sheshyekani, K., Karimi, H., Mahseredjian, J., 2020. A scalable FMI-compatible cosimulation platform for synchrophasor network studies. IEEE Trans. Ind. Inform. 17 (1), 270–279.

Jangjou, M., Sohrabi, M.K., 2022. A comprehensive survey on security challenges in different network layers in cloud computing. Arch. Comput. Methods Eng. 1–22.

Jao, D., Feo, L.D., 2011. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: International Workshop on Post-Quantum Cryptography. Springer, pp. 19–34.

Joseph, D., Misoczki, R., Manzano, M., Tricot, J., Pinuaga, F.D., Lacombe, O., Leichenauer, S., Hidary, J., Venables, P., Hansen, R., 2022. Transitioning organizations to post-quantum cryptography. Nature 605 (7909), 237–243.

Joshi, S.B., 2018. Standards and techniques to remove data remanence in cloud storage. In: 2018 IEEE Punecon. IEEE, pp. 1–4.

Joy Persial, G., Prabhu, M., Shanmugalakshmi, R., 2011. Side channel attack-survey. Int. J. Adv. Sci. Res. Rev. 1 (4), 54–57.

Kannwischer, M.J., Genêt, A., Butin, D., Krämer, J., Buchmann, J., 2018. Differential power analysis of XMSS and SPHINCS. In: International Workshop on Constructive Side-Channel Analysis and Secure Design. Springer, pp. 168–188.

Kannwischer, M.J., Rijneveld, J., Schwabe, P., Stoffelen, K., 2019. pqm4: Testing and benchmarking NIST PQC on ARM Cortex-M4. In: Second PQC Standardization Conference. NIST, Santa Barbara, CA.

Kaps, J.-P., et al., 2019. Feasibility and performance of PQC algorithms on microcontrollers. In: NIST Second PQC Standardization Conference.

Karabulut, E., Aysu, A., 2021. FALCON down: Breaking FALCON post-quantum signature scheme through side-channel attacks. In: 2021 58th ACM/IEEE Design Automation Conference. DAC, pp. 691–696. http://dx.doi.org/10.1109/DAC18074.2021.9586131.

Karabulut, E., Aysu, A., 2024. Breaking Falcon post-quantum signature scheme through side-channel attacks. In: IEEE Design Automation Conference.

Karl, P., Schupp, J., Fritzmann, T., Sigl, G., 2024. Post-quantum signatures on RISC-V with hardware acceleration. ACM Trans. Embed. Comput. Syst. 23 (2), 1–23.

Kedia, P., Nagpal, R., Singh, T.P., 2013. A survey on virtualization service providers, security issues, tools and future trends. Int. J. Comput. Appl. 69 (24).

Khalil, S.M., Bahsi, H., Korõtko, T., 2023. Threat modeling of industrial control systems: A systematic literature review. Comput. Secur. 103543.

Khan, R., McLaughlin, K., Laverty, D., Sezer, S., 2017. STRIDE-based threat modeling for cyber-physical systems. In: 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe. ISGT-Europe, IEEE, pp. 1–6.

Kilber, N., Kaestle, D., Wagner, S., 2021. Cybersecurity for quantum computing. arXiv preprint arXiv:2110.14701.

Kim, J.-y., Kwon, H.-Y., 2022. Threat classification model for security information event management focusing on model efficiency. Comput. Secur. 102789.

Krautter, J., Gnad, D.R., Tahoori, M.B., 2018. FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. IACR Trans. Cryptogr. Hardw. Embed. Syst. 44–68.

Kumar, A., Garhwal, S., 2021. State-of-the-art survey of quantum cryptography. Arch. Comput. Methods Eng. 28, 3831–3868.

Le, D.-P., Yeo, S.L., Khoo, K., 2019. Algebraic differential fault analysis on simon block cipher. IEEE Trans. Comput. 68 (11), 1561–1572.

Lee, M., 2023. Quantum computing and cybersecurity. https://www.belfercenter.org/publication/quantum-computing-and-cybersecurity.

Li, S., Chen, Y., Chen, L., Liao, J., Kuang, C., Li, K., Liang, W., Xiong, N., 2023. Post-quantum security: Opportunities and challenges. Sensors 23 (21), http://dx.doi.org/10.3390/s23218744.

Lindsay, J.R., 2020. Demystifying the quantum threat: infrastructure, institutions, and intelligence advantage. Secur. Stud. 29 (2), 335–361.

Lipman, P., 2023. How quantum computing will transform cybersecurity. https://www.forbes.com/sites/forbestechcouncil/2021/01/04/how-quantum-computing-will-transform-cybersecurity/?sh=604540237d3f.

Lou, X., Zhang, T., Jiang, J., Zhang, Y., 2021. A survey of microarchitectural side-channel vulnerabilities, attacks, and defenses in cryptography. ACM Comput. Surv. 54 (6), 1–37.

Lounis, K., Zulkernine, M., 2020. Attacks and defenses in short-range wireless technologies for IoT. IEEE Access 8, 88892–88932.

Lu, Y., 2019. Attacking hardware aes with dfa. arXiv preprint arXiv:1902.08693.

Ma, C., Colon, L., Dera, J., Rashidi, B., Garg, V., 2021. CARAF: crypto agility risk assessment framework. J. Cybersecur. 7 (1), tyab013.

Mangla, C., Rani, S., Qureshi, N.M.F., Singh, A., 2023. Mitigating 5G security challenges for next-gen industry using quantum computing. J. King Saud Univ.-Comput. Inf. Sci. 35 (6), 101334.

Marzougui, S., Ulitzsch, V., Tibouchi, M., Seifert, J.-P., 2022. Profiling side-channel attacks on dilithium: A small bit-fiddling leak breaks it all. Cryptology ePrint Archive.
Masdari, M., Jalali, M., 2016. A survey and taxonomy of DoS attacks in cloud computing. Secur. Commun. Netw. 9 (16), 3724–3751.
Mashatan, A., 2022. A look at quantum resistant encryption & why it's critical. SSL Store.
McCarthy, S., Howe, J., Smyth, N., Brannigan, S., O'Neill, M., 2019. BEARZ attack FALCON: implementation attacks with countermeasures on the FALCON signature scheme. Cryptology ePrint Archive.
Melchor, C.A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Persichetti, E., Zémor, G., Bourges, I., 2018. Hamming quasi-cyclic (HQC). pp. 4–13, NIST PQC Round 2.
Merhav, N., Cohen, A., 2019. Universal randomized guessing with application to asynchronous decentralized brute–force attacks. IEEE Trans. Inform. Theory 66 (1), 114–129.
Meta Engineering, 2024. Post-quantum readiness for TLS at Meta. https://engineering.fb.com/2024/05/22/security/post-quantum-readiness-tls-pqr-meta/.
Microsoft, 2023. Data execution prevention. Microsoft Learn. https://learn.microsoft.com/en-us/windows/win32/memory/data-execution-prevention.
Microsoft Azure, 2023. Azure quantum cloud service. https://azure.microsoft.com/en-us/products/quantum/.
Migliore, V., Gérard, B., Tibouchi, M., Fouque, P.-A., 2019. Masking dilithium. In: International Conference on Applied Cryptography and Network Security. Springer, pp. 344–362.
Montenegro, J., et al., 2024. The impact of data-heavy, post-quantum TLS 1.3. In: NIST Fifth PQC Standardization Conference.
Moriarty, K., Kaliski, B., Jonsson, J., Rusch, A., 2016. PKCS# 1: RSA cryptography specifications version 2.2. Tech. rep.
Mosca, M., 2018. Cybersecurity in an era with quantum computers: will we be ready? IEEE Secur. Priv. 16 (5), 38–41.
Mosca, M., Piani, M., 2024. 2024 quantum threat timeline report. https://globalriskinstitute.org/publication/2024-quantum-threat-timeline-report/.
Müller, M., de Jong, J., van Heesch, M., Overeinder, B., van Rijswijk-Deij, R., 2020. Retrofitting post-quantum cryptography in internet protocols: a case study of DNSSEC. ACM SIGCOMM Comput. Commun. Rev. 50 (4), 49–57.
Müller, A., Schmidt, K., 2024. Deutsche Telekom Quantum-Safe Network Infrastructure Trials. Tech. rep. Deutsche Telekom, (Accessed December 2025).
Mushtaq, M., Akram, A., Bhatti, M.K., Rais, R.N.B., Lapotre, V., Gogniat, G., 2018. Run-time detection of prime+ probe side-channel attack on AES encryption algorithm. In: 2018 Global Information Infrastructure and Networking Symposium. IEEE, pp. 1–5.
Naik, P., Chatterjee, U., 2021. Network data remanence side channel attack on SPREAD, H-SPREAD and reverse AODV. In: International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, pp. 129–147.
National Cybersecurity Center of Excellence, 2024. Migration to Post-Quantum Cryptography: Cryptographic Discovery. NIST Special Publication NIST SP 1800-38, National Institute of Standards and Technology.
National Institute of Standards and Technology, 2001a. FIPS PUB 140-2: Security Requirements for Cryptographic Modules.
National Institute of Standards and Technology, 2001b. NIST Special Publication 800-38: Recommendation for Block Cipher Modes of Operation.
National Institute of Standards and Technology, 2018. NIST Special Publication 800-56A/B/C: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography.
National Institute of Standards and Technology, 2019a. FIPS PUB 140-3: Security Requirements for Cryptographic Modules.
National Institute of Standards and Technology, 2019b. NIST Special Publication 800-131A: Transitioning the Use of Cryptographic Algorithms and Key Lengths.
National Institute of Standards and Technology, 2020a. NIST Special Publication 800-57 Part 1: Recommendation for Key Management.
National Institute of Standards and Technology, 2020b. NIST Special Publication 800-90A/B/C: Recommendation for Random Number Generation Using Deterministic Random Bit Generators.
National Institute of Standards and Technology, 2023. NIST to Standardize Encryption Algorithms That Can Resist Attack by Quantum Computers. Tech. rep. NIST.
National Institute of Standards and Technology, 2024. Cryptographic Module Validation Program (CMVP). https://csrc.nist.gov/projects/cryptographic-module-validation-program. (Accessed December 2025).
National Institute of Standards and Technology, 2024a. Announcing approval of three federal information processing standards (FIPS) for post-quantum cryptography. https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved. (Accessed December 2025).
National Institute of Standards and Technology, 2025. NIST First Call for Multi-Party Threshold Schemes. NIST Interagency or Internal Report (IR) 8214C, U.S. Department of Commerce, URL https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8214C.2pd.pdf. Initial public draft (ipd) of FIPS 206 expected later in 2025, based on the Falcon submission.
National Institute of Standards and Technology (NIST), 2012. NIST Special Publication 800-30, Revision 1. NIST Special Publication 800-30, U.S. Department of Commerce.
National Institute of Standards and Technology (NIST), 2022. Post-quantum cryptography-selected algorithms 2022. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
National Institute of Standards and Technology (NIST), 2023a. Preparing for Post-Quantum Cryptography. Tech. rep. National Institute of Standards and Technology.
National Institute of Standards and Technology (NIST), 2023b. FIPS 203, module-lattice-based key-encapsulation mechanism standard. https://csrc.nist.gov/pubs/fips/203/ipd.
National Institute of Standards and Technology (NIST), 2023c. FIPS 204, module-lattice-based digital signature standard. https://csrc.nist.gov/pubs/fips/204/ipd.
National Institute of Standards and Technology (NIST), 2023d. FIPS 205, stateless hash-based digital signature standard. https://csrc.nist.gov/pubs/fips/205/ipd.
National Institute of Standards and Technology (NIST), 2023e. Post-quantum cryptography. https://csrc.nist.gov/projects/post-quantum-cryptography.
National Institute of Standards and Technology (NIST), Post-Quantum Cryptography Standardization: Security (Evaluation Criteria). https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria). (Accessed December 2025).
National Security Agency, 2022. Commercial National Security Algorithm Suite 2.0. Tech. rep.
Nature, 2023. Keeping secrets in a quantum world. Nature.
Nicula, S., Zota, R.D., 2019. Exploiting stack-based buffer overflow using modern day techniques. Procedia Comput. Sci. 160, 9–14.
NIST, 2022. Status report on the third round of the NIST post-quantum cryptography standardization process. https://csrc.nist.gov/publications/detail/nistir/8413/final.
NIST, 2025. Considerations for Achieving Crypto Agility: Strategies and Practices. Tech. rep. National Institute of Standards and Technology, http://dx.doi.org/10.6028/NIST.CSWP.39.2pd.
Oberheide, J., Cooke, E., Jahanian, F., 2008. Empirical exploitation of live virtual machine migration. In: Proc. of BlackHat DC Convention. Citeseer, pp. 1–6.
Oder, T., Schneider, T., Pöppelmann, T., Güneysu, T., 2016. Practical CCA2-secure and masked ring-LWE implementation. Cryptology ePrint Archive.
Office of Management and Budget, 2022. Migrating to Post-Quantum Cryptography (OMB Memorandum M-23-02). Tech. rep. Executive Office of the President, (Accessed December 2025).
Open Quantum Safe, 2025. OQS algorithm performance visualizations. https://openquantumsafe.org/benchmarking/. (Accessed June 2025).
Otmani, A., Tillich, J.-P., Dallot, L., 2010. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. Math. Comput. Sci. 3 (2), 129–140.
OWASP, 2023. Code injection - OWASP. https://owasp.org/www-community/attacks/Code_Injection.
Pacific Northwest National Laboratory, 2023. Inventory of Public Key Cryptography in US Electric Vehicle Charging Infrastructure. Tech. rep.
PCI Security Standards Council, 2022. Payment Card Industry Data Security Standard (PCI DSS) Requirements and Testing Procedures, Version 4.0. Tech. rep. PCI Security Standards Council, (Accessed December 2025).
Pessl, P., Primas, R., 2019. More practical single-trace attacks on the number theoretic transform. In: International Conference on Cryptology and Information Security in Latin America. Springer, pp. 130–149.
Post-Quantum Cryptography Coalition, 2025. Post-Quantum Cryptography Coalition (PQCC). https://pqcc.org/. (Accessed December 2025).
Post-Quantum Cryptography Coalition, 2025. Post-Quantum Cryptography (PQC) Migration Roadmap. https://pqcc.org/wp-content/uploads/2025/05/PQC-Migration-Roadmap-PQCC-2.pdf. (Accessed December 2025).
Poulios, G., Ntantogian, C., Xenakis, C., 2015. Ropinjector: Using return oriented programming for polymorphism and antivirus evasion. Blackhat USA.
PQShield, 2024. NCSC's annual review raises the profile of PQC migration. https://pqshield.com/ncscs-annual-review-raises-the-profile-of-pqc-migration/.
Quantum-Readiness Working Group (QRWG) of the Canadian Forum for Digital Infrastructure Resilience (CFDIR), 2023. Canadian national quantum-readiness, best practices and guidelines, version 03. https://ised-isde.canada.ca/site/spectrum-management-telecommunications/sites/default/files/attachments/2023/cfdir-quantum-readiness-best-practices-v03.pdf.
Ravi, P., Bhasin, S., Roy, S.S., Chattopadhyay, A., 2020a. Drop by Drop you break the rock-Exploiting generic vulnerabilities in Lattice-based PKE/KEMs using EM-based Physical Attacks. Cryptology ePrint Archive.
Ravi, P., Bhasin, S., Roy, S.S., Chattopadhyay, A., 2021. On exploiting message leakage in (Few) NIST PQC candidates for practical message recovery attacks. IEEE Trans. Inf. Forensics Secur. 17, 684–699.
Ravi, P., Chattopadhyay, A., D'Anvers, J.P., Baksi, A., 2024. Side-channel and fault-injection attacks over lattice-based post-quantum schemes (Kyber, Dilithium): Survey and new results. ACM Trans. Embed. Comput. Syst. 23 (2), 1–54.
Ravi, P., Jhanwar, M.P., Howe, J., Chattopadhyay, A., Bhasin, S., 2019a. Exploiting determinism in lattice-based signatures: practical fault attacks on pqm4 implementations of NIST candidates. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. pp. 427–440.
Ravi, P., Roy, D.B., Bhasin, S., Chattopadhyay, A., Mukhopadhyay, D., 2019b. Number "not used" once-practical fault attack on pqm4 implementations of NIST candidates. In: International Workshop on Constructive Side-Channel Analysis and Secure Design. Springer, pp. 232–250.

Ravi, P., Roy, S.S., Chattopadhyay, A., Bhasin, S., 2020b. Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020 (3), 307–335.
Rawal, B.S., Curry, P.J., 2024. Challenges and opportunities on the horizon of post-quantum cryptography. APL Quantum 1 (2), 026110.
Rawal, B.S., Manogaran, G., Peter, A., 2023. Malware. In: Cybersecurity and Identity Access Management. Springer, pp. 103–116.
Rosenberg, I., 2019. Post-quantum cryptography and quantum-safe security. IBM Research Blog.
Roy, S., Farheen, T., Tajik, S., Forte, D., 2022. Self-timed sensors for detecting static optical side channel attacks. In: 2022 23rd International Symposium on Quality Electronic Design. ISQED, IEEE, pp. 1–6.
Saeed, A., Hussain, S.A., Garraghan, P., 2020. Cross-VM network channel attacks and countermeasures within cloud computing environments. IEEE Trans. Dependable Secur. Comput.
Sayakkara, A., Le-Khac, N.-A., Scanlon, M., 2019. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. Digit. Investig. 29, 43–54.
Schaad, J., 2003. Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS). Tech. rep.
Schamberger, T., Renner, J., Sigl, G., Wachter-Zeh, A., 2020. A power side-channel attack on the CCA2-secure HQC KEM. In: International Conference on Smart Card Research and Advanced Applications. Springer, pp. 119–134.
Schellenberg, F., Gnad, D.R., Moradi, A., Tahoori, M.B., 2018. Remote inter-chip power analysis side-channel attacks at board-level. In: 2018 IEEE/ACM International Conference on Computer-Aided Design. ICCAD, IEEE, pp. 1–7.
Septien-Hernandez, J.-A., et al., 2022. A comparative study of post-quantum cryptosystems for Internet-of-Things applications. Sensors 22 (2), 489.
Shahzad, A., Litchfield, A., 2015. Virtualization technology: Cross-VM cache side channel attacks make it vulnerable.
Shapna Akter, M., 2023. Quantum cryptography for enhanced network security: A comprehensive survey of research, developments, and future directions. pp. arXiv–2306, arXiv e-Prints.
Sharma, T., Protection and Security in Operating System. https://www.scaler.com/topics/protection-and-security-in-operating-system/.
Shen, C., Chen, C., Zhang, J., 2018. Micro-architectural cache side-channel attacks and countermeasures. In: Asia and South Pacific Design Automation Conference.
Shen, C., Chen, C., Zhang, J., 2021. Micro-architectural cache side-channel attacks and countermeasures. In: Proceedings of the 26th Asia and South Pacific Design Automation Conference. ASPDAC '21, Association for Computing Machinery, New York, NY, USA, pp. 441–448. http://dx.doi.org/10.1145/3394885.3431638.
Shen, W., Su, Y., Hao, R., 2020. Lightweight cloud storage auditing with deduplication supporting strong privacy protection. IEEE Access 8, 44359–44372.
Shibly, M., De Soto, B.G., 2020. Threat modeling in construction: An example of a 3D concrete printing system. In: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, vol. 37, IAARC Publications, pp. 625–632.
Shor, P.W., 1994. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. IEEE, pp. 124–134.
Shor, P.W., 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. 41 (2), 303–332.
Singh, R., Islam, S., Sunar, B., Schaumont, P., 2022. An end-to-end analysis of EMFI on bit-sliced post-quantum implementations. arXiv preprint arXiv:2204.06153.
Singh, N., Tripathy, S., 2025. Unveiling the veiled: An early stage detection of fileless malware. Comput. Secur. 150, 104231. http://dx.doi.org/10.1016/j.cose.2024.104231, URL https://www.sciencedirect.com/science/article/pii/S0167404824005376.
Smith, J., Chen, W., 2024. AT&T Post-Quantum Cryptography Pilot Deployment. Tech. rep. AT&T Labs Research, (Accessed December 2025).
Smith, J., Doe, J., 2019. Understanding POST request collision and web security. J. Web Secur. 27 (3), 101–115.
Souppaya, M., Scarfone, K., 2013. Guide to Malware Incident Prevention and Handling for Desktops and Laptops. Tech. Rep. Special Publication 800-83 Revision 1, National Institute of Standards and Technology, Gaithersburg, MD, http://dx.doi.org/10.6028/NIST.SP.800-83r1.
Stebila, D., Mosca, M., 2020. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. In: NIST Second PQC Standardization Conference.
System, E., 2021. Insider threat: Definition, types & countermeasures. https://www.ekransystem.com/en/blog/insider-threat-definition.
TechTarget, 2021. A cloud services cheat sheet for AWS, azure and google cloud. TechTarget.
TechTarget, 2023. Explore the Impact of Quantum Computing on Cryptography. Tech. rep. TechTarget.
The White House, 2022. National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems (NSM-10). https://www.whitehouse.gov/briefing-room/statements-releases/2022/05/04/national-security-memorandum-on-promoting-united-states-leadership-in-quantum-computing-while-mitigating-risks-to-vulnerable-cryptographic-systems/. (Accessed December 2025).
Tirado, E., Turpin, B., Beltz, C., Roshon, P., Judge, R., Gagneja, K., 2018. A new distributed brute-force password cracking technique. In: International Conference on Future Network Systems and Security. Springer, pp. 117–127.
TNO, AIVD, CWI, 2024. The PQC Migration Handbook: Guidelines for Migrating to Post-Quantum Cryptography. Tech. rep. Netherlands Organisation for Applied Scientific Research (TNO) and General Intelligence and Security Service (AIVD) and Centrum Wiskunde & Informatica (CWI), Revised and Extended Second Edition.
Turner, S., Brown, D., 2010. Use of elliptic curve cryptography (ECC) algorithms in cryptographic message syntax (CMS). Tech. rep.
Turnip, T.N., Andersen, B., Vargas-Rosales, C., 2025. Towards 6G authentication and key agreement protocol: A survey on hybrid post quantum cryptography. IEEE Commun. Surv. Tutor. 1. http://dx.doi.org/10.1109/COMST.2025.3567439.
Wafo-Tapa, G., Bettaieb, S., Bidoux, L., Gaborit, P., Marcatel, E., 2020. A practicable timing attack against HQC and its countermeasure. Adv. Math. Commun.
Wang, X., Backer, J., 2016. SIGDROP: Signature-based ROP detection using hardware performance counters. arXiv preprint arXiv:1609.02667.
Wang, W., Tian, S., Jungk, B., Bindel, N., Longa, P., Szefer, J., 2020. Parameterized hardware accelerators for lattice-based cryptography and their application to the HW/SW co-design of qTESLA. IACR Trans. Cryptogr. Hardw. Embed. Syst. 269–306.
Xagawa, K., Ito, A., Ueno, R., Takahashi, J., Homma, N., 2021. Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer, pp. 33–61.
Xu, Z., Pemberton, O., Roy, S.S., Oswald, D., Yao, W., Zheng, Z., 2021. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: the case study of kyber. IEEE Trans. Comput. 71 (9), 2163–2176.
Zhang, Z., Wang, Y., Xie, L., 2018. A novel data integrity attack detection algorithm based on improved grey relational analysis. IEEE Access 6, 73423–73433.
Zhou, Q., Dai, H., Liu, L., Shi, K., Chen, J., Jiang, H., 2022. The final security problem in IOT: Don't count on the canary!. In: 2022 7th IEEE International Conference on Data Science in Cyberspace. DSC, IEEE, pp. 599–604.