Machine Learning and Side-Channel Attacks on Post-Quantum Cryptography

Abiodun Olaluwe, Nouf Nur Nabilah, Sheikh Tareq Ahmed, Akshay Raghavendra Kulkarni and Annamalai Annamalai

Prairie View A&M University, Prairie View TX USA aolaluwe@pvamu.edu,nnabilah@pvamu.edu,sahmed13@pvamu.edu,arkulkarni@pvamu.edu,aaannamalai@pvamu.edu

Abstract. The transition to post-quantum cryptography (PQC) is accelerating due to the potential of quantum computing to compromise classical public-key cryptosystems. While standardized schemes such as CRYSTALS-Kyber, CRYSTALS-Dilithium, and SPHINCS+ offer strong theoretical security, practical deployments remain susceptible to physical-layer vulnerabilities, notably side-channel attacks (SCAs). SCAs exploit unintentional leakages in hardware and software implementations—such as power traces, electromagnetic emissions, and timing variations—to recover secret keys without altering the target system. These attacks are non-invasive, cost-effective, and applicable across diverse platforms, making them a critical threat vector for PQC in embedded and resource-constrained environments.

This survey provides a structured, in-depth review of SCAs targeting PQC implementations, encompassing both classical methods—such as Simple Power Analysis, Differential Power Analysis, Correlation Power Analysis, Template Attacks, and Mutual Information Analysis—and emerging machine learning (ML)-driven approaches. Special attention is given to deep learning models, including CNNs, RNNs, and MLPs, which have demonstrated superior performance in profiling attacks by automatically learning leakage patterns from high-dimensional trace data, even in the presence of countermeasures like masking and desynchronization.

We categorize and compare recent attack strategies, analyze their effectiveness against various PQC schemes, and examine the limitations of existing countermeasures. Finally, we identify open research challenges and outline hybrid defense strategies that integrate classical protections with adaptive, ML-aware mitigation techniques. This comprehensive synthesis aims to bridge the gap between PQC algorithm design and secure, implementation-level deployment in the quantum era.

Keywords: Post-Quantum Cryptography · Side-Channel Attacks · Machine Learning · Cryptographic Hardware · Cybersecurity

1 Introduction

The advent of practical quantum computing threatens the foundational security of modern digital infrastructure. Classical public-key systems such as RSA and ECC, which underpin secure communications and authentication, are vulnerable to quantum algorithms like Shor's and Grover's, [1, 2]. In response, post-quantum cryptography (PQC) has emerged as strategic, imperative, and offering cryptographic schemes based on mathematically hard problems believed to be resistant to both classical and quantum adversaries. These schemes—including lattice-based, code-based, and hash-based constructions—are undergoing standardization by the National Institute of Standards and Technology (NIST) [3, 4].

However, beyond mathematical soundness, the physical realization of PQC algorithms introduces new attack vectors, such as side-channel attacks (SCAs), fault injection attacks

(FIAs), electromagnetic (EM) analysis, and related physical leakage attacks. These physical attacks are not new: they have historically posed significant risks to conventional cryptographic implementations like RSA and AES. [5–9]

In general-purpose embedded systems, SCAs have demonstrated the ability to extract sensitive secrets (e.g., encryption keys, biometric templates) by exploiting physical phenomena such as power consumption, EM emissions, timing variations, and acoustic signals. For instance, classic differential power analysis (DPA) attacks have successfully recovered AES keys with just a few thousand power traces. Timing attacks on RSA can exploit differences in operation time for different key bits, enabling secret exponent recovery. Even trusted hardware elements like smart cards, TPMs, and HSMs have proven vulnerable under real-world physical attacks. When these physical attacks are scaled to high-assurance applications like smart payment cards, automotive ECUs, or IoT edge devices, the consequences are far-reaching: cloned keys, fraudulent transactions, or total system compromise. Consequently, the hardware security community has invested heavily in countermeasures such as masking, hiding, randomization, and formal leakage resilience proofs for conventional ciphers. [5–9]

Transitioning to post-quantum cryptographic primitives changes the risk landscape but does not remove it. PQC algorithms—such as lattice-based, hash-based, and code-based schemes—resist known mathematical cryptanalysis under quantum adversaries. However, they still run on physical hardware with power and timing characteristics that can leak intermediate computation values. Many PQC schemes, especially lattice-based KEMs and signatures, rely on structured polynomial multiplications and number-theoretic transforms (NTT). They also involve sampling from discrete Gaussian or binomial distributions, as well as rejection sampling. Each of these operations can create exploitable leakage points [4, 10–14].

Several studies show that SCAs on PQC hardware implementations can be more severe than on classic ciphers. This is due to higher algorithmic complexity, larger key sizes, and repetitive arithmetic operations. Recent CPA attacks on CRYSTALS-Kyber and Dilithium FPGA implementations achieved practical key recovery with only tens of thousands of traces. These results hold even for first-order masked designs. Power analysis and EM attacks on hash-based schemes such as SPHINCS+ and code-based schemes such as McEliece are still emerging research areas. However, they are likely to pose similar risks as these algorithms become more widely deployed [11, 13, 15–18].

While post-quantum cryptography (PQC) strengthens resilience against quantum adversaries at the algorithmic level [1, 2], hardware implementations inherit — and sometimes amplify — the same side-channel risks that have long threatened traditional cryptosystems [5, 8]. This reality underscores the urgent need for robust side-channel resistance, formal leakage assessments, and hardware-aware countermeasure designs as PQC moves toward deployment in embedded, cloud, and Internet of Things (IoT) environments [10]. Side-channel attacks (SCAs) exploit unintentional information leakages such as variations in power consumption, timing, and electromagnetic (EM) emissions to recover secret keys during cryptographic execution [5, 6, 8, 19]. These attacks remain effective even when the underlying algorithm is mathematically secure, making physical implementation security a critical concern in real-world PQC adoption [11, 12]. The challenge is especially acute for IoT devices and resource-constrained embedded systems [20, 21]. Such platforms have limited computational resources, power budgets, and memory, which restrict the use of heavyweight cryptographic countermeasures [22]. As IoT systems increasingly require quantum-resilient security, integrating PQC in a side-channel-resilient manner becomes a pressing priority [13]. Schemes such as CRYSTALS-Kyber, Dilithium, and HQC are actively explored for these environments due to their balance of security, efficiency, and relatively compact implementations [16, 23, 24]. The National Institute of Standards and Technology (NIST) has explicitly emphasized the importance of side-channel resistance

in its PQC standardization process [3, 25, 26]. In Round 3 and the ongoing Round 4, submitters must demonstrate not only cryptographic strength but also practical resistance to side-channel and fault injection attacks [3]. NIST requires supporting evidence such as secure implementation guidelines, test vector leakage assessments (TVLA), and the application of masking or fault countermeasures [10]. This reflects a shift toward holistic evaluation that includes implementation-level resilience [27]. Recent initiatives such as the PQC-SEP (Power Side-Channel Evaluation Platform) [10] and systematic frameworks like KpqBench [27] further support the community in benchmarking side-channel robustness of PQC candidates on real-world hardware.

This paper is organized as follows. Section 2 surveys post-quantum cryptographic families, their hardness assumptions, and side-channel considerations, and summarizes standards-era choices (Tables 1 and 2; Fig.1). Section3 introduces side-channel attack fundamentals and threat models with illustrative examples (Fig.2) and a summary of core techniques (Table3). Section 3.1 details classical SCAs (timing, SPA/DPA/CPA, EM, cache/microarchitectural, and fault attacks) and practical considerations. Section 3.2 examines machine-learning-based SCAs, profiling pipelines, and model choices, including cross-device and ensemble strategies; Section 3.2.1 provides a concise methodology reference. Section 4 consolidates empirical results across PQC schemes (e.g., Kyber, Dilithium, Saber, HQC, Falcon) and compares trace efficiency and scenarios (Tables 5 and 6). Section 5 catalogs countermeasures (conventional and ML-aware) with visual taxonomies and comparative guidance (Figs.4 and 5; Table7). Section 6 motivates HQC as a focused platform for reproducible leakage evaluation. Section 7 outlines open problems and benchmarking needs, and Section 8 concludes.

2 Post-Quantum Cryptographic Algorithms

Post-quantum cryptography (PQC) refers to cryptographic primitives designed to withstand attacks from both classical and quantum adversaries. With the looming threat of large-scale quantum computers capable of breaking RSA and ECC via Shor's algorithm, the National Institute of Standards and Technology (NIST) launched a global standardization effort to identify practical and secure alternatives [3]. This section provides a comprehensive survey of the major PQC families under consideration, highlighting their underlying mathematical assumptions, unique design trade-offs, and side-channel considerations. This foundational understanding builds the basis for subsequent analysis of their real-world vulnerabilities.

2.1 PQC Families

2.1.1 Lattice-Based Cryptography

Lattice-based schemes are among the most prominent candidates in the NIST PQC standardization process [3, 4, 25, 26]. Their security relies on the presumed hardness of problems such as the Shortest Vector Problem (SVP), Learning with Errors (LWE), Ring-LWE (RLWE), and Module-LWE (MLWE) [1, 2].

Key examples include:

- **CRYSTALS-Kyber**: A key encapsulation mechanism (KEM) based on Module-LWE. It is favored for its performance and relatively small key sizes [25]. Kyber uses structured lattices to enable efficient polynomial operations via the Number Theoretic Transform (NTT) [11]. Multiple side-channel studies have demonstrated key recovery through power analysis and deep learning profiling attacks [15, 23, 28].

- **CRYSTALS-Dilithium**: A digital signature scheme derived from the Kyber design but adapted for signatures [26]. It employs rejection sampling to ensure

correctness [16, 29]. Dilithium implementations on FPGA and microcontrollers have shown susceptibility to correlation power analysis (CPA) and deep learning attacks [30, 31].

- **FrodoKEM**: A conservative MLWE-based KEM without ring structure [4]. This design offers stronger theoretical guarantees but results in larger key sizes and slower computation [3]. While fewer side-channel results are public, its use of standard LWE still involves leakage-prone matrix multiplications.

- **NTRU**: One of the earliest practical lattice-based schemes, based on polynomial rings [4]. It is recognized for efficient polynomial arithmetic. Side-channel attacks on NTRU-like schemes have exploited multiplication leakage and error sampling processes [32].

- **Saber**: An MLWE-based KEM that uses power-of-two moduli to simplify implementation [17]. It has been the subject of deep learning-based profiling attacks, which recovered keys from masked implementations with relatively few traces [33].

Side-channel relevance: Lattice schemes frequently involve structured polynomial arithmetic, discrete Gaussian or binomial sampling, and rejection sampling [11, 13]. These steps introduce repetitive operations that are prone to power and timing leakage [14, 23]. NTT operations, in particular, are recognized leakage sources in profiling attacks against Kyber, Dilithium, and Saber [15–17].

2.1.2 Code-Based Cryptography

Code-based schemes derive security from the hardness of decoding random linear codes [4, 18]. This problem is widely believed to remain intractable even for quantum computers [1, 2]. These schemes have one of the longest security records in public-key cryptography [3].

Representative schemes include:

- **Classic McEliece**: Based on binary Goppa codes, Classic McEliece remains one of the most mature code-based encryption schemes [18, 34]. It offers high security margins, but has the drawback of large public keys, often hundreds of kilobytes in size [4]. Side-channel studies have demonstrated that decoding steps in McEliece, such as syndrome computation, can leak sensitive information [18].

- **HQC (Hamming Quasi-Cyclic)**: HQC is a KEM that uses quasi-cyclic moderate-density parity-check (QC-MDPC) codes [24, 35]. It aims to balance security and efficiency with moderate key sizes and structured decoding algorithms. Recent research has demonstrated practical power analysis and chosen-ciphertext side-channel attacks on HQC hardware implementations [13, 36, 37].

- **BIKE (Bit Flipping Key Encapsulation)**: BIKE employs QC-MDPC codes with an iterative bit-flipping decoder [3]. While it benefits from small key sizes compared to Classic McEliece, its decoding process has been shown to be sensitive to timing and cache-based leakages [12].

Side-channel relevance: Decoding algorithms such as syndrome computation, error location, and error correction often involve data-dependent branching and memory access patterns [13, 24, 36]. These patterns can leak secret information through timing, power, and electromagnetic side channels [18, 37]. Recent attacks on HQC and McEliece have shown that even constant-time implementations may be vulnerable if physical leakages are exploited [13, 36].

2.1.3 Multivariate Cryptography

Multivariate public-key cryptography (MPKC) derives security from the difficulty of solving systems of multivariate quadratic (MQ) equations over finite fields [4]. Solving MQ systems is NP-hard, even for quantum adversaries [1,2]. This family of schemes has been studied for decades and offers efficient signature generation on constrained devices [3]. Key examples include:

- **Rainbow**: A multivariate signature scheme based on the Unbalanced Oil and Vinegar (UOV) construction [4]. Rainbow was a finalist in the NIST PQC standardization process [3]. It was later withdrawn after successful key-recovery attacks exploiting structural weaknesses [38].
- **GeMSS (Great Multivariate Short Signature)**: A candidate built on hidden field equations (HFE) with minus and internal perturbation modifications [4]. GeMSS aimed to provide short signatures and high performance, but was eventually not selected for standardization [3].

Side-channel relevance: The evaluation of quadratic polynomials in MPKC involves many non-linear multiplications and additions [4]. These operations can reveal sensitive information if the implementation leaks intermediate coefficients or partial results [5,8]. Although multivariate schemes often have smaller key sizes than code-based or lattice-based PQC, physical side-channel resistance remains under-studied [8,39]. Existing literature shows that generic SCA techniques such as correlation power analysis (CPA) [19] and template attacks [40] could be adapted to exploit leakage in MPKC implementations. Given the growing interest in deploying PQC on IoT and embedded platforms, further SCA evaluations of MPKC schemes are necessary.

2.1.4 Hash-Based Signatures

Hash-based signature schemes derive their security from the pre-image and collision resistance of cryptographic hash functions [4, 41]. They are conceptually simple and resistant to known quantum algorithms [1,2].

Important examples include:

- **SPHINCS+**: A stateless hash-based signature scheme that uses Merkle trees and hypertree structures to achieve forward security [41]. It is one of the first NIST-approved post-quantum digital signature standards [10].
- **XMSS**: The eXtended Merkle Signature Scheme, an earlier stateful hash-based signature with strong provable security [4]. It requires careful state management to avoid key reuse vulnerabilities.

Side-channel relevance: Hash-based schemes have minimal algebraic structure, which can reduce leakage compared to lattice- or code-based schemes [8]. However, repeated hash computations, tree traversal, and state updates can still produce exploitable timing and power side channels on embedded devices [13,39]. Evaluations show that EM and power analysis can reveal intermediate hash values if no countermeasures are applied [6,12].

2.1.5 Isogeny-Based Cryptography

Isogeny-based cryptography relies on the hardness of finding isogenies—structure-preserving maps—between supersingular elliptic curves [4, 42]. These schemes stand out for their small key sizes compared to other PQC families.

Example:

- **SIKE**: Supersingular Isogeny Key Encapsulation, designed for compact public keys and ciphertexts [42]. It was part of the NIST PQC process but was later broken by classical cryptanalysis exploiting its underlying structure [43].

Side-channel relevance: Isogeny-based computations require elliptic curve point multiplications and additions, operations known to be vulnerable to timing and power analysis [5,8]. Leakage can occur during scalar multiplication loops and coordinate transformations, making implementations on embedded or FPGA platforms susceptible without proper countermeasures [9, 44].

2.2 Comparison of PQC Schemes

To summarize the diversity of PQC algorithms, Table 1 provides a comparative view of key properties and known side-channel considerations.

Table 1: Comparison of Major Post-Quantum Cryptographic Families

PQC Family	Representative Schemes	Mathematical Hardness	Key/Ciphertext Sizes	Side-Channel Notes
Lattice-Based	Kyber, Dilithium, Saber, FrodoKEM	SVP, LWE, MLWE	Small-to-moderate	Leakage in NTT, sampling, and polynomial multiplication
Code-Based	HQC, Classic McEliece, BIKE	Decoding random linear codes	Moderate-to-large	Syndrome decoding leaks; error correction branches
Multivariate	Rainbow, GeMSS	MQ problem	Small keys, large signatures	Polynomial evaluation may leak coefficients
Hash-Based	SPHINCS+, XMSS	Hash function preimage/collision resistance	Small keys, large signatures	Tree traversal and state updates may leak sensitive data
Isogeny-Based	SIKE	Supersingular isogeny problem	Very small keys	EC arithmetic operations such as point multiplication may leak information

This overview provides the technical basis for systematically examining side-channel vulnerabilities and defenses tailored to each PQC family. In July 2022, the NIST Post-Quantum Cryptography (PQC) Standardization Project finalized its initial round, marking the first major global step toward cryptographic primitives designed to resist quantum attacks [3, 10]. This decision culminated in the standardization of four leading schemes: ML-KEM (FIPS 203), ML-DSA (FIPS 204), SLH-DSA (FIPS 205), and FN-DSA (FIPS 206). These schemes span module-lattice, hash-based, and NTRU-lattice constructions, each with specific physical leakage profiles when deployed on real hardware.

Importantly, each of these schemes draws its security from fundamentally different mathematical hardness assumptions — ranging from the Module-LWE problem for Kyber and Dilithium to the collision resistance of hash functions for SPHINCS+, and the FFT-NTRU lattice structure for FALCON. Recent research has shown that even though these theoretical underpinnings are quantum-secure, physical implementations introduce vulnerabilities through side-channel leakages, which attackers can exploit with advanced profiling techniques [11, 15, 16, 27].

This section provides a detailed comparison of each standardized scheme and HQC (a leading code-based alternative). The aim is to map each family's structure, implementation challenges, and known side-channel vectors in depth, creating a clear roadmap for targeted machine-learning-based SCA research in the sections that follow.:

- **Module-Lattice-Based KEM:** ML-KEM (FIPS 203) [25]
- **Module-Lattice-Based DSA:** ML-DSA (FIPS 204) [26]
- **Stateless Hash-Based DSA:** SLH-DSA (FIPS 205) [41]

- **FFT-based NTRU Lattice DSA:** FN-DSA (FIPS 206)
- **Hamming Quasi-Cyclic (HQC):** Widely studied, code-based PQC [24, 36]

These algorithms span different hardness assumptions and show distinct side-channel attack (SCA) surfaces when deployed on hardware, especially in power, EM, and thermal leakage contexts [10, 11, 15, 16].

Table 2 serves as a comprehensive lens into the practical trade-offs and side-channel security considerations across the most critical post-quantum cryptographic (PQC) families. While many surveys touch briefly on the mathematical underpinnings, few explicitly juxtapose the unique implementation realities and physical leakage modes that differentiate lattice-based, hash-based, FFT-enhanced, and code-based schemes in the real world [10, 27, 45].

Each row in the table distills key insights from both standardization documents — including the recently ratified FIPS 203–206 standards — and the growing corpus of empirical side-channel analysis (SCA) research. For instance, the ML-KEM row highlights not only its robust Module-LWE foundation but also the precise butterfly operations in its Number Theoretic Transform (NTT) that serve as a well-documented leakage vector when mapped to hardware [15, 28]. In contrast, FN-DSA's FFT layers open distinct profiling windows for adversaries leveraging single-trace or EM-based attacks [31, 46].

Another unique strength of this table is its explicit coupling of key/ciphertext size profiles with implementation-specific risks. By clearly showing that SPHINCS+ (SLH-DSA) trades large signature sizes for a stateless design and hash function security, it invites implementers to weigh memory usage against the benefits of removing ephemeral state leakage channels [41]. Similarly, the HQC row demonstrates how repeating structures in quasi-cyclic code-based operations can be a double-edged sword: they boost mathematical efficiency but provide predictable patterns for deep learning profilers to exploit [36, 37].

Critically, this table does not stand alone — it is a springboard for your full technical sections ahead. By capturing these connections, you pave the way for a deep dive into concrete side-channel experiments, comparative cross-device and cross-family benchmarks, and your survey's broader ambition: to showcase how machine learning models, LLM-inspired trace tokenization, and formal verification can expose or mitigate subtle leakages hidden within each algorithm's physical manifestation.

The purpose of laying out these families side-by-side is not just to highlight differences in abstract security guarantees, but to force a practical reckoning with what happens when these algorithms are integrated into real silicon. In an era where deployment-readiness means secure key generation, encapsulation, signing, and verification in diverse, often resource-constrained embedded environments, the table's side-channel dimension is what moves it beyond a standard textbook chart. It becomes a blueprint for design teams, cert labs, and academic researchers who must anticipate real-world attackers exploiting not just mathematical weaknesses, but hardware-level imperfections and implementation oversights.

Future sections will return to each family, extracting lessons for how profiling attacks have evolved — from classical correlation power analysis to multi-modal, cross-platform transfer attacks powered by CNNs, Transformers, and LLMs. With this baseline, the survey is poised to shape a new generation of resilient, formally verified, and ML-aware countermeasure strategies tailored for the PQC era.

2.2.1 Module-Lattice KEM (ML-KEM)

Standard: ML-KEM, formalized in FIPS 203 [25], implements CRYSTALS-Kyber, which is built around the hardness of the Module-LWE problem. Kyber's efficient polynomial operations make it attractive for high-speed hardware acceleration. However, the reliance on the Number Theoretic Transform (NTT) introduces structural regularities that attackers

Table 2: Detailed Comparison of NIST/FIPS PQC Standards and Known Leakage Vectors

Standard	Algorithm	Hardness	Key/Signature Sizes	Known SCA Risks
FIPS 203	ML-KEM (Kyber)	Module-LWE	Small-to-medium	NTT butterfly leaks, rejection sampling, ephemeral key reuse [11, 15]
FIPS 204	ML-DSA (Dilithium)	Module-LWE	Small keys, medium signatures	Sampling leakage, modular multiplication, zeroization [16]
FIPS 205	SLH-DSA (SPHINCS+)	Hash preimage/collision	Small keys, large signatures	Merkle tree traversal, buffer reuse [10]
FIPS 206	FN-DSA (FALCON)	NTRU lattice, FFT	Small keys/signatures	FFT leaks, Gaussian sampler side-channels [46]
N/A	HQC	Quasi-cyclic code	Moderate-to-large	Syndrome decoding, EM single-trace extraction [37, 47]

can exploit, particularly through the butterfly operations in the transform stages [15]. Leakage arises when the same ephemeral keys are reused or when sampling operations are imperfectly masked.

Beyond the NTT, Kyber's rejection sampling has also been demonstrated to leak subtle dependencies on secret values, especially in constrained embedded implementations that cut corners on constant-time constraints [28]. Profiling attacks like those by Park et al. [15] and new works on deep-learning-assisted CPA show that leakage persists even when basic masking is deployed, because higher-order correlations often survive simplistic countermeasures.

NIST SP 800-227-IPD [48] offers official implementation guidelines for ML-KEM to mitigate such risks, including strict parameter checks, constant-time rejection sampling, and efficient error rejection. Nonetheless, hardware deployments on FPGAs and microcontrollers remain highly susceptible to EM and power-based profiling if countermeasures are not integrated with formal verification [27].

Guideline: NIST SP 800-227-IPD [48] clarifies implementation guidance for ML-KEM including parameter sets and efficient constant-time sampling. Recent hardware studies [27] confirm that power and EM leakages propagate across the sampling module and polynomial multiplication, making it a prime target for profiling attacks with deep learning.

2.2.2 Module-Lattice DSA (ML-DSA)

Standard: ML-DSA, covered under FIPS 204 [26], specifies CRYSTALS-Dilithium, which also depends on the Module-LWE problem. Dilithium extends Kyber's polynomial operations to a digital signature context, requiring additional rejection sampling to ensure signature correctness. As Zhou et al. [16] demonstrated, these steps can leak when sampling and noise generation logic are unmasked or insufficiently randomized.

In practice, the signature generation flow in Dilithium includes intermediate rejection loops, polynomial decompositions, and hash-based commitments. Each of these adds points where EM or power analysis can correlate leaked side-channel traces with secret noise vectors. This is especially critical in real-time secure boot and RoT modules where Dilithium signatures must be verified and generated under strict time constraints.

Recent design guidance emphasizes the need for robust zeroization, multi-share masking, and compositional proofs that cover cross-module leakage [10, 27]. Because ML-DSA is likely to be embedded into SoCs and FPGAs at scale, it is crucial to formally verify that all masking operations remain resilient against advanced profiling attacks, especially when reused sampling hardware interacts with Kyber cores.

Countermeasure Context: Since ML-DSA is likely to be integrated into Silicon

Root of Trust (RoT) designs, the secure erasure (zeroization) process and masking for modular arithmetic must be robust [10, 27].

2.2.3 Stateless Hash-Based DSA (SLH-DSA)

Standard: FIPS 205 [41] standardizes SPHINCS+, a stateless hash-based signature scheme that stands out for its minimal reliance on algebraic structures. Instead, it relies entirely on hash function security — in theory, offering strong resilience against quantum attacks if the hash functions themselves remain unbroken [10].

The stateless design of SPHINCS+ removes the risk of leaking per-session states that could compromise the key, which is a notable strength compared to older hash-based stateful schemes like XMSS. However, practical side-channel attacks can still exploit the repeated Merkle tree traversals, leaf generation, and PRF calls. For instance, improper memory reuse or non-constant-time implementations can open pathways for attackers to reconstruct tree paths with simple CPA or EM probes.

A key insight from recent leakage analyses is that SPHINCS+ requires extensive buffer and stack management to ensure that no sensitive intermediate states survive beyond their useful lifetime. Novel defenses, such as runtime anomaly detection and ML-aware trace monitoring, are under investigation to guarantee that hash tree computations remain resistant to high-resolution power analysis [10, 17].

2.2.4 FFT-based NTRU Lattice DSA (FN-DSA)

Standard: FIPS 206 defines FN-DSA, the signature scheme based on the FALCON algorithm, which uses the NTRU lattice structure combined with the Fast Fourier Transform (FFT) [34]. This unique combination enables very compact signatures and high verification speeds but also introduces subtle vulnerabilities in the FFT and Gaussian sampling operations.

Recent works such as SHIFT SNARE [46] and single-trace leakage attacks [31] have highlighted how FFT stages can leak correlations through predictable signal patterns. This is especially problematic when Gaussian samplers fail to fully randomize ephemeral values, leaving residual information for ML-based profilers to exploit.

Countermeasure research for FN-DSA focuses on integrating hardened FFT implementations, rigorous randomness extractors, and formal proofs that guarantee low mutual information between ephemeral samples and the final signature output. Because FALCON is a leading candidate for use in constrained IoT settings, lightweight hardware countermeasures must be balanced against performance and latency constraints [10].

2.2.5 Hamming Quasi-Cyclic (HQC)

HQC remains a prominent code-based PQC candidate [24, 36]. Its security derives from the hardness of decoding quasi-cyclic random linear codes, a well-established NP-hard problem. Its structured error-correcting operations make it naturally resilient against some mathematical cryptanalysis but create unique side-channel challenges when syndrome decoding is implemented in hardware.

Recent studies, including key recovery attacks under chosen-ciphertext conditions [47], show that the decoding loop leaks distinguishable side-channel patterns, especially under high-resolution EM and power analysis. Notably, the quasi-cyclic nature introduces repeating patterns that are easier for deep learning profilers to detect and correlate across traces.

Emerging research focuses on robust masking for the decoding process, shuffling of matrix-vector multiplication steps, and verifying the correctness of error-correcting steps in hardware. HQC, while not yet standardized under FIPS, provides a crucial fallback

for institutions interested in diversifying their quantum-safe portfolios, and highlights the need for cross-family comparative SCA resilience [37].

2.3 Layered Taxonomy of FIPS PQC Standards and Leakage Vectors

Figure 1 shows a uniquely layered, color-coded diagram that distills the structural relationships across the five FIPS-standardized PQC families and their most critical leakage hotspots. This taxonomy serves as more than a pictorial aid — it is a conceptual scaffold that helps unify the entire survey's dual focus: understanding where side-channel leakage emerges in post-quantum implementations, and how machine learning can be leveraged to either exploit or mitigate these physical weaknesses.

At the root, the diagram begins with the newly formalized "FIPS PQC Standards" block, representing the high-level standardization of the quantum-safe primitives that will soon replace RSA and ECC in practical deployments [10, 25, 26, 41]. Each child branch colorfully fans out into a specific algorithm family — module-lattice, hash-based, FFT-based, or code-based — visually highlighting how distinct mathematical foundations naturally lead to different hardware modules and, consequently, different leakage modes. For example, the ML-KEM branch points directly to its unique reliance on the Number Theoretic Transform (NTT) and rejection sampling stages — modules that repeated empirical studies have shown to be susceptible to butterfly pattern leakage and ephemeral key reuse issues [15, 27].

What makes this visual especially compelling is its emphasis on **modular composability**: each child node's sub-block breaks down the precise operations that are known to leak, clarifying how an attacker or designer must think at the microarchitecture level. The ML-DSA node, for instance, calls out the interplay of noise sampling and modular arithmetic that can open subtle CPA or EM pathways if masking is insufficient [16]. The SLH-DSA branch stands apart in yellow, signaling the fundamentally different nature of stateless hash-based operations, while still acknowledging that Merkle tree traversal remains a persistent vulnerability if buffers are reused or not wiped securely [10].

This diagram also sets the stage for cross-family transferability analysis — by mapping the overlap in leakage modules, it becomes evident where profiling models trained on, say, Kyber's polynomial transforms could partially adapt to Dilithium's noise sampling steps, or how FFT-based weaknesses in Falcon share structural analogies with NTT-based ones in Kyber. This visual relationship is vital for the sections ahead, which will explore the practical viability of transfer learning and domain adaptation in real-world attacks.

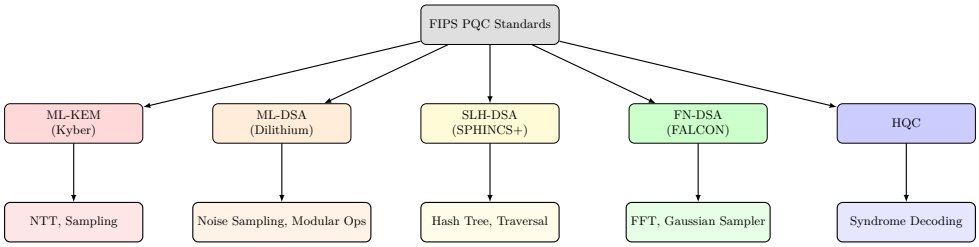


Figure 1: Layered taxonomy of FIPS-standardized PQC families with key modules and leakage vectors [10, 25–27, 41].

In essence, Figure 1 is not just an architectural chart — it is a blueprint for the next generation of resilient, formally verified implementations. It reminds us that while the underlying mathematics may promise quantum resistance, the physical realities of leakage through power, EM, or thermal channels demand a new era of holistic, ML-informed countermeasure design. This layered view becomes the anchor point for your

entire discussion on profiling accuracy, multi-channel fusion, and the need for explainable, composable hardware security guarantees.

3 Side-Channel Attacks (SCA)

Side-channel attacks (SCAs) exploit physical leakages—such as power, timing, or EM emissions—from cryptographic devices to extract secret information without breaking the underlying algorithm [6, 7]. Kocher's timing attack revealed that even small variations in RSA or DH operations can leak private keys [7], and power analysis has successfully recovered keys from otherwise secure ciphers [49].

These attacks are often stealthy and leave no trace [50], making them especially dangerous. A strong algorithm is not enough—leakage in implementation can compromise security. As a result, side-channel resistance has become a vital aspect of secure cryptographic design [49]. Figure 2 visualizes how an attacker exploits physical leakages to recover cryptographic keys.

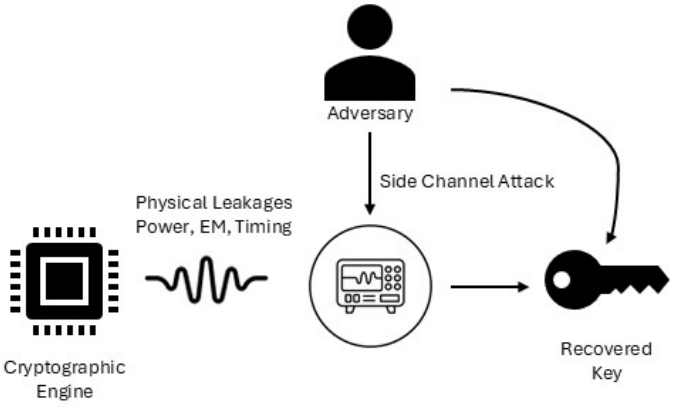


Figure 2: Illustration of a side-channel attack on a PQC device. The adversary collects power/EM leakage during cryptographic operations and recovers the secret key using CPA.

While post-quantum cryptography (PQC) focuses on quantum resistance, its implementations still run on classical hardware that leaks [18]. Early research shows that many PQC schemes, especially lattice-based ones, are vulnerable to similar SCAs as RSA or AES. For example, a single-trace attack on a masked lattice implementation succeeded by exploiting leakage in number-theoretic transforms [51].

In short, PQC does not inherently prevent SCAs—its complexity may even create new leakage points. Without strong protections, PQC systems risk being broken by classical SCAs [10]. Thus, post-quantum systems must be designed with side-channel resilience in mind to ensure practical security.

3.1 Classical Side-Channel Attacks

Classical SCAs exploit *implementation* leakage rather than a cipher's mathematics. Foundational work (e.g., Kocher's timing attack on RSA/DH) shows that small runtime variations can reveal secret material [7], while power-based methods demonstrate that even tamper-resistant chips leak enough information for key recovery [6].

A generic leakage model for the i^{th} operation is

$$L_i = f(K, M_i) + \eta_i, \tag{1}$$

where M_i is a known input (plaintext/ciphertext), K is the secret (or subkey), and η_i denotes measurement noise.

Table 3 shows the different methods of side channel analysis and how they are carried out.

Table 3: Overview of Common Side-Channel Attack Techniques

Attack Type	Description
Timing Attacks	Measure execution time T for chosen inputs; correlation between T and secret-dependent control flow (loops/table indices) leaks key bits. First demonstrated by Kocher against RSA/DH; defended by constant-time code [7].
Simple Power Analysis (SPA)	Inspect a single power waveform for operation patterns (e.g., square vs. multiply). Effective primarily on unprotected or leaky implementations.
Differential/Correlation Power Analysis (DPA/CPA)	Use many traces and a leakage model $L_i = f(K, M_i) + \eta_i$; for each key guess k, compute hypothetical leakage $H(k)$ (e.g., via Hamming weight) and apply difference-of-means or correlation. Typical data needs: tens to thousands of traces, SNR-dependent; even tamper-resistant chips can be vulnerable [6].
Electromagnetic Analysis (EMA)	Probe EM emissions instead of current draw. EM offers spatial selectivity and non-contact access; SEMA/DEMA/CPA apply to EM traces as in power analysis [52].
Cache/Micro-architectural Attacks	Exploit secret-dependent behavior in CPU caches/speculative units (e.g., Flush+Reload, Prime+Probe); avoid secret-dependent memory access in software crypto [53–55].
Template Attacks	Profile multivariate Gaussian leakage models on a clone device (known key), then match to target traces; often highly effective even against protected implementations.
Fault Attacks (e.g., DFA)	Induce faults via clock/voltage glitches, EM pulses, or lasers; algebraic relations between correct and faulty outputs reveal keys [56].

Timing Attacks: Measure the computation time of cryptographic operations. Variations in runtime (due to branches, loops, table lookups) can leak key bits [7]. For example, modular exponentiation may take longer when more exponent bits are 1, enabling recovery of Diffie–Hellman exponents or RSA factors [7]. Formally, with observed time $T(k, x)$ for key k and input x, attackers correlate T with key-dependent patterns; constant-time implementations mitigate this class.

Power Analysis (SPA/DPA/CPA): Monitor the device's power consumption over time. Simple Power Analysis (SPA) inspects individual traces for visible patterns (e.g., square vs. multiply in RSA). Differential Power Analysis (DPA) aggregates many traces with statistical tests. For a key guess k, one constructs a hypothetical leakage—often via a Hamming-weight model $\mathrm{HW}(v)$—and compares predicted vs. measured groups (difference-of-means) or uses correlation (CPA). Kocher showed that even tamper-resistant chips leak sufficient power variations for key recovery [6]. In practice, DPA/CPA recover symmetric or asymmetric keys using tens to thousands of traces, depending on SNR.

In CPA, the Pearson correlation between a hypothetical leakage vector $H(k)$ and observed traces L is

$$\rho(H(k), L) \;=\; \frac{\mathrm{Cov}(H(k), L)}{\sqrt{\mathrm{Var}(H(k))\;\mathrm{Var}(L)}}, \tag{2}$$

and the correct key guess k typically maximizes ρ (near-optimal under Gaussian noise).

Electromagnetic (EM) Analysis: Conceptually similar to power analysis, but probes EM emissions from the chip. Time-varying currents radiate fields that carry information comparable to power consumption, often with higher spatial resolution and non-invasive access. SEMA/DEMA mirror SPA/DPA on EM waveforms [52]. Specialized

probes can acquire leakage without contact; EM methods are routinely applied to secure microcontrollers and hardware accelerators.

Cache and Microarchitectural Attacks: Modern CPUs expose secret-dependent timing via caches, branch predictors, and speculation. Techniques such as Flush+Reload and Prime+Probe infer key bits from memory-access timing [53–55]. Although PQC on constrained hardware may use smaller caches, any data-dependent hardware behavior can act as a side channel [57].

Template Attacks: Profile a clone device to build multivariate leakage templates (often Gaussian) for key-dependent classes, then classify target traces against these templates. Introduced at CHES 2002 [40], template methods have defeated masking on AES (e.g., template-based DPA on 8-bit microcontrollers) and improved further via dimensionality reduction (PCA) for robust points-of-interest selection [58, 59]. For PQC, template techniques have been adapted to lattice KEMs: a voltage-template attack targets Kyber's modular polynomial subtraction, and cyclic-rotation templates recover secrets during Kyber decapsulation [60, 61].

Fault Injection: Fault attacks (e.g., clock/voltage glitches, EM pulses, laser) can be combined with side-channel methods. Precisely timed faults may force secret bits to propagate differently, aiding recovery. Differential Fault Analysis (DFA) has broken AES, RSA, and ECC by exploiting algebraic relations between correct and faulty outputs [56].

Other Channels (Acoustic, Optical, etc.): Some attacks exploit sound (acoustic emanations correlating with computation) or light (photonic emissions) [62, 63]. While less common, they underscore that any correlated emanation can be exploited.

Practical impact. These techniques are thoroughly practical: numerous works—including [6] through [63]—demonstrate full key recovery on unprotected microcontrollers with fewer than 10^3 traces. Even with protections, classical SCAs (especially CPA) can succeed unless masking, shuffling, constant-time execution, and EM/power-aware layout are enforced across abstraction layers. Given the low cost of equipment and broad attack surface, classical SCAs remain a critical threat vector for embedded cryptography, including PQC candidates.

3.2 Machine Learning-Based Side-Channel Attacks

Modern side-channel analysis increasingly leverages machine learning (ML) and deep learning (DL) to treat key recovery as supervised classification: traces t_i labeled by secret-dependent information (e.g., key byte/bit/coefficient) train a model f_θ to predict the label from t_i. In practice, the output layer often has 256 neurons (for an 8-bit class) with softmax to model $p(y \mid t)$. CNNs learn temporal patterns directly from traces; fully connected DNNs (MLPs) are effective when traces are well aligned, including in cross-device settings such as X-DeepSCA [64, 65].

Label design and leakage models. Common labels encode key-dependent intermediates, e.g., $y_i = \mathrm{HW}(\mathrm{Sbox}(P_i[j] \oplus K[j]))$ or $y_i = \mathrm{HW}(P_i[j] \oplus K[j])$ for byte j [66]; alternative designs use direct 256-class byte labels. Preprocessing typically includes denoising, normalization, and optional points-of-interest (PoI) selection [67]. During inference, scores across traces are aggregated (e.g., log-likelihood accumulation) to rank hypotheses and recover the key [66].

Trace efficiency and robustness. ML-based SCA can drastically reduce the number of required traces and improve robustness to desynchronization and device variation. For example, X-DeepSCA reports $\sim$10$\times$ fewer traces than classical CPA while improving cross-device generalization [64]. Hoang's CNN (CNNC) recovers a Kyber secret coefficient with $\sim$ 50 traces [68]. Rezaeezade uses an ensemble of five networks to recover a protected Ascon key with $<$3,000 traces [69]. These results illustrate that ML can exploit high-dimensional leakage that evades traditional statistics, especially when combined with normalization,

augmentation, and ensembling.

3.2.1 Overview of ML Approaches

Formally, ML-SCA minimizes a supervised loss over N labeled traces:

$$\min_{\theta} \sum_{i=1}^{N} \ell(f_\theta(t_i),\, y_i)\,, \tag{3}$$

typically cross-entropy; at inference,

$$\hat{y} = \arg\max_{y} f_\theta(t). \tag{4}$$

In key recovery, per-trace posteriors are aggregated across many traces to compute a key rank (e.g., guessing entropy).

Figure 3 shows a typical ML-SCA pipeline.

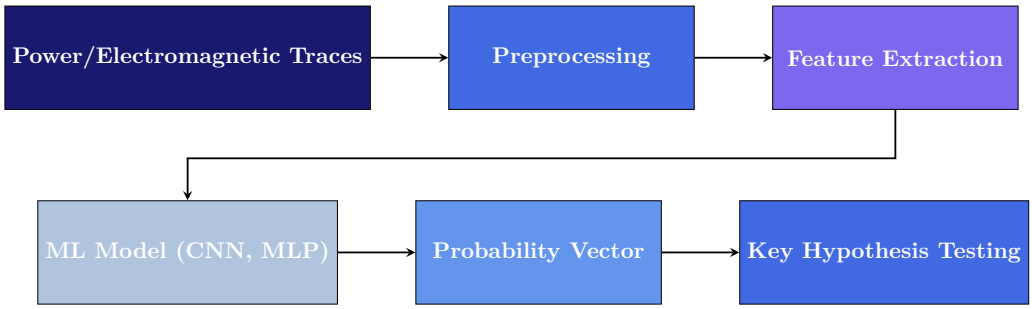


Figure 3: ML-SCA attack pipeline: traces are processed (preprocess/feature extraction) and fed to ML models to score key hypotheses.

Table 4: Overview of Machine-Learning Approaches for SCA

Approach	Description / Typical use / Notable refs
Convolutional Neural Networks (CNNs)	Learn local temporal patterns; robust to small misalignment and noise. Demonstrated end-to-end Kyber recovery with $\sim$ 50 traces [68].
Fully-Connected DNNs (MLPs)	Effective on well-aligned traces; used in cross-device classifiers (X-DeepSCA/DeepXCA) with domain normalization to improve generalization and reduce traces vs. CPA [64,65].
Signal Decomposition + DL-SCADS	Wavelet/EMD decomposition isolates frequency-localized features; ensembles over decomposed inputs improve robustness and trace efficiency on AES/PQC [70].
Ensemble Learning	Combine CNNs/MLPs trained on diverse inputs/subsets; boosts accuracy and stability (e.g., Ascon with $<$3,000 traces) [69,70].
Transfer / Domain Adaptation	Reuse models across keys/devices; normalize domain shifts to mitigate inter-device variance (X-DeepSCA/DeepXCA) and achieve $\sim$10$\times$ trace reduction vs. CPA [64,65].
Other ML (SVM, RF, Autoencoders)	Lightweight or auxiliary roles (baselines, denoising, pretraining); surveys summarize competitive hybrids and best practices [71].

In summary, ML-based SCA replaces manual leakage modeling with data-driven inference and often outperforms DPA/CPA—especially under masking, desynchronization, and device variation—through architectures such as CNNs/MLPs, decomposition-driven ensembles (DL-SCADS), and cross-device frameworks (X-DeepSCA/DeepXCA) [64,65,68–71].

4 Side-Channel Attacks on PQC Implementations

Extensive studies have shown that ML-based SCAs achieve superior trace efficiency compared to classical methods. CNN-based attacks can recover secret keys with orders of magnitude fewer traces than CPA under identical experimental conditions. Additionally, ML models exhibit remarkable resilience to noise and desynchronization, making them effective even when countermeasures such as shuffling or clock jitter are employed [72].

Machine learning–based SCAs have been applied to many candidate PQC schemes, demonstrating their broad impact. Table5 and Table6 shows an extensive study of different ML based SCAs on Different PQC algorithms where Table5 demontrates attack scenario and Table6 elaborates the key features and limitations of those attack models.

Table 5: Overview of ML and Classical Side-Channel Attacks on PQC: Success, Traces, and Scenario

Attack Method (Case Study)	**Success Rate**	**Traces Required**	**Attack Scenario**
CNN-based profiled power attack on unmasked Kyber [68]	∼100%	∼50 traces	Kyber decapsulation on ARM Cortex-M4 without masking or shuffling with Ciphertext known.
Non-profiled attack exploiting shuffle on masked Kyber [28]	∼100%	∼4,608 traces	Masked and shuffled Kyber decapsulation; targets index permutation leakage.
Recursive DNN on 5th-order masked Kyber [73]	>99% (bitwise)	1 trace per bit	Recursive deep learning model across increasing masking orders.
MLP profiled attack on unmasked Saber [74]	∼100%	1 trace	Message decoding leakage in Saber decapsulation (no countermeasures).
MLP attack on masked & shuffled Saber [33]	∼100%	1–few traces	Side-channel on first-order masked and shuffled Saber decapsulation.
Hybrid LR + CNN template attack on Dilithium [31]	92–100%	1–2 traces	Leakage from polynomial addition in Dilithium-II signing process.
CPA on unprotected Dilithium [29]	100%	157 traces	Power analysis during NTT/multiplication in software implementation.
LDA-based SCA on HQC [47]	100%	∼20k traces	Oracle-based decoding leakage using chosen ciphertexts in HQC decapsulation.
DPA on “constant-time” HQC [37]	∼99%	∼83 traces	Microarchitectural leakage using repeated decryptions with SIMD disabled.

Kyber (LWE/KEM): Multiple studies have demonstrated side-channel vulnerabilities in Kyber. Ravi used electromagnetic (EM) side-channel leakage to mount chosen-ciphertext attacks on CCA-secure lattice KEMs, including Kyber [12]. Hoang [68] employed a convolutional neural network (CNNC) to attack an unprotected Kyber-512 implementation on an ARM Cortex-M processor. Impressively, they recovered a single secret polynomial coefficient with only ~50 power traces. The results confirm that both classical and ML-based SCAs can effectively breach Kyber when implemented without robust protections.

Saber (LWR/KEM): Ngo conducted the first successful SCA on a masked Saber

Table 6: Key Technical Insights and Limitations of ML and Classical SCA Attacks on PQC

Attack Method (Case Study)	**Key Features**	**Limitations**
CNN-based Kyber	Learns 12-bit coefficient leakage patterns using 1D CNN with ciphertext as input.	Requires known ciphertext; fails under masking; untested on hardened implementations.
Shuffle-exploiting attack on masked Kyber	Uses ciphertext rotation to defeat Fisher-Yates shuffle; 13× trace reduction vs. prior art.	Requires many carefully crafted decaps; assumes control over ciphertext and shuffle logic.
Recursive DNN on masked Kyber	Recursive model construction enables training on increasing mask orders.	Bit-by-bit recovery; depends on same masking configuration for profiling and test.
MLP on unmasked Saber	MLP + SOST analysis recovers message bits from strong leakage points.	Only works on unprotected implementation; recovers session key only.
MLP on masked & shuffled Saber	Residual leakage in conversion step exploited with single-trace profiling.	Profiling required; attack fails with higher-order masking or stronger shuffling.
Hybrid LR + CNN on Dilithium	Combines y via linear regression and cs_1 via CNN; full key reconstructed by solving $z = y + cs_1$.	Profiling-intensive; sensitive to noise and unprotected code; fails under masking.
CPA on unprotected Dilithium	Reduces key space via correlation analysis and conservative guessing strategy.	Requires known message or randomness; vulnerable to masking/shuffling.
LDA-based HQC attack	Detects decoding failure structure using LDA classifier; support bits extracted.	High trace requirement; assumes consistent decoder behavior and no masking.
DPA on "constant-time" HQC	Exploits hardware-level microarchitectural leakage even in constant-time code.	Depends on repeated ciphertext input; fixed by SIMD vectorization and better masking.

KEM [33]. Using deep learning techniques on just 16 power traces, they recovered both the session key and the long-term secret key. Their approach bypassed masking by learning leakage end-to-end, without explicitly extracting mask values. This work highlights that even first-order masked implementations of Saber remain vulnerable. Traditional DPA attacks on Saber's Number Theoretic Transform (NTT) have also shown effectiveness in unmasked versions.

Dilithium (LWE/Signature): Wang profiled side-channel leakage during secret key unpacking in Dilithium-2 using neural networks [30]. From a single power trace, they recovered approximately half of the secret polynomial coefficients, solving the rest through linear algebra. Their method achieved ~9% success from a single trace on a different device, reaching nearly 100% with just a few traces. The attack exploited leakage from matrix-vector multiplication, a core component of Dilithium's implementation.

Falcon (NTRU/Signature): A 2025 study [46] revealed critical power leakage during Falcon's key generation. Analyzing power consumption while sampling secret polynomials, they recovered entire private keys with near-perfect accuracy using a single trace. The technique utilized Gaussian modeling to distinguish between coefficient values, exposing Falcon's vulnerability in real-world ARM Cortex-M4 systems. Despite Falcon's limited adoption due to efficiency issues, this attack underscores the risks in signature schemes relying on arithmetic transforms.

Other Lattice Schemes: Beyond the primary NIST finalists, various other lattice-

based proposals have suffered SCAs. Schemes like NTRU, NewHope, and FrodoKEM have been targeted by generic DPA on polynomial multiplication [32], [11] . Furthermore, alternative schemes such as SIKE (supersingular isogeny-based) exhibited critical leakage: simple power analysis of scalar multiplication revealed secrets, contributing to SIKE's deprecation [75].

Across all these attacks, a consistent theme emerges: implementation artifacts such as rejection sampling, polynomial loops, or error-correction steps introduce leakages that attackers exploit. Machine learning techniques amplify these vulnerabilities, extracting useful features from minuscule leakages. In totality, ML-enhanced SCAs have compromised both unprotected and protected versions of nearly all leading PQC candidates. This emphasizes the urgency of integrating robust countermeasures—such as high-order masking, hiding, and noise injection—into future PQC hardware and software designs.

4.1 Effectiveness and Impact of SCAs on PQC Implementations

While post-quantum cryptographic (PQC) schemes are designed to resist quantum and classical mathematical cryptanalysis, their physical instantiations remain vulnerable to implementation-level attacks—especially side-channel attacks (SCAs). Unlike purely algorithmic exploits, SCAs leverage unintended physical leakages (power, EM, timing) to recover secrets from otherwise secure algorithms.

Classical Power and Timing Attacks. Simple Power Analysis (SPA) and Correlation Power Analysis (CPA) have repeatedly proven effective against real PQC prototypes. In lattice-based schemes (e.g., CRYSTALS-Kyber, Saber, and Dilithium), CPA/related attacks have recovered key material by exploiting leakage in polynomial multiplication, modular reductions, and particularly the Number Theoretic Transform (NTT) datapath [23, 30, 76]. Moreover, hardware and protocol-level designs can enable single-trace or horizontal attacks that combine many key-dependent operations within one trace [14]. Even when masking is applied, weak orders or flawed implementations can leave exploitable residual leakage.

SPA against decoding/logical flow. SPA targets visible variations in traces due to branching and table lookups. Code-based schemes such as HQC and Classic McEliece rely on decoding routines (e.g., RM/RS or BCH), which often include secret-dependent control flow or memory access; leakage from these paths has been used to reveal key-dependent decoding behavior [77, 78]. Timing-cache effects can likewise be exploited when implementations fail to enforce constant-time behavior [79].

Fault Injection Attacks (FIA). Faults induced by voltage/clock glitches or EM pulses can corrupt computations in ways that expose secrets. Classical differential fault analysis shows that even a single fault can be devastating in symmetric cryptography [80]; analogous techniques and threat models apply to PQC decapsulation/decoding paths (e.g., to perturb syndrome checks or reconstruction steps), making FIA an additional concern for code- and lattice-based designs.

Practical Amplifiers of Leakage. Several structural properties of PQC algorithms tend to amplify SCA risk: highly regular and repetitive arithmetic (polynomial or matrix multiplications) creates consistent leakage over large key sizes; discrete sampling for noise/error terms can leak via non-constant-time samplers; and rejection sampling (e.g., in Dilithium) introduces secret-dependent control flow if not implemented carefully [14, 76].

Real-World Evidence. Published attacks on embedded/FPGAs demonstrate practical key recovery with modest trace counts—including non-profiled CPA on Dilithium and single-trace/horizontal strategies on lattice KEMs—underscoring that these are not merely theoretical issues [14, 30, 76].

Emerging Trends. Machine-learning (ML) profiling has strengthened SCAs by learning complex leakage distributions, often undermining low-order masking and desynchronization. Surveys and recent empirical work highlight both the effectiveness and the

portability challenges of deep learning–based SCAs [81, 82].

Implication for PQC Standardization. Physical leakage channels that plagued RSA/AES persist for PQC—and can even be amplified by regular algorithmic structures (e.g., NTT datapaths, sampling and rejection-sampling logic) unless implementations enforce strict constant-time behavior and leakage-resilient design [83, 84]. Without rigorous, verified countermeasures, real-world deployments can be compromised despite strong asymptotic proofs. Consequently, implementation-aware standards and test methodologies that explicitly cover side-channel and non-invasive fault resistance must complement algorithm standards—spanning algorithm selection, hardware mapping, and system validation [85–90].

Consequently, the demonstrated effectiveness of SCAs against PQC underscores the need for comprehensive, implementation-aware standardization that bakes in side-channel (and fault) resilience across the lifecycle—from *algorithm selection* (e.g., the new FIPS standards for ML-KEM, ML-DSA, and SLH-DSA) through *implementation choices* such as thresholdized designs, to *module validation and non-invasive testing* using established metrics and guidance [85–92]. This paper builds on that evidence and examines concrete countermeasure strategies under these standards-aligned practices to ensure robust physical-layer defenses for real-world quantum-safe security [93].

5 Countermeasures

To defend against side-channel attacks (SCAs), a wide array of countermeasures has been developed, ranging from classical approaches to machine learning (ML)-inspired defenses. In the context of post-quantum cryptography (PQC), these countermeasures must not only address traditional vulnerabilities but also withstand sophisticated profiling attacks enabled by modern ML techniques. This section surveys and compares both non-ML and ML-based countermeasure strategies, highlighting their respective strengths and limitations.

5.1 Conventional Countermeasures

Classical countermeasures against side-channel attacks (SCAs) encompass a combination of software and hardware-based defenses aimed at disrupting or eliminating exploitable leakages from cryptographic implementations. On the software side, **masking** techniques remain a cornerstone approach: sensitive intermediate values are split into multiple statistically independent random shares, such that any individual share, if observed, reveals no information about the underlying secret [5]. These shares are combined only at the final stage of computation, significantly raising the difficulty of first-order attacks.

Constant-time design ensures that both control flow and memory access patterns remain independent of secret data [9], thereby mitigating timing-based SCAs. This includes eliminating secret-dependent branches, avoiding table lookups indexed by key material, and ensuring that compiler optimizations do not reintroduce leakage channels. In practice, constant-time constraints are applied across the entire cryptographic pipeline, from arithmetic operations to random number generation.

Hiding strategies reduce the signal-to-noise ratio (SNR) of measured leakages by making power, electromagnetic (EM), or timing characteristics less correlated with secret data [94]. For example, power equalization methods pad operations with dummy instructions or adjust execution so that each cycle consumes approximately the same amount of power. Similarly, EM noise generators can mask emissions with random fluctuations, and random process shuffling can misalign leakage patterns across traces.

Randomness-based techniques add variability to execution through random delays, dummy operations, or shuffled computation sequences [94]. These methods force adversaries to collect more traces and to apply complex realignment techniques before statistical attacks,

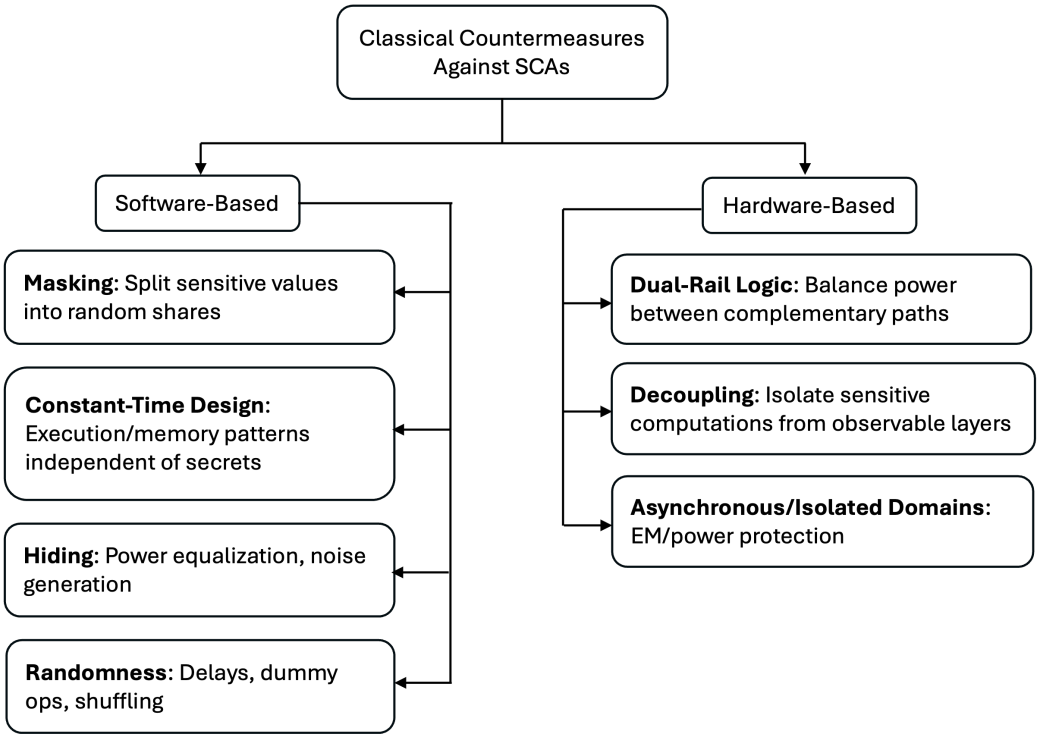


Figure 4: Classical countermeasures against SCAs: software-based and hardware-based

thereby increasing both the cost and time required for a successful attack.

Complementing these are **hardware-based countermeasures**. **Dual-rail logic** is a well-known design style in which each logical value is represented by two complementary signals that switch in a balanced fashion, ideally producing identical power consumption regardless of the processed bit [95]. This approach reduces data-dependent variations at the transistor switching level. **Decoupling** techniques physically or electrically isolate sensitive computation blocks from I/O and measurement points, limiting leakage propagation to accessible domains. Finally, **asynchronous logic** and **isolated clock/power domains** disrupt the correlation between computation and measurable physical signals by removing a global clock reference and reducing simultaneous switching activity [96].

Recent studies have highlighted that even these countermeasures require rigorous validation against sophisticated attacks [96]. Standardized evaluation frameworks, such as Test Vector Leakage Assessment (TVLA), are now widely adopted to quantify the effectiveness of these defenses in both classical and post-quantum cryptographic contexts.

5.2 ML-Based Countermeasures Against Side-Channel Attacks

The rise of ML-driven SCAs has motivated the development of new countermeasures that leverage adversarial principles or exploit the attacker's learning process. Adversarial perturbations intentionally inject minimal noise into side-channel traces to mislead ML classifiers without significantly degrading system performance [97]. Noise pattern learning adapts runtime leakage characteristics to mimic random or misleading patterns, making it harder for deep learning models to distinguish meaningful features [98].

Trace obfuscation using generative models, such as Generative Adversarial Networks (GANs), represents a promising frontier. These models can dynamically transform leakage traces into synthetic distributions that preserve functionality while masking leakage

points [99]. ML-aware masking techniques combine classical randomization with learned noise models to optimize leakage suppression specifically against profiling attacks. Signal decorrelation methods further aim to remove statistical dependencies between sensitive intermediates and observed leakages, targeting both classical and ML-based analyses.

Table 7: Comparison of Classical and ML-Based Countermeasures for PQC Side-Channel Security

Countermeasure Type	**Strengths**	**Limitations**	**Recommended Strategy**
Classical Defenses (e.g., masking, constant-time execution)	Proven effectiveness against first-order side-channel attacks. Well-understood and widely supported in existing toolchains.	Reduced effectiveness against higher-order and ML-enhanced attacks. Overhead grows significantly with higher protection orders.	Use as a baseline protection layer and complement with ML-aware techniques for higher-order resilience [30].
ML-Based Defenses (e.g., leakage classification, adaptive obfuscation)	Promising in resource-constrained environments where traditional redundancy is costly. Can dynamically adapt to detected leakage patterns.	Higher implementation complexity. Potential issues with model transferability and stability under varied conditions. Risk of performance degradation if not tuned.	Integrate with classical masking and randomized execution to form hybrid, ML-aware countermeasures [72].

The recent surge in machine learning (ML)-enhanced side-channel attacks (SCAs) has motivated the exploration of defensive strategies that directly exploit or disrupt an attacker's learning process. Unlike classical countermeasures — such as masking or constant-time execution — ML-based countermeasures are designed to mislead, obfuscate, or adaptively thwart profiling models that infer secrets from complex leakage distributions.

Adversarial Perturbations. Inspired by adversarial examples in computer vision, one emerging technique injects carefully crafted perturbations into side-channel leakage traces to intentionally fool a trained classifier [97]. This approach modifies leakage in ways imperceptible to functional correctness but impactful enough to misclassify sensitive operations when analyzed by a deep neural network (DNN). Such perturbations can be statically pre-computed or adaptively generated during execution, although balancing minimal performance impact remains challenging.

Generative Trace Obfuscation. Generative Adversarial Networks (GANs) offer another promising defensive paradigm [99]. Here, a generator network learns to produce synthetic leakage traces that closely resemble real traces in distribution but break the statistical correlation between secret-dependent operations and observable signals. A discriminator network iteratively refines this process, forcing the attacker's profiling model to generalize poorly. This adversarial framework aims to provide high entropy in the leakage without altering the device's logical output.

Noise Pattern Learning. Instead of adding constant noise, advanced techniques can learn optimal noise patterns that mimic legitimate operations but randomize the trace signatures. This approach enhances classical hiding by introducing structured randomness that confounds feature extraction steps in CNN-based SCAs [98]. Learned noise profiles can be periodically updated to remain resilient against attackers who attempt to re-profile with larger datasets.

ML-Aware Masking and Runtime Adaptation. Combining classical masking with dynamic ML feedback has emerged as a hybrid defense trend. For example, runtime monitors can employ lightweight ML models to detect anomalous leakage patterns or evidence of probing, and then adapt masking orders, polynomial sampling seeds, or operation scheduling on the fly. This effectively forces the attacker to re-collect profiling data under continuously shifting leakage conditions.

Signal Decorrelation and Feature Disentanglement. Another novel direction targets the core statistical dependencies that ML models rely on. Signal decorrelation

methods aim to orthogonalize or disguise the relationship between intermediate sensitive variables and measurable side-channel signals [30]. Feature disentanglement techniques attempt to remove low-level features (such as coefficient leakage in polynomial operations) from power and EM signatures without significantly degrading throughput.

Emerging Limitations and Research Challenges. Despite their promise, ML-based countermeasures face practical and theoretical challenges. First, integrating adaptive perturbation or generative obfuscation into embedded or low-power devices requires additional hardware resources and may induce latency overhead. Second, these techniques assume that the defender's model generalizes better than the attacker's — an assumption that breaks down when attackers have access to transfer learning or significantly larger training sets. Third, rigorous formal proofs for the security guarantees of ML-based defenses remain scarce; performance in one profiling scenario may not generalize to diverse architectures or leakage models.

Nevertheless, the field is rapidly advancing toward hybrid frameworks that combine classical countermeasures with ML-aware obfuscation and runtime adaptation. Such multi-layered defenses may offer the best balance between leakage suppression, implementation efficiency, and resilience to next-generation profiling SCAs. Future research must address the cost-benefit trade-offs of these dynamic defenses and develop standardized benchmarks for evaluating their real-world effectiveness.

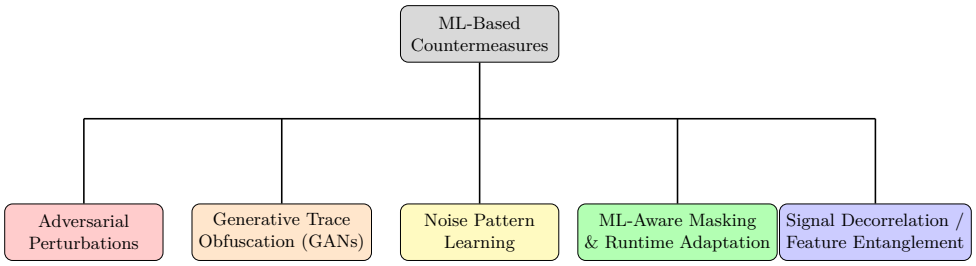


Figure 5: Taxonomy of ML-Based Countermeasures for Side-Channel Leakage: the root concept branches to five evenly spaced strategies.

Table 8: Comparison of ML-Based Countermeasures Against SCAs

Method	**Key Principle**	**Strengths**	**Limitations**
Adversarial Perturbations	Inject minimal noise to mislead profiling models	Low overhead for some architectures; works without hardware redesign	Requires precise noise design; may not scale under adaptive re-profiling
Generative Trace Obfuscation (GANs)	Learn to produce realistic but misleading traces	Breaks statistical correlation; theoretically robust against template attacks	Computationally intensive to generate in real time; hardware cost
Noise Pattern Learning	Learn optimal noise shapes to mask leakage	Adapts to leakage structure; enhances hiding efficiency	Needs periodic retraining; model may overfit attacker profiles
ML-Aware Masking / Runtime Adaptation	Dynamically adjust masking or sampling seeds based on leakage detection	Makes leakage unpredictable for profiling	Complexity in monitoring; possible performance penalties
Signal Decorrelation	Remove statistical link between sensitive intermediates and leakage	Complements other methods; math-based leakage suppression	Difficult to prove universal security; hardware-level implementation may be costly

6 Case for HQC Evaluation

The Hamming Quasi-Cyclic (HQC) scheme represents a compelling candidate for evaluating side-channel vulnerabilities and countermeasures within post-quantum cryptography (PQC). HQC combines structured code-based encryption with error-correcting mechanisms, resulting in a favorable trade-off between key size, computational efficiency, and security. Unlike other code-based schemes such as Classic McEliece, HQC offers significantly smaller key and ciphertext sizes, making it more amenable to implementation in resource-constrained embedded systems [24].

Several factors render HQC an ideal subject for side-channel analysis. Its decoding procedures, based on syndrome computation and sparse vector recovery, involve conditional operations and iterative algorithms that are prone to timing and power leakage. Operations such as parity checks, error vector reconstruction, and syndrome validation frequently exhibit data-dependent behavior, introducing potential leakage points exploitable by both classical and machine learning-based side-channel attacks [13].

Among the critical leakage vectors in HQC, decoding operations constitute the most prominent threat. Specifically, the conditional branching involved in error correction and the memory access patterns during vector manipulations provide fertile grounds for leakage analysis. Syndrome processing, where partial knowledge about error positions is accumulated, presents additional opportunities for side-channel exploitation, particularly when standard countermeasures like masking are absent or insufficiently deployed [35].

To facilitate experimental evaluation of HQC's side-channel resistance, the ChipWhisperer-Lite platform emerges as an effective solution. ChipWhisperer-Lite offers a low-cost, open-source environment for conducting precise power analysis, electromagnetic analysis, and fault injection experiments. Its compatibility with custom FPGA designs and microcontroller firmware enables faithful, reproducible implementation of HQC's critical modules, including encapsulation, decapsulation, and syndrome decoding routines. Furthermore, the platform's integration with Python scripting allows automated trace collection and machine learning analysis pipelines, making it particularly suited for studying both classical and ML-based SCAs against HQC [100].

7 Future Directions

As side-channel threats continue to evolve, the research community must develop comprehensive frameworks to evaluate and strengthen the resilience of post-quantum cryptographic (PQC) implementations. Designing an end-to-end machine learning (ML)-based side-channel attack (SCA) evaluation framework constitutes a critical next step. Such a framework should integrate trace collection, preprocessing, feature extraction, attack modeling, and defense evaluation into a unified pipeline. This would enable standardized benchmarking of the effectiveness of different countermeasures against a diverse set of adversarial models, ranging from classical correlation power analysis to deep learning-based profiling attacks [30, 72].

Benchmarking machine learning versus classical attacks on PQC implementations remains a crucial and underexplored avenue. Comparative studies can reveal the conditions under which ML-based profiling attacks substantially outperform classical techniques and identify implementation features that exacerbate or mitigate vulnerability. Specifically, controlled experiments evaluating different countermeasures across varying noise levels, device types, and leakage models can yield valuable insights into defense robustness under realistic threat models [39, 101].

There is also an urgent need for standardized side-channel resilient PQC benchmarks. Current PQC standardization efforts by NIST primarily focus on mathematical security and functional correctness, with relatively limited emphasis on physical attack resilience.

Establishing a standardized suite of protected implementations, associated leakage models, and evaluation protocols would facilitate more rigorous and comparable side-channel evaluations across the cryptographic community. Such benchmarks would mirror efforts like ASCAD for AES and stimulate the development of inherently secure PQC designs [102].

Finally, several open problems remain regarding real-world deployment of side-channel resilient PQC. Key challenges include balancing the trade-offs between performance, energy consumption, and leakage resistance; ensuring the scalability of countermeasures across diverse hardware platforms; and adapting defensive strategies to evolving machine learning-based attack paradigms. Furthermore, the development of practical tools for automated leakage detection, profiling resistance certification, and leakage-aware optimization will be pivotal for future-proofing PQC deployments against both classical and intelligent adversaries.

8 Conclusions

This paper has presented a comprehensive perspective on the side-channel vulnerabilities of post-quantum cryptographic (PQC) implementations, highlighting the evolving threat landscape posed by both classical and machine learning (ML)-enhanced side-channel attacks (SCAs). The importance of post-quantum resilience was discussed within the context of quantum computing advancements, motivating the need for robust implementation-level protections beyond mathematical security proofs.

Key insights include the identification of structured computations within PQC schemes, such as decoding and syndrome processing, as critical leakage points. The review demonstrated that while classical SCAs such as Simple Power Analysis (SPA) and Correlation Power Analysis (CPA) remain potent under certain conditions, the integration of ML techniques has significantly elevated the adversarial capabilities, particularly against protected or noisy implementations. Case studies on Kyber, Dilithium, and HQC confirmed the effectiveness of profiling attacks leveraging deep learning architectures.

The findings reinforce the necessity for joint analysis of attacks and defenses from both classical and ML-based perspectives. Traditional countermeasures, including masking and hiding, while essential, must be augmented with emerging strategies such as adversarial perturbations and trace obfuscation to withstand intelligent attacks. A hybrid defensive architecture that combines classical hardening techniques with ML-aware adaptations emerges as the most viable pathway for securing future PQC deployments.

Finally, the paper emphasizes the importance of hardware-aware PQC implementations. The physical realization of cryptographic algorithms introduces complexities that must be proactively addressed through comprehensive side-channel evaluations, standardized benchmarking, and the adoption of leakage-resilient design principles. As PQC transitions from research to widespread adoption, integrating physical security considerations into the design lifecycle will be paramount for achieving trustworthy quantum-safe cryptography.

References

[1] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994, pp. 124–134.

[2] L. Grover, "A fast quantum mechanical algorithm for database search," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 212–219, 1996.

[3] N. I. of Standards and T. (NIST), "Post-quantum cryptography standardization," 2022. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography

[4] L. Chen, S. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, "Report on post-quantum cryptography," *US Department of Commerce, NIST*, 2016.

[5] S. Mangard, E. Oswald, and T. Popp, "Power analysis attacks: Revealing the secrets of smart cards," in *Springer US*, 2007.

[6] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO'99*. Springer, 1999, pp. 388–397.

[7] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1109. Springer, 1996, pp. 104–113.

[8] S. Bhasin and S. Guilley, "Hardware attacks and countermeasures: A survey," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1271–1283, 2014.

[9] J. Coron, P. Kocher, and D. Naccache, "Resistance against differential power analysis for elliptic curve cryptosystems," in *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, 1999, pp. 292–302.

[10] PQShield, "Quantum threat: The first nist post-quantum cryptographic standards," https://www.pqshield.com/whitepapers, 2022.

[11] C. Mujdei, A. Beckers, J. M. B. Mera, A. Karmakar, L. Wouters, and I. Verbauwhede, "Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication," Cryptology ePrint Archive, Paper 2022/474, 2022. [Online]. Available: https://eprint.iacr.org/2022/474

[12] P. Ravi, S. Sinha Roy, A. Chattopadhyay, and S. Bhasin, "Generic side-channel attacks on cca-secure lattice-based pke and kems," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 3, p. 307–335, Jun. 2020. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8592

[13] M. Bronchain *et al.*, "Message recovery and full key recovery side-channel attacks on pqc kems," in *CHES 2020*. Springer, 2020, pp. 192–211.

[14] T. Oder, T. Schneider, and T. Pöppelmann, "Practical cca2-secure and masked ring-lwe implementations," IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES), 2018, https://tches.iacr.org/index.php/TCHES/article/view/7347.

[15] Y. Park, D. Lee, and S. Hong, "A side-channel attack on a hardware implementation of crystals-kyber," in *Proc. IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2021, pp. 55–64.

[16] H. Zhou, Y. Liu, and W. Zhang, "Deep learning profiling attacks on dilithium hardware," in *Proc. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2022, pp. 31–39.

[17] H. Zhou, Y. Liu, and L. Wang, "A deep learning side-channel analysis of saber," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1231–1242, 2023.

[18] C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt, "Differential power analysis of a mceliece cryptosystem," in *Applied Cryptography and Network Security*, T. Malkin, V. Kolesnikov, A. B. Lewko, and M. Polychronakis, Eds. Cham: Springer International Publishing, 2015, pp. 538–556.

[19] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," *CHES 2004*, pp. 16–29, 2004.

[20] S. T. Ahmed, A. A. Ahmed, A. Annamalai, and M. F. Chouikha, "A scalable and energy-efficient lorawan-based geofencing system for remote monitoring of vulnerable communities," *IEEE Access*, vol. 12, pp. 48 540–48 554, 2024.

[21] S. T. Ahmed, A. A. Ahmed, A. Annamalai, and M. Chouikha, "Efficient rebroadcast location-unaware protocol for lorawan mesh networks in the iot domain," in *2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2024, pp. 104–110.

[22] S. T. Ahmed, A. Annamalai, and M. Chouikha, "Strengthening LoRaWAN Servers: A Comprehensive Update with AES Encryption and Grafana Mapping Solutions," *Advances in Science, Technology and Engineering Systems Journal*, vol. 9, no. 1, pp. 33–41, 2024.

[23] X. Ji, S. Ahmed, and A. Annamalai, "In-depth correlation power analysis attacks on a hardware implementation of crystals-kyber," Cryptology ePrint Archive, Report 2025/357, 2025, https://eprint.iacr.org/2025/357.

[24] P. Gaborit, R. Misoczki, N. Sendrier, and J. Tillich, "Hqc: Hamming quasi-cyclic kem," NIST PQC Submission Specification, 2021, https://pqc-hqc.org/.

[25] National Institute of Standards and Technology (NIST), "Fips 203: Module-lattice-based key-encapsulation mechanism (ml-kem)," Federal Information Processing Standard, 2024, available: https://csrc.nist.gov/publications/detail/fips/203/final.

[26] ——, "Fips 204: Module-lattice-based digital signature algorithm (ml-dsa)," Federal Information Processing Standard, 2024, available: https://csrc.nist.gov/publications/detail/fips/204/final.

[27] N. Kamucheka, A. Ghosh, and A. Annamalai, "Kpqbench: Performance and implementation security analysis of nist pqc round 1 candidates," in *Proc. IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2021, pp. 45–54.

[28] L. Backlund, K. Ngo, J. Gärtner, and E. Dubrova, "Secret key recovery attacks on masked and shuffled implementations of CRYSTALS-kyber and saber," Cryptology ePrint Archive, Paper 2022/1692, 2022. [Online]. Available: https://eprint.iacr.org/2022/1692

[29] H. Wang, Y. Gao, Y. Liu, Q. Zhang, and Y. Zhou, "In-depth correlation power analysis attacks on a hardware implementation of CRYSTALS-dilithium," Cryptology ePrint Archive, Paper 2023/1891, 2023. [Online]. Available: https://eprint.iacr.org/2023/1891

[30] H. Wang, H. Kim, and A. Saha, "Deep learning-based profiling attacks on crystals-dilithium," in *Proc. International Conference on Security, Privacy and Applied Cryptographic Engineering (SPACE)*, 2024, pp. 115–133.

[31] Z. Qiao, Y. Liu, Y. Zhou, Y. Zhao, and S. Chen, "Single trace is all it takes: Efficient side-channel attack on dilithium," Cryptology ePrint Archive, Paper 2024/512, 2024. [Online]. Available: https://eprint.iacr.org/2024/512

[32] A. Askeland and S. Rønjom, "A side-channel assisted attack on NTRU," Cryptology ePrint Archive, Paper 2021/790, 2021. [Online]. Available: https://eprint.iacr.org/2021/790

[33] K. Ngo, E. Dubrova, Q. Guo, and T. Johansson, "A side-channel attack on a masked ind-cca secure saber kem implementation," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, no. 4, p. 676–707, Aug. 2021. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/9079

[34] National Institute of Standards and Technology (NIST), "Fips 206: Fft (fast-fourier transform) ntru lattice digital signature algorithm (fn-dsa)," Federal Information Processing Standard, 2024, available: https://csrc.nist.gov/publications/detail/fips/206/final.

[35] P. Gaborit, J. Murat, O. Ruatta, and G. Zémor, "Hqc: A hamming quasi-cyclic public key encryption scheme," in *International Conference on Cryptology and Information Security*. Springer, 2017, pp. 118–135.

[36] G. Goy, A. Loiseau, and P. Gaborit, *A New Key Recovery Side-Channel Attack on HQC with Chosen Ciphertext*. Springer, 2022, pp. 353–371.

[37] C. Nugier, N. Maillet, V. Migliore, and J.-C. Deneuville, "Key recovery from side-channel power analysis attacks on non-simd hqc decryption." Springer-Verlag, 2025.

[38] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Applied Cryptography and Network Security (ACNS 2005)*, ser. Lecture Notes in Computer Science, J. Ioannidis, A. D. Keromytis, and M. Yung, Eds., vol. 3531. Springer, 2005, pp. 164–175. [Online]. Available: https://doi.org/10.1007/11496137_12

[39] J. Breier, W. He, and D. Jap, "Deep learning side-channel analysis and its application on pqc," in *Proc. ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2020, pp. 169–182.

[40] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, vol. 2523. Springer, 2002, pp. 13–28.

[41] National Institute of Standards and Technology (NIST), "Fips 205: Stateless hash-based digital signature algorithm (slh-dsa)," Federal Information Processing Standard, 2024, available: https://csrc.nist.gov/publications/detail/fips/205/final.

[42] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Post-Quantum Cryptography (PQCrypto 2011)*. Springer, 2011, pp. 19–34.

[43] National Institute of Standards and Technology (NIST), "Post-quantum cryptography standardization project," https://csrc.nist.gov/projects/post-quantum-cryptography, 2022.

[44] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology — CRYPTO '96*. Springer, 1996, pp. 104–113.

[45] Y. Zhang, Q. Li, and Y. Chen, "Entropy analysis and leakage in pqc implementations: A survey," *Entropy*, vol. 27, no. 3, pp. 112–133, 2025.

[46] J. Qiu and A. Aysu, "SHIFT SNARE: Uncovering secret keys in FALCON via single-trace analysis," Cryptology ePrint Archive, Paper 2025/146, 2025. [Online]. Available: https://eprint.iacr.org/2025/146

[47] G. Goy, A. Loiseau, and P. Gaborit, *A New Key Recovery Side-Channel Attack on HQC with Chosen Ciphertext*, 09 2022, pp. 353–371.

[48] National Institute of Standards and Technology (NIST), "Special publication 800-227: Recommendation for the implementation and management of module-lattice-based key-encapsulation mechanism (ml-kem)," Available: https://csrc.nist.gov/publications/detail/sp/800-227/ipd/final, National Institute of Standards and Technology, Tech. Rep. NIST SP 800-227 (IPD), 2024, initial Public Draft.

[49] H. Eldib, C. Wang, M. Taha, and P. Schaumont, "Qms: Evaluating the side-channel resistance of masked software from source code," 06 2014.

[50] M. Yan, C. W. Fletcher, and J. Torrellas, "Cache telepathy: Leveraging shared resource attacks to learn DNN architectures," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 2003–2020. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/yan

[51] R. Primas, P. Pessl, and S. Mangard, "Single-trace side-channel attacks on masked lattice-based encryption," Cryptology ePrint Archive, Paper 2017/594, 2017. [Online]. Available: https://eprint.iacr.org/2017/594

[52] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "Stellar: A generic em side-channel attack protection through ground-up root-cause analysis," 05 2019, pp. 11–20.

[53] Y. Yarom and K. Falkner, "Flush+reload: a high resolution, low noise, l3 cache side-channel attack," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, ser. SEC'14. USA: USENIX Association, 2014, p. 719–732.

[54] D. Gruss, C. Maurice, K. Wagner, and S. Mangard, "Flush+flush: A fast and stealthy cache attack," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, J. Caballero, U. Zurutuza, and R. J. Rodríguez, Eds. Cham: Springer International Publishing, 2016, pp. 279–299.

[55] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: The case of aes," in *Topics in Cryptology – CT-RSA 2006*, D. Pointcheval, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–20.

[56] É. Brier, D. Naccache, P. Q. Nguyen, and M. Tibouchi, "Modulus fault attacks against rsa-crt signatures," in *Cryptographic Hardware and Embedded Systems – CHES 2011*, B. Preneel and T. Takagi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 192–206.

[57] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, gauss, and reload – a cache attack on the BLISS lattice-based signature scheme," Cryptology ePrint Archive, Paper 2016/300, 2016. [Online]. Available: https://eprint.iacr.org/2016/300

[58] E. Oswald and S. Mangard, "Template attacks on masking—resistance is futile," in *Topics in Cryptology – CT-RSA 2007*, M. Abe, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 243–256.

[59] C. Archambeau, E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Template attacks in principal subspaces," in *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop*, ser. Lecture Notes in Computer Science, vol. 4249. Springer, 2006, pp. 1–14. [Online]. Available: https://iacr.org/archive/ches2006/01/01.pdf

[60] J. Mu, Y. Zhao, Z. Wang, J. Ye, J. Fan, S. Chen, H. Li, X. Li, and Y. Cao, "A voltage template attack on the modular polynomial subtraction in kyber," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2022, pp. 672–677.

[61] Y. Chang, Y. Yan, C. Zhu, and P. Guo, "Template attack of lwe/lwr-based schemes with cyclic message rotation," *Entropy*, vol. 24, no. 10, 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/10/1489

[62] H. Wang, H. Sayadi, A. Sasan, S. Rafatirad, and H. Homayoun, "Hybridg: Hybrid dynamic time warping and gaussian distribution model for detecting emerging zero-day microarchitectural side-channel attacks," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020, pp. 604–611.

[63] Y. Mao, V. Migliore, and V. Nicomette, "Rehad: Using low-frequency reconfigurable hardware for cache side-channel attacks detection," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*, 2020, pp. 704–709.

[64] D. Kim, S. Lee, K.-H. Lee *et al.*, "Deepxca: A deep neural network framework for side-channel analysis against cryptographic implementations," *IEEE Transactions on Information Forensics and Security*, 2022.

[65] D. Das, A. Golder, J. Danial, S. Ghosh, A. Raychowdhury, and S. Sen, "X-deepsca: Cross-device deep learning side channel attack," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[66] F. Hameed and H. Alkhzaimi, "Hybrid genetic algorithm and deep learning techniques for advanced side-channel attacks," *Scientific Reports*, vol. 15, 07 2025.

[67] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: Deep learning-based physical side-channel analysis," Cryptology ePrint Archive, Paper 2021/1092, 2021. [Online]. Available: https://eprint.iacr.org/2021/1092

[68] A. T. Hoang, M. Kennaway, D. T. Pham, T. S. Mai, A. Khalid, C. Rafferty, and M. O'Neill, "Deep learning enhanced side channel analysis on crystals-kyber," in *2024 25th International Symposium on Quality Electronic Design (ISQED)*, 2024, pp. 1–8.

[69] A. Rezaeezade, A. Basurto-Becerra, L. Weissbart, and G. Perin, "One for all, all for ascon: Ensemble-based deep learning side-channel analysis," Cryptology ePrint Archive, Paper 2023/1922, 2023. [Online]. Available: https://eprint.iacr.org/2023/1922

[70] D. Saha and F. Farahmandi, "DL-SCADS: Deep learning-based post-silicon side-channel analysis using decomposed signal," Cryptology ePrint Archive, Paper 2025/011, 2025. [Online]. Available: https://eprint.iacr.org/2025/011

[71] L. Zhang, M. He, J. Li *et al.*, "Dl-scads: Deep learning side-channel attack data set," *Journal of Cryptographic Engineering*, 2023.

[72] L. Batina, S. Bhasin, and D. Jap, "Csi nn: Reverse engineering of neural network architectures through electromagnetic side-channel," *Proc. USENIX Security Symposium*, 2019.

[73] E. Dubrova, K. Ngo, and J. Gärtner, "Breaking a fifth-order masked implementation of CRYSTALS-kyber by copy-paste," Cryptology ePrint Archive, Paper 2022/1713, 2022. [Online]. Available: https://eprint.iacr.org/2022/1713

[74] B.-Y. Sim, J. Kwon, J. Lee, I.-J. Kim, T.-H. Lee, J. Han, H. Yoon, J. Cho, and D.-G. Han, "Single-trace attacks on message encoding in lattice-based kems," *IEEE Access*, vol. 8, pp. 183 175–183 191, 2020.

[75] A. Genêt, N. L. de Guertechin, and N. Kaluđerović, "Full key recovery side-channel attack against ephemeral SIKE on the cortex-m4," Cryptology ePrint Archive, Paper 2021/858, 2021. [Online]. Available: https://eprint.iacr.org/2021/858

[76] Z. Chen, E. Karabulut, A. Aysu, Y. Ma, and J. Jing, "An efficient non-profiled side-channel attack on the crystals-dilithium post-quantum signature," in *Proceedings (camera-ready preprint)*, 2021, demonstrates CPA on Dilithium polynomial/NTT operations. [Online]. Available: https://research.ece.ncsu.edu/wp-content/uploads/sites/8/Dilithium_SCA_cameraready.pdf

[77] T. Schamberger, L. Holzbaur, J. Renner, A. Wachter-Zeh, and G. Sigl, "A power side-channel attack on the reed-muller reed-solomon version of the hqc cryptosystem," in *Post-Quantum Cryptography (PQCrypto)*, ser. LNCS, vol. 13418. Springer, 2022, pp. 327–352.

[78] G. Goy, A. Loiseau, and P. Gaborit, "A new key recovery side-channel attack on hqc with chosen ciphertext," in *Post-Quantum Cryptography (PQCrypto)*, ser. LNCS, vol. 13418. Springer, 2022, pp. 353–371.

[79] S. Huang, R. Sim, C. Chuengsatiansup, Q. Guo, and T. Johansson, "Cache-timing attack against hqc," Manuscript/Preprint, 2023, demonstrates chosen-ciphertext cache-timing on HQC. [Online]. Available: https://www.researchgate.net/publication/371465468_Cache-Timing_Attack_Against_HQC

[80] M. Tunstall, D. Mukhopadhyay, and S. Ali, "Differential fault analysis of the advanced encryption standard using a single fault," in *WISTP 2011*, ser. LNCS, vol. 6633. Springer, 2011, pp. 224–233.

[81] S. Jin and et al., "Recent advances in deep learning-based side-channel analysis," *ETRI Journal*, vol. 42, no. 6, pp. 848–861, 2020.

[82] M. Ninan, B. Wang, and J. M. Emmert, "A second look at the portability of deep learning side-channel attacks," in *RAID 2024*, 2024. [Online]. Available: https://homepages.uc.edu/~wang2ba/files/pub/raid24_mabon.pdf

[83] M.-J. O. Saarinen, "Introduction to side-channel security of nist pqc standards," NIST PQC Seminar presentation, Apr. 2023. [Online]. Available: https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/pqc-seminars/presentations/2-side-channel-security-saarinen-04042023.pdf

[84] R. Gorla, D. Moody, and editors, "Status report on the second round of the nist post-quantum cryptography standardization process," National Institute of Standards and Technology, Tech. Rep. NISTIR 8309, Jul. 2020, includes implementation considerations such as constant-time guidance. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2020/nist.ir.8309.pdf

[85] "Module-lattice-based key-encapsulation mechanism standard (fips 203)," National Institute of Standards and Technology, Tech. Rep. FIPS 203, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.203.pdf

[86] "Module-lattice-based digital signature standard (fips 204)," National Institute of Standards and Technology, Tech. Rep. FIPS 204, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.204.pdf

[87] "Stateless hash-based digital signature standard (fips 205)," National Institute of Standards and Technology, Tech. Rep. FIPS 205, Aug. 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.205.pdf

[88] "Implementation guidance for fips 140-3 and iso/iec 19790/24759," NIST & CCCS, Cryptographic Module Validation Program (CMVP), Tech. Rep., 2020, living document; accessed 2025-08-09. [Online]. Available: https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf

[89] "Iso/iec 17825:2024, information technology — security techniques — testing methods for the mitigation of non-invasive attack classes against cryptographic modules," International Organization for Standardization, 2024. [Online]. Available: https://www.iso.org/standard/82422.html

[90] "Fips 140-3 cmvp management manual," NIST CMVP, Tech. Rep., May 2025. [Online]. Available: https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS-140-3-CMVP%20Management%20Manual.pdf

[91] L. T. A. N. B. ao, M. Davidson, and A. Vassilev, "Nist roadmap toward criteria for threshold schemes for cryptographic primitives," National Institute of Standards and Technology, Tech. Rep. NISTIR 8214A, Jul. 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8214A.pdf

[92] K. Schaffer, "Cmvp approved non-invasive attack mitigation test metrics," National Institute of Standards and Technology, Tech. Rep. NIST SP 800-140F, 2020, replaces ISO/IEC 19790 Annex F non-invasive test metrics. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140F.pdf

[93] "Submission requirements and evaluation criteria for the post-quantum cryptography standardization process," National Institute of Standards and Technology, Tech. Rep., Dec. 2016. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/post-quantum-cryptography/documents/call-for-proposals-final-dec-2016.pdf

[94] B. Timon, "Shuffler: Side-channel resilient shuffle," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2017, pp. 273–289.

[95] C.-F. Tokunaga and D. Blaauw, "Secure aes engine with a local switching power network," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2009, pp. 79–84.

[96] T. Kamucheka, M. Fahr, T. Teague, A. Nelson, D. Andrews, and M. Huang, "Power-based side channel attack analysis on pqc algorithms," in *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2022, pp. 125–136.

[97] D. Kim and S. Lee, "Adversarial learning-based side-channel attack countermeasures," *IEEE Transactions on Information Forensics and Security*, 2021.

[98] Y. Bo and J. Wang, "Noise augmentation techniques for side-channel attack countermeasures," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2020, pp. 189–207.

[99] S. Picek and et al., "On the performance of gans for side-channel analysis," in *Constructive Side-Channel Analysis and Secure Design (COSADE)*. Springer, 2019.

[100] M. Ossmann and E. Oswald, "Chipwhisperer: An open-source platform for hardware security research," in *Proc. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2014.

[101] A. Heuser and M. Zohner, "Intelligent machine learning side-channel attacks," in *Proc. Smart Card Research and Advanced Applications (CARDIS)*, 2015, pp. 79–94.

[102] E. Prouff, M. Rivain, and T. Roche, "Ascad: A side-channel dataset for profiling attacks on aes," Cryptology ePrint Archive, Report 2018/734, 2018, https://eprint.iacr.org/2018/734.