Migration to Post-Quantum Cryptography: From ECDSA to ML-DSA

Daniel Dinu
INT31, Intel

***Abstract*—Cryptography is a fundamental building block of many security features like secure boot, remote attestation, trusted platform module (TPM), memory/disk encryption, and secure communication, providing confidentiality, data integrity, authentication, and non-repudiation. Post-Quantum Cryptography (PQC) marks an important milestone in the history of modern cryptography. It encompasses cryptographic algorithms designed to withstand cryptanalytic attacks from both quantum and classical computers.**

Organizations around the world are currently in the process of migrating to the PQC algorithms standardized by the National Institute of Standards and Technologies (NIST). Compared to the previous changes of cryptographic algorithms, the transition to PQC poses new challenges. We exemplify some of them by analyzing implementation attacks (e.g., side-channel and fault injection) and countermeasures applicable to the signature generation of the Elliptic Curve Digital Signature Algorithm (ECDSA), a widely used cryptographic algorithm, and the Module-Lattice-Based Digital Signature Algorithm (ML-DSA), a quantum-resistant algorithm set to replace the former.

***Index Terms*—post-quantum cryptography (PQC), implementation attacks, side channel, fault injection, countermeasures.**

I. Introduction

Cryptography is a crucial component for the security and privacy of a wide range of applications. Typically, cryptographic algorithms are meticulously designed and then vetted by the community with the aim of identifying any security issues. Some of the algorithms resist this scrutiny without any significant security issues being identified and therefore may be standardized to serve as robust and interoperable cryptographic systems. The confidence in the security of cryptographic algorithms may weaken with time as adversary capabilities increase or new attack techniques are discovered. These advancements may motivate the standardization of new algorithms based on constructions or hard problems that are not susceptible to known attacks. The standardization must be completed well before the attacks are practical to allow for replacement of the old algorithms with the new ones. This is the case for most public-key cryptography algorithms, which are vulnerable to attacks by a cryptographically relevant quantum computer (CRQC), a quantum computer powerful enough to solve their underlying hard problems by running Shor's algorithm [1]. Symmetric cryptography is less affected by the use of Grover's algorithm [2], and increasing the key size for block ciphers (e.g., from 128 to 256 bits) and digest size for hash functions (e.g., from 256 bits to 384 or 512 bits) is considered sufficient to prevent attacks by quantum computers. The term Post-Quantum Cryptography (PQC), also known as quantum-resistant or quantum-safe cryptography, encompasses all cryptographic algorithms designed to withstand attacks by classical and quantum computers. Typically, it is used to refer to public-key algorithms only. In this paper, we focus on quantum-vulnerable and quantum-safe public-key algorithms.

A. The NIST PQC Standardization Process

Motivated by the threat posed by the evolution of quantum computing to the security of most public-key algorithms currently in use, the National Institute of Standards and Technologies (NIST) started in December 2016 the PQC Standardization Process, a public competition for selection of public-key cryptosystems designed to resist attacks by a quantum computer. After three rounds of evaluation, in July 2022, NIST announced the first four proposals to be standardized, which include one key-establishment mechanism (i.e., CRYSTALS-KYBER [3], [4]) and three digital signatures (i.e., CRYSTALS-Dilithium [5], [6], FALCON [7] and SPHINCS$^+$ [8], [9]). CRYSTALS-Kyber and CRYSTALS-Dilithium are the primary algorithms recommended for most use cases, while FALCON and SPHINCS$^+$ are proposed for use cases that require small signatures and signatures not based on lattices, respectively. Shortly after NIST's announcement, in September 2022, the National Security Agency (NSA) published the Commercial National Security Algorithm Suite (CNSA) 2.0 advisory on protection of National Security Systems (NSS), which includes the approved PQC algorithms and the transition timeline [10]. In August 2023, NIST requested public comments on the drafts of the standards derived from CRYSTALS-KYBER, CRYSTALS-Dilithium, and SPHINCS$^+$. Standards based on CRYSTALS-KYBER, CRYSTALS-Dilithium, and SPHINCS$^+$ were published in August 2024 as Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) [11], Module-Lattice-Based Digital Signature Algorithm (ML-DSA) [12], and Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) [13], respectively. In March 2025, NIST announced the selection of HQC [14] as a backup algorithm for ML-KEM. The second round for the selection of additional signature schemes is ongoing with 14 candidate algorithms under consideration.

B. Migration to PQC

The migration to post-quantum cryptography involves the change of currently used cryptographic algorithms, such as Rivest–Shamir–Adleman (RSA) [15]–[17], Elliptic Curve Digital Signature Algorithm (ECDSA) [16], [18] and Edwards-curve Digital Signature Algorithm (EdDSA) [16], [19], with new cryptographic algorithms designed to resist attacks from classical and large-scale quantum computers, such as Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) [11] and Module-Lattice-Based Digital Signature Algorithm (ML-DSA) [12]. This migration is a major milestone in the evolution of cryptography and is expected to end by 2033 [10] or 2035 [20]. A similar transition was the replacement of the Data Encryption Standard (DES) [21] with the Advanced Encryption Standard (AES) [22], which took more than a decade [20]. More than two decades after the standardization of the AES, some systems still support the Data Encryption Standard (DES) for legacy reasons, and some of them plan to remove it [23], [24].

C. Implementation Attacks and Countermeasures

Implementations of cryptographic algorithms are susceptible to two main categories of attacks, namely side-channel analysis attacks and fault injection attacks. A side-channel analysis attack aims to recover a secret key by exploiting information, such as time, power consumption, or electromagnetic radiation, passively measured from a device that executes cryptographic operations with that key. A fault injection attack perturbs the normal operation of a device by altering its clock signal, voltage levels, or by inducing electromagnetic pulses. A fault may affect some operations or values and therefore bypass a verification or produce a different output from which the attacker may recover the secret key of the algorithm.

These two categories of attacks are commonly applied against implementations of cryptographic algorithms and are sometimes referred to as physical attacks, although in certain settings they can be performed remotely. Therefore, detection and prevention mechanisms are required when these attacks are in scope. Detection techniques can use various types of sensors to identify abnormal operating conditions. They typically require careful fine-tuning to increase detection rate without too many false positives. The owner of a system can decide how to react to an abnormal event. Countermeasures against side-channel attacks are divided into masking and hiding techniques. Masking represents a sensitive value as two or more shares, of which all but one are randomly generated for each execution of the algorithm, and the remaining one is computed knowing that the combination of all the shares gives the sensitive value. Operations are computed on the shared representation. Masking comes with some theoretical security guarantees that may hold in practice if properly implemented but has a significant overhead. Hiding countermeasures typically have a lower overhead and include techniques such as shuffling the order of operations and adding random operations. Finally, various types of redundancy, integrity protections, and checks are useful against fault injection attacks.

Two good books on the topic are [25], [26]. A survey of physical security in the era of quantum computers can be found in [27].

D. Context

Several papers summarized the existing attacks and countermeasures applicable to ECDSA [28]–[32] and ML-DSA [33]–[35]. In addition to that, numerous publications discuss various aspects of the migration to PQC [20], [36]–[46]. Side-channel attacks and countermeasure applicable to PQC algorithms are systematized in [47]. Besides NIST and NSA, similar organizations from around the world published various documents on the topic [48]–[53].

Our goal is to raise awareness of some important practical engineering challenges on the way to a successful transition to PQC. To that end, we present a comparative analysis of two digital signature algorithms at the opposite ends of the migration path. The starting point for our discussion is the quantum-vulnerable ECDSA, an algorithm currently used in many systems. The transition approach may include an optional intermediate step consisting of a hybrid signature scheme based on a quantum-vulnerable and a quantum-safe algorithm. The destination is the quantum-resistant ML-DSA. Along the way, we focus on two categories of implementation attacks, namely side-channel analysis attacks and fault injection attacks, applicable to the signature generation operation as well as countermeasures that detect and prevent these attacks.

This work provides an overview of the state of the art, systematizing and referencing relevant resources that may be useful for a successful migration from ECDSA to ML-DSA. It provides a certain level of technical details and insights into the subject, but it does not aim at an exhaustive or detailed treatment of all possible aspects in the interest of simplicity and brevity. For more details about a particular topic (e.g., a specific attack or countermeasure), we recommend the reader to consult the referenced papers. Finally, we do not discuss attacks and countermeasures applicable to the key generation and signature verification operations.

II. Public-Key Cryptography

Table I provides a summary of the public-key cryptography algorithms approved, standardized, and selected for standardization by NIST along with some important characteristics.

A. Quantum-Vulnerable Algorithms

Seven classical public-key algorithms are approved by NIST. The first four are pair-wise key establishment schemes based on discrete logarithm cryptography. They were standardized by the Accredited Standards Committee (ASC) X9, an organization accredited by the American National Standards Institute (ANSI), before being approved by NIST in the Special Publication (SP) 800-series [54]. The Finite Field Cryptography Diffie-Hellman (FFC DH) was proposed in [55], while the Elliptic Curve Cryptography Diffie-Hellman (ECC DH) was introduced in [56], [57]. The Finite Field Cryptography Menezes-Qu-Vanstone (FFC MQV) and the Elliptic Curve Cryptography Menezes-Qu-Vanstone (ECC MQV) were described in [58].

The remaining three algorithms are part of the Digital Signature Standard (DSS) [16] and were standardized at different

TABLE I
Overview of Public-Key Cryptography

Algorithm	Con	KE	Sig	Pro	NIST	PQC	NSA
FFC DH	①	✔	✖	1976	2006	✖	✖
FFC MQV	①	✔	✖	1995	2006	✖	✖
ECC DH	②	✔	✖	1985	2006	✖	✖
ECC MQV	②	✔	✖	1995	2006	✖	✖
RSA	③	✔	✔	1977	1998	✖	✖
ECDSA	②	✖	✔	1992	2000	✖	✖
EdDSA	②	✖	✔	2011	2023	✖	✖
LMS	④	✖	✔	1995	2020	✔	✔
XMSS	④	✖	✔	2011	2020	✔	✔
ML-KEM	⑥	✔	✖	2016	2024	✔	✔
ML-DSA	⑥	✖	✔	2016	2024	✔	✔
SLH-DSA	⑤	✖	✔	2016	2024	✔	✖
FN-DSA	⑥	✖	✔	2016	TBA	✔	✖
HQC*	⑦	✔	✖	2016	TBA	✔	✖

Con – Construction/Hard Problem
① Discrete Logarithm Problem (DLP).
② Elliptic Curve Discrete Logarithm Problem (ECDLP).
③ Large integer factorization.
④ Stateful hash-based.
⑤ Stateless hash-based.
⑥ Lattice-based.
⑦ Code-based.
KE – Key Establishment/Key Encapsulation Mechanism
Sig – Signature Algorithm
Pro – When proposed by its designers
NIST – When initially approved by NIST (FIPS, SP)
TBA = To be announced.
PQC – Post-Quantum Cryptography
NSA – Approved by NSA (CNSA 2.0)
* The name of the standard is to be announced.

times, starting with the Rivest-Shamir-Adleman (RSA) [15], continuing with the Elliptic Curve Digital Signature Algorithm (ECDSA) [18] and most recently the Edwards-curve Digital Signature Algorithm (EdDSA) [19]. All algorithms were standardized by other organizations before being standardized by NIST as a Federal Information Processing Standard (FIPS). The use of RSA for pair-wise key establishment is described in [17].

B. Quantum-Resistant Algorithms

The Leighton-Micali Signature (LMS) and the eXtended Merkle Signature Scheme (XMSS) [59] are stateful hash-based signature (HBS) schemes. A major limitation of stateful schemes is that the signing process must be done in a highly controlled environment to ensure that the one-time signature (OTS) keys are not reused. They were standardized by the Internet Research Task Force (IRTF) and later added by NIST to the SP 800-series [60]. The remaining five algorithms were selected from the submissions to the NIST PQC Standardization Process (see Section I-A). Other algorithms may be selected from the additional signature scheme proposals to diversify the portfolio of digital signature schemes with algorithms that do not use structured lattices as well as schemes with short signatures and fast verification.

C. Discussion

The quantum-vulnerable algorithms (five key establishment schemes and three digital signature schemes) are expected to be replaced by quantum-safe algorithms. The current list of quantum-safe algorithms includes seven algorithms (two key encapsulation mechanisms and five digital signature algorithms) and more may be added.

One thing to notice is that the list of quantum vulnerable algorithms includes five key establishment schemes and three signature algorithms, while the current inventory of PQC algorithms contains two key encapsulation mechanisms and five digital signature schemes. Hence, the ratio between key establishment and digital signature algorithms for PQC is close to the inverse ratio for classical public-key algorithms. The proportion of quantum-safe signature algorithms may increase as new digital signature schemes may be added to the catalog.

Although the number of PQC algorithms is similar to the number of quantum-vulnerable algorithms they will replace, the latter were added gradually over a period of about 25 years, often after more than a decade since they were initially proposed by their designers. Moreover, they have been approved by NIST after being standardized by other organizations. On the other hand, the quantum-safe algorithms were approved by NIST in the last five years. That implies a large volume of work if all algorithms are to be supported in applications by 2033 [10] or 2035 [20]. This is often referred to as the *scale* of the PQC migration, which is unprecedented in the history of modern cryptography.

A similarity between the two algorithms further studied in this paper, namely ECDSA and ML-DSA, is that both were standardized by NIST after about eight years after they were initially proposed. A difference is that ECDSA was first standardized by the Accredited Standards Committee (ASC) X9, an organization accredited by the American National Standards Institute (ANSI), before being standardized by NIST about one year later [54], while ML-DSA emerged from the standardization process organized by NIST [12].

Given that not all the quantum-vulnerable algorithms are widely-used today, one may expect that may be the case for PQC as well, despite the crypto agility ideals. A gradual transition approach addresses the scale challenge. It may first implement the stateful hash-based signatures, then add support for the primary PQC algorithms, and finally the alternate schemes. Some organizations may start directly with the primary or alternate algorithms, and others may implement only a subset of the algorithms. Supporting a single PQC algorithm for key encapsulation or digital signatures may expose a system if that scheme is broken in the future. A hybrid scheme [44] may provide a hedge against that, provided a CRQC is not available at that time. It entails using both a quantum-vulnerable and a quantum-safe algorithm such that if only one of them is broken, the scheme is still secure thanks to the other one. This is recommended as an intermediate step before the full transition to PQC [20], [50], [51], [53].

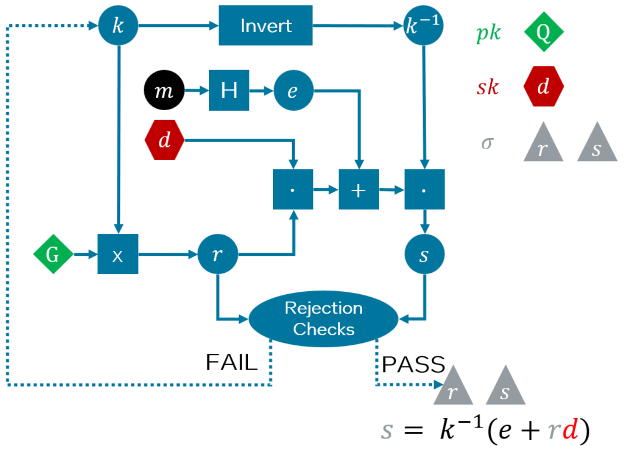

Fig. 1. Simplified ECDSA signature generation.

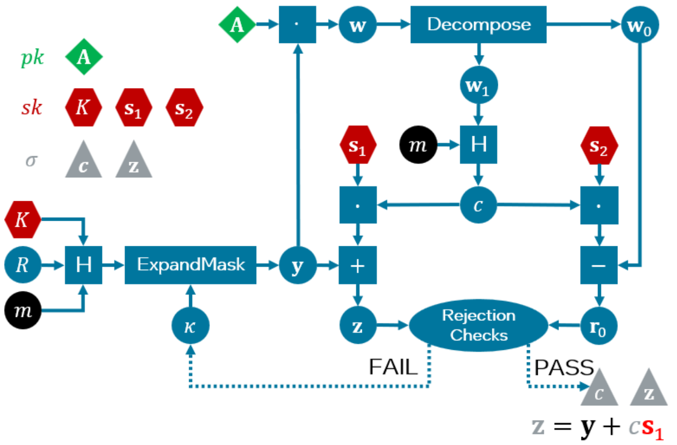

Fig. 2. Simplified ML-DSA signature generation.

III. Comparative Study

We focus on the signature generation operation of two digital signature schemes, namely the quantum-vulnerable ECDSA and the quantum-secure ML-DSA. The sign operation produces a signature σ for a given message m and a secret key sk, and is a frequent target of implementation attacks. Next, we consider the randomized versions of the algorithms. We remind the reader that the deterministic versions of these algorithm are more susceptible to physical attacks, especially fault injection attacks [61]–[65], and should be avoided if a strong random number generator is available on the signing device.

A. Algorithms

1) ECDSA: The Elliptic Curve Digital Signature Algorithm (ECDSA) [16] was proposed by Vanstone in response to the NIST request for comments on the Digital Signature Standard (DSS) [18]. It is a variant of the deprecated Digital Signature Algorithm (DSA) based on Elliptic-Curve Cryptography (ECC). A simplified graphical representation of the signature generation operation is shown in Fig. 1. The inputs to the algorithm are the message to be signed m, which is shown as a black circle, the private key d, depicted as a red hexagon, and the public elliptic curve base point G, illustrated as a green diamond. The algorithm returns a signature $\sigma = (r, s)$ which consists of two integers, drawn as gray triangles. The first signature component is the integer r obtained from the multiplication ($\times$) of the elliptic curve point G by a randomly generated integer, or scalar, k. The second component is $s = k^{-1}(e + rd)$, where k^{-1} is the modular inverse of the scalar k, and e is the hash of the message m to be signed. In the very unlike case that $r = 0$ or $s = 0$, the process is repeated by generating a new random scalar k.

There are two curves currently recommended by NIST, namely P-384 and P-512 [66]. They will be disallowed after 2035 [20]. ECDSA is not on the list of approved algorithms by NSA (i.e., CNSA 2.0) [10], and only the curve P-384 was approved in a previous version (i.e., CNSA 1.0). Hence, we consider ECDSA 384, also known as ECDSA-384 or ECDSA P-384, which uses the NIST P-384 curve (also called secp384r1) and provides a security strength of 192 bits, as a baseline for our discussion.

2) ML-DSA: The Module-Lattice-based Digital Signature Algorithm (ML-DSA) [12] is the primary PQC digital signature algorithm. It is derived from the CRYSTALS-Dilithium [5], [6], a structured lattice scheme. A simplified version of the sign operation is provided in Fig. 2. The inputs to the sign algorithm are the message to be signed m, the secret key $sk = (K, \mathbf{s}_1, \mathbf{s}_2)$, and the public matrix of polynomials $\mathbf{A}$. The components of the private key are the stream of bits K, and the vectors of polynomials with short coefficients $\mathbf{s}_1$ and $\mathbf{s}_2$. We use the same graphical representations as before, namely the message to be signed is drawn as a black circle, each component of the secret key is illustrated using a red hexagon, the public key components are shown as green diamonds, and the signature components are represented by gray triangles. The signature $\sigma = (c, \mathbf{z})$ consists of a polynomial c and a vector of polynomials $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$, where $\mathbf{y}$ is a vector of polynomials with coefficients in a large range. The polynomial c has a fixed small number of coefficients equal to -1 or $+1$, while the others are set to 0.

NIST standardized three parameter sets for ML-DSA, namely ML-DSA-44, ML-DSA-65, and ML-DSA-87 [12], while NSA approved in CNSA 2.0 only ML-DSA-87 [10]. Therefore, we use ML-DSA-87, which has a security strength of 256 bits [20], for our comparisons.

3) Discussion: The most time consuming operation of the ECDSA sign algorithm is the elliptic curve point multiplication ($\times$). It consists of a sequence of point addition and doubling operations, which are defined using modular arithmetic (i.e., addition, subtraction, multiplication). Another important operation is the `Invert` function which computes the modular inverse of k. The remaining operations are a hash computation (`H`), a modular addition ($+$) and two modular multiplications ($\cdot$). The signature generation loop is expected to generate a valid signature after one single iteration, and

TABLE II
KEY AND SIGNATURE SIZE IN BYTES

Algorithm	**Private Key**		**Public Key**		**Signature**
	Full	**Compact**	**Full**	**Compact**	
ECDSA 384	48	48	97	49	96
ML-DSA-87	4,896	32	2,592	2,592	4,627
Ratio	1:102	1.5:1	1:26.7	1:52.9	1:48.2

execution of more iterations is very unlikely.

Most of the execution time of ML-DSA is spent on the matrix-vector multiplication $\mathbf{Ay}$, the polynomial multiplications in $c\mathbf{s}_1$ and $c\mathbf{s}_2$, and several hash computations ($\mathtt{H}$). Polynomial operations boil down to modular arithmetic. The exact number of polynomial multiplications, additions, and subtractions performed in a single iteration depends on the parameters of the algorithm. Other operations are the `ExpandMask` function which generates the vector $\mathbf{y}$ and the `Decompose` function which computes the vectors $\mathbf{w}_0$ and $\mathbf{w}_1$. The algorithm typically performs several iterations to generate a valid signature, and each iteration implies new executions of the aforementioned operations. For example, the expected number of repetitions of the rejection loop is 3.85 for ML-DSA-87 [12].

Comparing the operations required by each of the two algorithms, their number, and cost, we identify another challenge, namely the *complexity* of the PQC algorithms.

B. Key and Signature Size

Table II summarizes the size in bytes of the public key, private key, and signature for ECDSA 384 and ML-DSA-87. We include the size of a compressed (or compact) representation of a key when that exists. Additional computation is required to obtain a full (or uncompressed) representation of a key from a compact representation.

1) ECDSA: The private key pk is the scalar d, a 384-bit integer, which can be represented on 48 bytes, and does not have a compressed representation.

The public key pk is a point $Q = d \times G$ on the elliptic curve that consists of two 384-bit coordinates, x and y. Hence, each coordinate can be represented using 48 bytes. Additionally, one byte is used as a prefix. Thus, the uncompressed representation requires $2 \cdot 48 + 1 = 97$ bytes, while the compressed version needs only $48 + 1 = 49$ bytes. It is possible to get the full public key from the compressed public key because the y coordinate can be computed from the curve equation knowing the value of the x coordinate and a bit in the prefix that indicates if the y coordinate is even or odd. Encoding of the keys may further increase their size by several bytes depending on the format, but is not considered.

Finally, the signature has two components, r and s, each of which is a 384-bit integer. Hence, the signature size is $2 \cdot 48 = 96$ bytes.

2) ML-DSA: The private key sk includes the stream of bits K, and the two polynomials with small coefficients $\mathbf{s}_1$ and $\mathbf{s}_2$. In addition to that, it includes three components which are omitted for simplicity. The total size of the encoded secret key is 4,896 bytes, and a compressed representation stores only the 32-byte seed from which the entire secret key is generated (see [12, Sec. 4] for more details).

The public key pk includes the matrix $\mathbf{A}$ and vector $\mathbf{t} = \mathbf{As}_1 + \mathbf{s}_2$. However, instead of storing the entire matrix and vector, one can save only the 32-byte seed from which the matrix was generated and a compressed value of vector $\mathbf{t}$. This representation of the public key takes 2,592 bytes. If additional space is available, one can store the full matrix $\mathbf{A}$ to speed up the signing and verification (see [12, Sec. 4] for more details).

3) Discussion: By analyzing the values in Table II, one can see that with the exception of the compressed private key of ML-DSA, the public key, private key, and signature of ML-DSA and are larger then those of ECDSA. Hence, when comparing ML-DSA to ECDSA, more memory is required to store some of these values and more data is transmitted. Certain products and protocols may have to be modified or even redesigned to support ML-DSA.

NIST considered the size of the public keys and signatures as part of the second evaluation criterion (i.e., cost) that was used to compare the candidate algorithms [67]–[69].

C. Implementations

Implementations can be categorized into hardware, software, and a combination thereof. They are crucial for real-world deployment of cryptographic algorithms, and this reflected in the second evaluation criterion (i.e., performance) used by NIST to compare candidate algorithms throughout the standardization process [67]–[69].

1) ECDSA: There are numerous publications describing hardware implementations of ECDSA such as [70]–[74]. Some implementations include countermeasures against physical attacks (e.g., [74]). A survey of ECC implementations applicable to blockchains is conducted in [75].

2) ML-DSA: To the best of our knowledge, there are three open-source hardware implementations of ML-DSA [76]–[78]. The first two of them target Field-Programmable Gate Arrays (FPGAs) and are described in [79], [80], while the third one is available as a discrete cryptographic accelerator and is also integrated into the open-sourced Caliptra Root of Trust (RoT) [81]. It includes some countermeasures. Finally, the authors of [82] describe a first-order masked implementation based on [77], but its source code is not publicly available.

3) Discussion: In general, the more mature a standardized algorithm is, the more implementations it has. Typically, optimized software implementations are more numerous than optimized hardware implementations of the same algorithm because of factors like development time and flexibility. These observations hold for implementations ECDSA and ML-DSA. We do not aim to compare implementation characteristics of the two algorithms as this is done in other works. A study of supported of PQC algorithms in cryptographic libraries is described in [83]. Detailed benchmarking results can be found at [84]–[90].

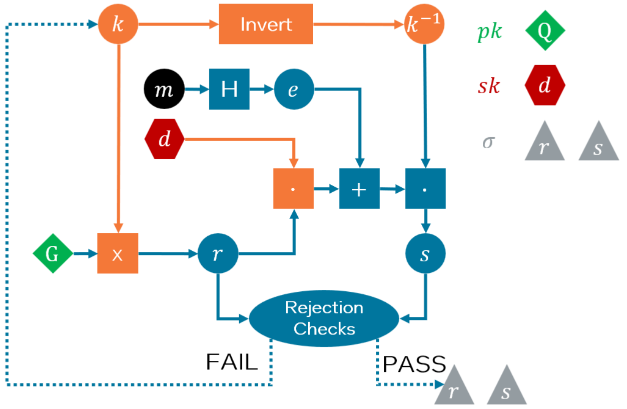

Fig. 3. Simplified ECDSA signature generation with operations susceptible to side-channel attacks colored in orange.

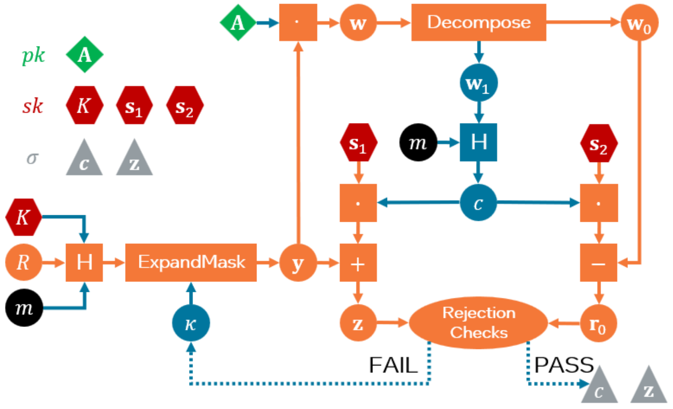

Fig. 4. Simplified ML-DSA signature generation with operations susceptible to side-channel attacks in orange.

D. Side-Channel Attacks

We colored in orange those operations susceptible to side-channel and fault injection attacks in Fig. 3 and Fig. 4 for ECDSA and ML-DSA, respectively. The operations that may be targeted by a side-channel attacker were identified using a so-called sensitivity analysis, which tracks the use of sensitive variables through the algorithm. That was correlated with the existing attacks published in the literature.

1) ECDSA: Most side-channel attacks published in the literature focus on the elliptic curve point multiplication (×). Other vulnerable operations include the modular multiplications (rd), and the modular inversion (`Invert`). The rejection checks must be protected against fault injection attacks, but they do not pose any side-channel security risk. Template attacks are described in [91], [92]. Known side-channel attacks are summarized in [28]–[32].

2) ML-DSA: It is simpler to enumerate the operations and intermediate variables that are not sensitive to attacks. The list includes the vector of polynomials $\mathbf{w}_1$ because it can be computed from the public information used in the signature verification procedure, the hash function (`H`) applied to m and $\mathbf{w}_1$, the challenge polynomial c, and the iteration counter κ. Yuanyuan, Weijia, Yiteng and Yu [93] described an attack that exploits two sources of leakage, namely the coefficients of the polynomial in c and the leakage of the coefficients that cause rejection of a signature $\mathbf{z}$ for the same polynomial c, building on the work of Karabulut and Aysu [94]. The aforementioned attack would not be possible if at least one of the target operations is protected. All other parts of the algorithm are possible targets for implementation attacks. The rejection checks must be protected against both attack categories.

The literature includes numerous examples of attacks on the polynomial multiplication and/or the Number Theoretic Transform (NTT) [95]–[105], which is used to efficiently perform polynomial multiplication. Two frequent targets are the multiplications between c and $\mathbf{s}_1$ and c and $\mathbf{s}_2$ because they involve small-coefficient polynomials. Only two [104], [105] of the aforementioned papers considered a hardware implementation, with the rest focused on software implementations running on microcontrollers or simulated traces. To the best of our knowledge, no paper demonstrated an attack on the polynomial multiplication between $\mathbf{A}$ and $\mathbf{y}$ despite such an attack seems possible.

A significant number of papers focused on profiled attacks. Profiled attacks on NTT were described in [101], [106], [107] using traces from a 32-bit microcontroller. Profiled single-trace attacks on the hash function (`H`) were exemplified on simulated leakage for 8-bit and 32-bit microcontrollers in [108]. Profiled attacks were shown on the implementation of `BitUnpack` function on 32-bit microcontrollers in [109], [110]. This function is used by `ExpandMask` as well as to load the secret polynomials $\mathbf{s}_1$ and $\mathbf{s}_2$ from the secret key sk. Finally, a profiled attack on the computation of the coefficients of $\mathbf{w}_0$ inside the `Decompose` function was demonstrated using traces from a 32-bit microcontroller in [111]. Several papers summarized the side channel attacks applicable to ML-DSA [33], [35].

3) Real-World Attacks: Notable real-world attacks against implementations of ECDSA include extraction of the secret key from: PlayStation 3 due to the reuse of the scalar k [112], several security chips and cryptographic libraries due the leakage of the bit-length of the scalar used in the elliptic curve point multiplication [113], and the Google Titan secure element due to electromagnetic leakage in the scalar multiplication [114]. The last two attacks affected Common Criteria (CC) certified devices.

A Correlation Power Analysis (CPA) targeting $c\mathbf{s}_1$ was demonstrated on an open-sourced hardware implementation of ML-DSA in a Root of Trust (RoT) in [105].

4) Discussion: In contrast to the several operations of the ECDSA algorithm that can be a target for side-channel attacks, almost all operations of the ML-DSA algorithm are susceptible to these attacks. In other words, *the attack surface of the PQC algorithm is larger*. This stems from the *complex* structure of the algorithm, in which almost every intermediate value is a sensitive value that can reveal information about the secret

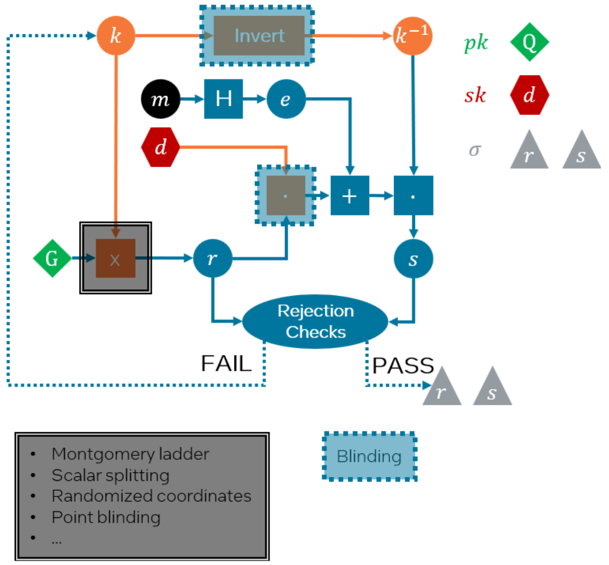

Fig. 5. Simplified ECDSA signature generation with side-channel countermeasures.

key.

Compared to the rejection checks of ECDSA, the rejection checks of ML-DSA also require protection against side-channel analysis attacks.

With respect to timing attacks, the rejection loop does not pose a threat as the number of iterations does not depend on the secret key of the algorithms. All sensitive operations of ML-DSA are easy to implement in constant time [6]. That is not necessarily the case for ECDSA (See Section III-D3).

Most attacks demonstrated in the literature on ML-DSA targeted software implementations running on microcontrollers, and therefore may not be easily transferable to hardware implementations due to their inherent parallelism which may hinder some of the attacks.

E. Side-Channel Countermeasures

1) ECDSA: The elliptic curve point multiplication (×) needs robust countermeasures against side-channel attacks. There are various methods to compute the point multiplication, some of which are better than others from a side-channel perspective. For example, a constant-time Montgomery ladder is preferred to the Double-and-Add-Always method, which is better than the Double-and-Add algorithm. Other useful countermeasures include splitting the scalar, randomizing the base point coordinates, and blinding the base point. A good level of protection can be obtained by a combination of the aforementioned countermeasures as well as other mitigations. The modular multiplication between the scalar and secret key can be efficiently protected using a blinding countermeasure. Similarly, the modular inversion operation can also be protected using blinding. These countermeasures are summarized in Fig. 5. For more details, see [115]. A protected software implementation is descrived in [116].

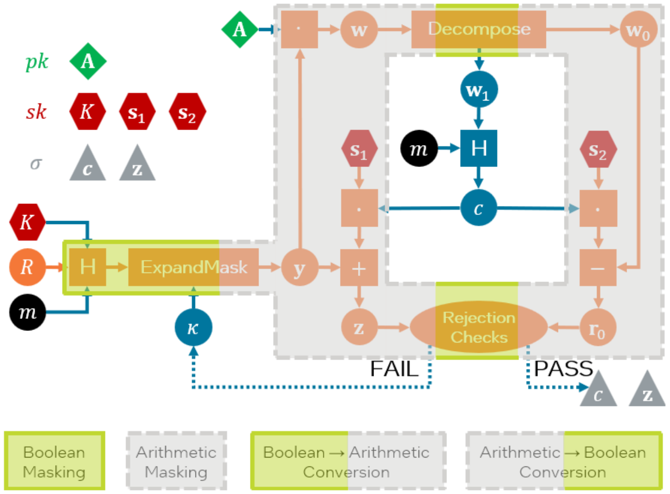

Fig. 6. Simplified ML-DSA signature generation with side-channel countermeasures.

2) ML-DSA: For efficient protection against all known side-channel attacks, ML-DSA needs a combination of shuffling and masking countermeasures. Shuffling is effective against some profiled attacks, while masking can hinder differential attacks as well as some profiled attacks. For better protection, both masking and shuffling can be combined. Masking all the sensitive operations of ML-DSA requires a combination of Boolean and arithmetic masking because some operations are masked more easily with the former (e.g., `H`, parts of `ExpandMask` and `Rejection Checks`), while others with the latter (e.g., polynomial operations). While the Boolean to arithmetic conversion has a low cost, the arithmetic to Boolean conversion adds a considerable overhead. The application of the masking countermeasure is graphically represented in Fig. 6.

Countermeasures are typically discussed to various extents in the papers that demonstrate attacks. First, we look at countermeasures for the polynomial multiplication, including the Number Theoretic Transform (NTT). The authors of [100] proposed masking as a countermeasure against their attack. An attack on a first-order masked implementation was described in [102]. The authors of [101] conclude that masking is not sufficient, and shuffling is recommended as an effective countermeasure. The authors of [103] demonstrated that their attack is effective against shuffled implementations of the small-norm polynomial multiplication, and recommend masking countermeasures thanks to their exponential impact on the attack complexity. The shuffling countermeasures proposed in [117] were successfully attacked in [99]. Finally, several papers recommend both masking and shuffling [98], [104], [105].

Second, we summarize countermeasures that can prevent profiled attacks. Masking and shuffling were proposed to prevent the attacks targeting the Number Theoretic Transform (NTT) [106] and the hash function [108]. Regarding the attacks on the `BitUnpack` function, the authors of [109]

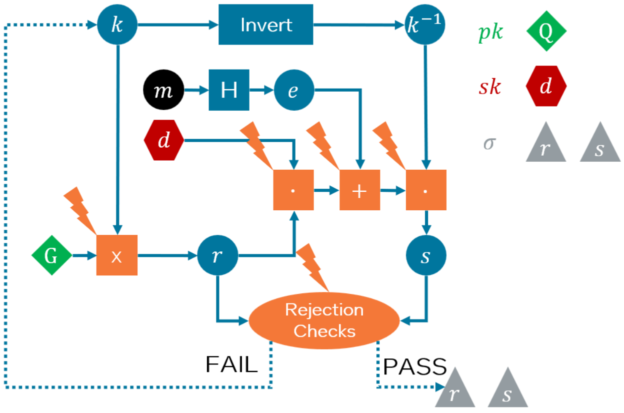

Fig. 7. Simplified ECSA signature generation with operations susceptible to fault injection attacks colored in orange.

remark that their attack is difficult to prevent without masking the entire signature generation algorithm, while the authors of [110] discuss masking, shuffling, and encoding of the secret key using a constant-weight code. Shuffling and masking are also proposed to prevent the attack on the coefficients of $\mathbf{w}_0$ described in [111].

Several papers describe side-channel countermeasures [100], [117]–[120]. Masked implementations of Dilithium are presented in [82], [120].

3) Discussion: We described the countermeasures applicable to the signature generation of ECDSA and ML-DSA to achieve a similar protection against side-channel attacks. By comparing Fig. 5 and Fig. 6, one can notice that ECDSA requires mitigations only for several of its operations, while ML-DSA must be almost entirely protected. Corroborating this with the complexity of each algorithm, the overhead of a side-channel resistant implementation of ML-DSA is expected to be higher than the overhead of a protected implementation of ECDSA.

In practice, the appropriate side-channel protections are selected using a combination of factors, including the intended use cases, their associated threat models, and overheads. Therefore, in some situation less or no countermeasures will suffice, while in others more may be required.

F. Fault Injection Attacks

As in the case of side-channel attacks, we conducted a vulnerability analysis and corroborated it with known attacks published in the literature to identify those parts of the algorithms that are sensitive to fault injection attacks, and therefore need protection against these attacks.

1) ECDSA: An important class of fault attacks comprises techniques that switch the scalar multiplication from the intended strong elliptic curve to a weaker one by faulting either the curve point or the parameters of the curve [121]–[123]. A second category includes the safe-error attacks. The C safe-error attacks skip the use of dummy operations in implementations like Double-and-Add-Always [124], while the M safe-error attacks introduce a temporary memory fault inside a register or memory location [125]. Some of these aforementioned attacks use differential fault analysis (e.g., [121], [123]). Finally, fault attacks can also skip operations like the modular addition $(+)$, modular multiplication $(\cdot)$, and rejection checks. Fault injection attacks applicable to ECDSA are synthesized in [28]–[32].

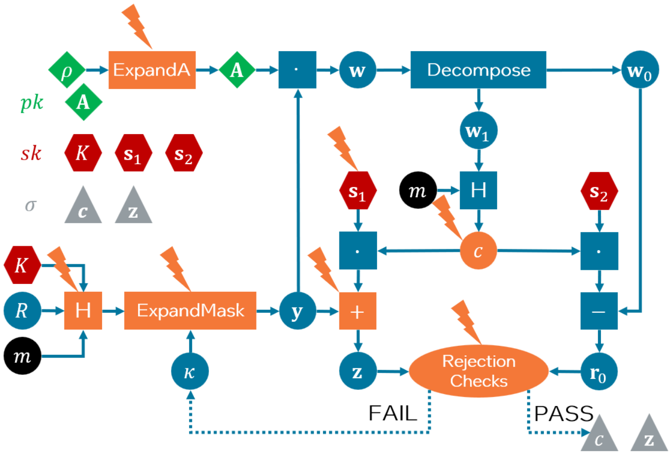

Fig. 8. Simplified ML-DSA signature generation with operations susceptible to fault injection attacks colored in orange.

2) ML-DSA: Most attacks described in the literature target the computations within the `ExpandMask` function [103], [126]–[130]. For a summary of the loop abort fault attacks, see also [131]. A fault on the absorption of data in H allows an attacker to predict the input to the `ExpandMask` function [132]. Attacks that flip bits in the coefficients of the secret polynomial $\mathbf{s}_1$ were discussed in [133], [134]. Attacks that skip the addition of $\mathbf{y}$ and $c\mathbf{s}_1$ are presented in [126], [135]. Attacks on the `ExpandA` function affect the computation of matrix $\mathbf{A}$ [136], [137]. Finally, [138] describes fault attacks on the Number Theoretic Transform (NTT) of the challenge polynomial c and vector of polynomials $\mathbf{y}$ for deterministic and randomized versions of the algorithm, respectively. The attack assumes that the precomputed twiddle constants are stored in memory. Some attacks use the RowHammer mechanism to inject faults [134], [139]. Several papers summarized the fault injection attacks applicable to ML-DSA [33]–[35].

3) Discussion: Both algorithms need countermeasures as they are vulnerable to simple and differential fault analysis attacks. The rejection checks of both ECDSA and ML-DSA need protections against fault attacks.

G. Fault Injection Countermeasures

1) ECDSA: The selection of the scalar multiplication method has an influence on the resistance to fault injection attacks, and a constant-time Montgomery ladder is preferred. Always verify a signature before releasing it. Check that the coordinates of a point are on the elliptic curve. Various levels of redundancy, up to executing the signature generation multiple times and comparing the results, can be explored. For more details, see [28]–[32].

2) ML-DSA: A signature should be verified before being released. Several simple checks can be implemented to determine whether certain operations (e.g., loops) were completely executed, the vector $\mathbf{y}$ and the matrix $\mathbf{A}$ were correctly generated. Various amounts of redundancy can be added. For example, the last iteration of the rejection loop can be computed twice to ensure the obtained signatures match.

3) Discussion: A common trait of both algorithms is that there is no single countermeasure that prevents all known fault attacks. Another similarity is the need to protect the rejection checks. As in the case of side-channel countermeasures, several factors influence the choice of the appropriate fault injection countermeasures.

IV. Conclusion

We conducted a comparative study of the sign operation of two digital signature algorithms, the widely used quantum-vulnerable ECDSA and the quantum-secure ML-DSA, in the context of the PQC migration. We used three different criteria, namely algorithm specifications, implementation attacks, and countermeasures, to exemplify some of the similarities and differences between the two algorithms as well as some important practical challenges (unprecedented scale, algorithm complexity, key and signature size, attack surface, protection cost) to consider when planning a successful migration to PQC. Finally, the transition to PQC is a necessary, but not easy, step that could benefit from well-informed planning.

Acknowledgment

We would like to thank to all our colleagues and collaborators for the valuable discussions and insightful comments on earlier drafts of this paper.

References

[1] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, November 20-22, 1994*. IEEE Computer Society, 1994, pp. 124–134. [Online]. Available: https://doi.org/10.1109/SFCS.1994.365700

[2] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, G. L. Miller, Ed. ACM, 1996, pp. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[3] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM,” in *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*. IEEE, 2018, pp. 353–367. [Online]. Available: https://doi.org/10.1109/EuroSP.2018.00032

[4] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS-Kyber algorithm specifications and supporting documentation (version 3.0),” *Round 3 Submission to the NIST Post-Quantum Standardization Process*, pp. 1–42, 2020. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[5] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS-Dilithium: A lattice-based digital signature scheme,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 238–268, 2018. [Online]. Available: https://doi.org/10.13154/tches.v2018.i1.238-268

[6] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS-Dilithium: Algorithm specifications and supporting documentation,” *Round 3 Submission to the NIST Post-Quantum Standardization Process*, pp. 1–38, 2020.

[7] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, “Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU (Specification v1.2),” *Round 3 Submission to the NIST Post-Quantum Standardization Process*, pp. 1–67, 2020.

[8] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, “The SPHINCS$^+$ signature framework,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 2129–2146. [Online]. Available: https://doi.org/10.1145/3319535.3363229

[9] J.-P. Aumasson, D. J. Bernstein, W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, and B. Westerbaan, “SPHINCS$^+$ submission to the NIST post-quantum project, v.3,” *Round 3 Submission to the NIST Post-Quantum Standardization Process*, pp. 1–62, 2020.

[10] National Security Agency, “Announcing the Commercial National Security Algorithm Suite 2.0,” 2022. [Online]. Available: https://media.defense.gov/2025/May/30/2003728741/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS.PDF

[11] National Institute of Standards and Technology, “Module-Lattice-Based Key-Encapsulation Mechanism Standard,” Department of Commerce, Washington, D.C., Federal Information Processing Standards Publication (FIPS) NIST FIPS 203, 2024. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.203

[12] ——, “Module-Lattice-Based Digital Signature Standard,” Department of Commerce, Washington, D.C., Federal Information Processing Standards Publication (FIPS) NIST FIPS 204, 2024. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.204

[13] ——, “Stateless Hash-Based Digital Signature Standard,” Department of Commerce, Washington, D.C., Federal Information Processing Standards Publication (FIPS) NIST FIPS 205, 2024. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.205

[14] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, A. Dion, P. Gaborit, J. Lacan, E. Persichetti, J.-M. Robert, P. Véron, and G. Zémor, “Hamming Quasi-Cyclic (HQC) (Fourth round version),” *Round 4 Submission to the NIST Post-Quantum Standardization Process*, pp. 1–49, 2022.

[15] R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[16] National Institute of Standards and Technology, “Digital Signature Standard (DSS),” Department of Commerce, Washington, D.C., Federal Information Processing Standards Publication (FIPS) NIST FIPS 186-5, 2023. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.186-5

[17] ——, “Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography,” Department of Commerce, Washington, D.C., Special Publication (SP) NIST SP 800-56Br2, 2019. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-56Br2

[18] S. Vanstone, “Responses to NIST’s proposal,” *Communications of the ACM*, vol. 35, no. 7, pp. 50–52, 1992.

[19] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B. Yang, “High-speed high-security signatures,” *J. Cryptogr. Eng.*, vol. 2, no. 2, pp. 77–89, 2012. [Online]. Available: https://doi.org/10.1007/s13389-012-0027-1

[20] National Institute of Standards and Technology, “Transition to Post-Quantum Cryptography Standards,” Department of Commerce, Washington, D.C., Internal Report (IR) NIST IR 8547 ipd, 2024, initial Public Draft. [Online]. Available: https://doi.org/10.6028/NIST.IR.8547.ipd

[21] ——, “Data Encryption Standard (DES),” Department of Commerce, Washington, D.C., Federal Information Processing Standards Publication (FIPS) NIST FIPS 46-3, 1999, withdrawn on May 19, 2005. [Online]. Available: https://csrc.nist.gov/files/pubs/fips/46-3/final/docs/fips46-3.pdf

[22] ——, “Advanced Encryption Standard (AES),” Department of Commerce, Washington, D.C., Federal Information Processing Standards

Publication (FIPS) NIST FIPS 197-upd1, 2001, updated May 9, 2023. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.197-upd1

[23] Anushka Khare, "Removal of DES in Kerberos for Windows Server and Client," 2025. [Online]. Available: https://techcommunity.microsoft.com/blog/windowsservernewsandbestpractices/removal-of-des-in-kerberos-for-windows-server-and-client/4386903

[24] Cybersecurity and Infrastructure Security Agency, "Transition to Advanced Encryption Standard," 2024. [Online]. Available: https://www.cisa.gov/sites/default/files/2024-05/23_0918_fpic_AES-Transition-WhitePaper_Final_508C_24_0513.pdf

[25] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[26] J. Van Woudenberg and C. O'Flynn, *The Hardware Hacking Handbook: Breaking Embedded Security with Hardware Attacks*. No Starch Press, 2021.

[27] S. Chowdhury, A. Covic, R. Y. Acharya, S. Dupee, F. Ganji, and D. Forte, "Physical security in the post-quantum era," *J. Cryptogr. Eng.*, vol. 12, no. 3, pp. 267–303, 2022. [Online]. Available: https://doi.org/10.1007/s13389-021-00255-w

[28] R. M. Avanzi, "Side channel attacks on implementations of curve-based cryptographic primitives," *IACR Cryptol. ePrint Arch.*, p. 17, 2005. [Online]. Available: http://eprint.iacr.org/2005/017

[29] J. Fan, X. Guo, E. D. Mulder, P. Schaumont, B. Preneel, and I. Verbauwhede, "State-of-the-art of secure ECC implementations: A survey on known side-channel attacks and countermeasures," in *HOST 2010, Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 13-14 June 2010, Anaheim Convention Center, California, USA*, J. Plusquellic and K. Mai, Eds. IEEE Computer Society, 2010, pp. 76–87. [Online]. Available: https://doi.org/10.1109/HST.2010.5513110

[30] J. Fan and I. Verbauwhede, "An updated survey on secure ECC implementations: Attacks, countermeasures and cost," in *Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, ser. Lecture Notes in Computer Science, D. Naccache, Ed., vol. 6805. Springer, 2012, pp. 265–282. [Online]. Available: https://doi.org/10.1007/978-3-642-28368-0_18

[31] J. Danger, S. Guilley, P. Hoogvorst, C. Murdica, and D. Naccache, "A synthesis of side-channel attacks on elliptic curve cryptography in smart-cards," *J. Cryptogr. Eng.*, vol. 3, no. 4, pp. 241–265, 2013. [Online]. Available: https://doi.org/10.1007/s13389-013-0062-6

[32] R. Abarzúa, C. V. Cordero, and J. López, "Survey on performance and security problems of countermeasures for passive side-channel attacks on ECC," *J. Cryptogr. Eng.*, vol. 11, no. 1, pp. 71–102, 2021. [Online]. Available: https://doi.org/10.1007/s13389-021-00257-8

[33] M. Taha and T. Eisenbarth, "Implementation attacks on post-quantum cryptographic schemes," *IACR Cryptol. ePrint Arch.*, p. 1083, 2015. [Online]. Available: http://eprint.iacr.org/2015/1083

[34] F. Valencia, I. Polian, and F. Regazzoni, "Fault sensitivity analysis of lattice-based post-quantum cryptographic components," in *Embedded Computer Systems: Architectures, Modeling, and Simulation - 19th International Conference, SAMOS 2019, Samos, Greece, July 7-11, 2019, Proceedings*, ser. Lecture Notes in Computer Science, D. N. Pnevmatikatos, M. Pelcat, and M. Jung, Eds., vol. 11733. Springer, 2019, pp. 107–123. [Online]. Available: https://doi.org/10.1007/978-3-030-27562-4_8

[35] P. Ravi, A. Chattopadhyay, J. D'Anvers, and A. Baksi, "Side-channel and fault-injection attacks over lattice-based post-quantum schemes (Kyber, Dilithium): Survey and new results," *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 35:1–35:54, 2024. [Online]. Available: https://doi.org/10.1145/3603170

[36] A. Wiesmaier, N. Alnahawi, T. Grasmeyer, J. Geißler, A. Zeier, P. Bauspieß, and A. Heinemann, "On PQC migration and crypto-agility," *CoRR*, vol. abs/2106.09599, 2021. [Online]. Available: https://arxiv.org/abs/2106.09599

[37] N. von Nethen, A. Wiesmaier, O. Weissmann, and N. Alnahawi, "Managing the migration to post-quantum-cryptography," *CoRR*, vol. abs/2301.04491, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2301.04491

[38] Cybersecurity and Infrastructure Security Agency (CISA), National Security Agency (NSA) National Institute of Standards and Technology (NIST), "Quantum-Readiness: Migration to Post-Quantum Cryptography," 2023. [Online]. Available: https://www.cisa.gov/resources-tools/resources/quantum-readiness-migration-post-quantum-cryptography

[39] P. Kampanakis and T. Lepoint, "Vision paper: Do we need to change some things? - open questions posed by the upcoming post-quantum migration to existing standards and deployments," in *Security Standardisation Research - 8th International Conference, SSR 2023, Lyon, France, April 22-23, 2023, Proceedings*, ser. Lecture Notes in Computer Science, F. Günther and J. Hesse, Eds., vol. 13895. Springer, 2023, pp. 78–102. [Online]. Available: https://doi.org/10.1007/978-3-031-30731-7_4

[40] Alessandro Amadori, Thomas Attema, Maxime Bombar, João Diogo Duarte, Vincent Dunning, Simona Etinski, Daniël van Gent, Matthieu Lequesne, Ward van der Schoot, Marc Stevens and AIVD Cryptologists and Advisors, "The PQC Migration Handbook: Guidelines for Migrating to Post-Quantum Cryptography," 2024, revised and Extended Edition. [Online]. Available: https://publications.tno.nl/publication/34643386/fXcPVHsX/TNO-2024-pqc-en.pdf

[41] K. F. Hasan, L. Simpson, M. A. R. Baee, C. Islam, Z. Rahman, W. Armstrong, P. Gauravaram, and M. McKague, "A framework for migrating to post-quantum cryptography: Security dependency analysis and case studies," *IEEE Access*, vol. 12, pp. 23 427–23 450, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3360412

[42] NIS Cooperation Group, "A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography," 2025, Part 1, Version 1.1, EU PQC Qorkstream. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography

[43] Post Quantum Cryptography Coalition, "Post-Quantum Cryptography (PQC) Migration Roadmap," 2025. [Online]. Available: https://pqcc.org/post-quantum-cryptography-migration-roadmap/

[44] NXP Post-Quantum Cryptography Team, "Post-Quantum Cryptographic: Migration Challenges for Embedded Devices." [Online]. Available: https://www.nxp.com/docs/en/white-paper/POSTQUANCOMPWPA4.pdf

[45] Secure-IC, "Redefining Security - Confronting the Industrial Challenges of Implementing Post-Quantum Cryptography (PQC)." [Online]. Available: https://hub.secure-ic.com/post-quantum-cryptography

[46] A. Aysu, D. Dinu, K. Gaj, F. Ganji, M. Hashemi, R. J, D. Mehta, M.-J. O. Saarinen, P. Schaumont, and C. Tol, "Open tools, interfaces and metrics for implementation security testing: Testing post-quantum cryptography implementation security," Online, June 2025, working Document. [Online]. Available: https://optimist-ose.org/assets/files/pqc05-06308116ce3dcd2d2edd34255985f303.pdf

[47] P. Dobias, A. Rezaeezade, L. Chmielewski, L. Malina, and L. Batina, "Sok: Reassessing side-channel vulnerabilities and countermeasures in PQC implementations," *IACR Cryptol. ePrint Arch.*, p. 1222, 2025. [Online]. Available: https://eprint.iacr.org/2025/1222

[48] ENISA – The European Union Agency for Cybersecurity, "Post-Quantum Cryptography: Current state and quantum mitigation," 2021, May 2021, v2. [Online]. Available: https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation

[49] ——, "Post-Quantum Cryptography - Integration study," 2022, October 2022. [Online]. Available: https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study

[50] Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI; English: French National Agency for the Security of Information Systems), "ANSSI views on the Post-Quantum Cryptography transition," 2022, January 2022. [Online]. Available: https://cyber.gouv.fr/en/publications/anssi-views-post-quantum-cryptography-transition

[51] Bundesamt für Sicherheit in der Informationstechnik (BSI; English: Federal Office for Information Security), "Quantum-safe cryptography – fundamentals, current developments and recommendations," 2022, 18.05.2022. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Brochure/quantum-safe-cryptography.html

[52] European Commission, "Recommendation on a Coordinated Implementation Roadmap for the transition to Post-Quantum Cryptography," 2024, c(2024) 2393 final. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/recommendation-coordinated-implementation-roadmap-transition-post-quantum-cryptography

[53] Bundesamt für Sicherheit in der Informationstechnik (BSI; English: Federal Office for Information Security), "BSI TR-02102-1 "Cryptographic Mechanisms: Recommendations and Key Lengths" Version: 2025-1," 2025, 04.03.2025. [Online].

Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Brochure/quantum-safe-cryptography.html

[54] National Institute of Standards and Technology, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography,” Department of Commerce, Washington, D.C., Special Publication (SP) NIST SP 800-56Ar3, 2018. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-56Ar3

[55] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976. [Online]. Available: https://doi.org/10.1109/TIT.1976.1055638

[56] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[57] V. S. Miller, “Use of elliptic curves in cryptography,” in *Advances in Cryptology - CRYPTO ’85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, ser. Lecture Notes in Computer Science, H. C. Williams, Ed., vol. 218. Springer, 1985, pp. 417–426. [Online]. Available: https://doi.org/10.1007/3-540-39799-X_31

[58] A. Menezes, M. Qu, and V. S. A, “Some new key agreement protocols providing implicit authentication,” in *2nd Workshop on Selected Areas in Cryptography (SAC)’95*, 1995.

[59] J. Buchmann, E. Dahmen, and A. Hülsing, “XMSS - A practical forward secure signature scheme based on minimal security assumptions,” in *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, ser. Lecture Notes in Computer Science, B. Yang, Ed., vol. 7071. Springer, 2011, pp. 117–129. [Online]. Available: https://doi.org/10.1007/978-3-642-25405-5_8

[60] National Institute of Standards and Technology, “Recommendation for Stateful Hash-Based Signature Schemes,” Department of Commerce, Washington, D.C., Special Publication (SP) NIST SP 800-208, 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-208

[61] A. Barenghi and G. Pelosi, “A note on fault attacks against deterministic signature schemes,” in *Advances in Information and Computer Security - 11th International Workshop on Security, IWSEC 2016, Tokyo, Japan, September 12-14, 2016, Proceedings*, ser. Lecture Notes in Computer Science, K. Ogawa and K. Yoshioka, Eds., vol. 9836. Springer, 2016, pp. 182–192. [Online]. Available: https://doi.org/10.1007/978-3-319-44524-3_11

[62] C. Ambrose, J. W. Bos, B. Fay, M. Joye, M. Lochter, and B. Murray, “Differential attacks on deterministic signatures,” in *Topics in Cryptology - CT-RSA 2018 - The Cryptographers’ Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, ser. Lecture Notes in Computer Science, N. P. Smart, Ed., vol. 10808. Springer, 2018, pp. 339–353. [Online]. Available: https://doi.org/10.1007/978-3-319-76953-0_18

[63] D. Poddebniak, J. Somorovsky, S. Schinzel, M. Lochter, and P. Rösler, “Attacking deterministic signature schemes using fault attacks,” in *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*. IEEE, 2018, pp. 338–352. [Online]. Available: https://doi.org/10.1109/EuroSP.2018.00031

[64] L. G. Bruinderink and P. Pessl, “Differential fault attacks on deterministic lattice signatures,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 21–43, 2018. [Online]. Available: https://doi.org/10.13154/tches.v2018.i3.21-43

[65] W. Cao, H. Shi, H. Chen, J. Chen, L. Fan, and W. Wu, “Lattice-based fault attacks on deterministic signature schemes of ECDSA and eddsa,” in *Topics in Cryptology - CT-RSA 2022 - Cryptographers’ Track at the RSA Conference 2022, Virtual Event, March 1-2, 2022, Proceedings*, ser. Lecture Notes in Computer Science, S. D. Galbraith, Ed., vol. 13161. Springer, 2022, pp. 169–195. [Online]. Available: https://doi.org/10.1007/978-3-030-95312-6_8

[66] National Institute of Standards and Technology, “Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters,” Department of Commerce, Washington, D.C., Special Publication (SP) NIST SP 800-186, 2023. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-186

[67] ——, “Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process,” Department of Commerce, Washington, D.C., Internal Report (IR) NIST IR 8413-upd1, 2022, July 2022. [Online]. Available: https://doi.org/10.6028/NIST.IR.8413-upd1

[68] ——, “Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process,” Department of Commerce, Washington, D.C., Internal Report (IR) NIST IR 8309, 2020, July 2020. [Online]. Available: https://doi.org/10.6028/NIST.IR.8309

[69] ——, “Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process,” Department of Commerce, Washington, D.C., Internal Report (IR) NIST IR 8240, 2019, January 2019. [Online]. Available: https://doi.org/10.6028/NIST.IR.8240

[70] A. Sghaier, Z. Medien, and M. Machhout, “Fast hardware implementation of ECDSA signature scheme,” in *International Symposium on Signal, Image, Video and Communications, ISIVC 2016, Tunis, Tunisia, November 21-23, 2016*. IEEE, 2016, pp. 343–348. [Online]. Available: https://doi.org/10.1109/ISIVC.2016.7894012

[71] M. Bedoui, B. Bouallegue, A. M. Ahmed, B. Hamdi, M. Machhout, and M. M. Khattab, “A secure hardware implementation for elliptic curve digital signature algorithm,” *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2177–2193, 2023. [Online]. Available: https://doi.org/10.32604/csse.2023.026516

[72] M. Mühlberghuber, “Comparing ECDSA hardware implementations based on binary and prime fields,” 2011, master Thesis. Accessed: September 25, 2025.

[73] P. Pessl and M. Hutter, “Curved tags - A low-resource ECDSA implementation tailored for RFID,” in *Radio Frequency Identification: Security and Privacy Issues - 10th International Workshop, RFIDSec 2014, Oxford, UK, July 21-23, 2014, Revised Selected Papers*, ser. Lecture Notes in Computer Science, N. Saxena and A. Sadeghi, Eds., vol. 8651. Springer, 2014, pp. 156–172. [Online]. Available: https://doi.org/10.1007/978-3-319-13066-8_10

[74] A. Salman, A. Ferozpuri, E. Homsirikamol, P. Yalla, J. Kaps, and K. Gaj, “A scalable ECC processor implementation for high-speed and lightweight with side-channel countermeasures,” in *International Conference on ReConFigurable Computing and FPGAs, ReConFig 2017, Cancun, Mexico, December 4-6, 2017*. IEEE, 2017, pp. 1–8. [Online]. Available: https://doi.org/10.1109/RECONFIG.2017.8279769

[75] R. Ifrim, D. Loghin, and D. Popescu, “A systematic review of fast, scalable, and efficient hardware implementations of elliptic curve cryptography for blockchain,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 17, no. 4, pp. 62:1–62:33, 2024. [Online]. Available: https://doi.org/10.1145/3696422

[76] Chair for Security Engineering @ Ruhr University Bochum, “dilithium-artix7,” https://github.com/Chair-for-Security-Engineering/dilithium-artix7, 2025, accessed: September 25, 2025.

[77] Cryptographic Engineering Research Group at George Mason University, “High-Performance Hardware Implementation of CRYSTALS-Dilithium,” https://github.com/GMUCERG/Dilithium, 2025, accessed: September 25, 2025.

[78] CHIPS Alliance, “Adam’s Bridge,” https://github.com/chipsalliance/adams-bridge/, 2025, accessed: September 25, 2025.

[79] G. Land, P. Sasdrich, and T. Güneysu, “A hard crystal - implementing Dilithium on reconfigurable hardware,” in *Smart Card Research and Advanced Applications - 20th International Conference, CARDIS 2021, Lübeck, Germany, November 11-12, 2021, Revised Selected Papers*, ser. Lecture Notes in Computer Science, V. Grosso and T. Pöppelmann, Eds., vol. 13173. Springer, 2021, pp. 210–230. [Online]. Available: https://doi.org/10.1007/978-3-030-97348-3_12

[80] L. Beckwith, D. T. Nguyen, and K. Gaj, “High-performance hardware implementation of CRYSTALS-Dilithium,” in *International Conference on Field-Programmable Technology, (IC)FPT 2021, Auckland, New Zealand, December 6-10, 2021*. IEEE, 2021, pp. 1–10. [Online]. Available: https://doi.org/10.1109/ICFPT52863.2021.9609917

[81] Mark Russinovich, “Adams Bridge: An Accelerator for Post-Quantum Resilient Cryptography,” 2024. [Online]. Available: https://techcommunity.microsoft.com/blog/azureconfidentialcomputingblog/adams-bridge-an-accelerator-for-post-quantum-resilient-cryptography/4269585

[82] K. Raj, P. Ravi, T. K. Chia, and A. Chattopadhyay, “Improved ML-DSA hardware implementation with first order masking countermeasure,” *Cryptology ePrint Archive*, 2024.

[83] N. Ahmed, L. Zhang, and A. Gangopadhyay, “A survey of post-quantum cryptography support in cryptographic libraries,” *CoRR*, vol. abs/2508.16078, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.16078

[84] D. J. Bernstein and T. Lange, “eBACS: ECRYPT Benchmarking of Cryptographic Systems,” https://bench.cr.yp.to, 2025, accessed: September 25, 2025.

[85] M. J. Kannwischer, R. Petri, J. Rijneveld, P. Schwabe, and K. Stoffelen, “PQM4: Post-quantum crypto library for the ARM Cortex-M4,” https://github.com/mupq/pqm4, accessed: September 25, 2025.

[86] M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, "pqm4: Testing and benchmarking NIST PQC on ARM cortex-m4," *IACR Cryptol. ePrint Arch.*, p. 844, 2019. [Online]. Available: https://eprint.iacr.org/2019/844

[87] C. E. R. G. (CERG), "Athena: Automated tools for hardware evaluation," https://cryptography.gmu.edu/athena/index.php, accessed: September 25, 2025.

[88] V. B. Dang, F. Farahmand, M. Andrzejczak, K. Mohajerani, D. T. Nguyen, and K. Gaj, "Implementation and benchmarking of round 2 candidates in the NIST post-quantum cryptography standardization process using hardware and software/hardware co-design approaches," *IACR Cryptol. ePrint Arch.*, p. 795, 2020. [Online]. Available: https://eprint.iacr.org/2020/795

[89] V. B. Dang, K. Mohajerani, and K. Gaj, "High-speed hardware architectures and FPGA benchmarking of CRYSTALS-Kyber, NTRU, and saber," *IEEE Trans. Computers*, vol. 72, no. 2, pp. 306–320, 2023. [Online]. Available: https://doi.org/10.1109/TC.2022.3222954

[90] J. Howe and B. Westerbaan, "Benchmarking and analysing the NIST PQC lattice-based signature schemes standards on the ARM cortex M7," in *Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings*, ser. Lecture Notes in Computer Science, N. E. Mrabet, L. D. Feo, and S. Duquesne, Eds., vol. 14064. Springer, 2023, pp. 442–462. [Online]. Available: https://doi.org/10.1007/978-3-031-37679-5_19

[91] M. Medwed and E. Oswald, "Template attacks on ECDSA," in *Information Security Applications, 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers*, ser. Lecture Notes in Computer Science, K. Chung, K. Sohn, and M. Yung, Eds., vol. 5379. Springer, 2008, pp. 14–27. [Online]. Available: https://doi.org/10.1007/978-3-642-00306-6_2

[92] A. C. Aldaya and B. B. Brumley, "When one vulnerable primitive turns viral: Novel single-trace attacks on ECDSA and RSA," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 2, pp. 196–221, 2020. [Online]. Available: https://doi.org/10.13154/tches.v2020.i2.196-221

[93] Y. Zhou, W. Wang, Y. Sun, and Y. Yu, "Rejected signatures' challenges pose new challenges: Key recovery of CRYSTALS-dilithium via side-channel attacks," Cryptology ePrint Archive, Paper 2025/214, 2025. [Online]. Available: https://eprint.iacr.org/2025/214

[94] E. Karabulut, E. Alkim, and A. Aysu, "Single-trace side-channel attacks on ω-small polynomial sampling: With applications to NTRU, NTRU Prime, and CRYSTALS-DILITHIUM," in *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2021, Tysons Corner, VA, USA, December 12-15, 2021*. IEEE, 2021, pp. 35–45. [Online]. Available: https://doi.org/10.1109/HOST49136.2021.9702284

[95] P. Ravi, M. P. Jhanwar, J. Howe, A. Chattopadhyay, and S. Bhasin, "Side-channel assisted existential forgery attack on Dilithium - A NIST PQC candidate," *IACR Cryptol. ePrint Arch.*, p. 821, 2018. [Online]. Available: https://eprint.iacr.org/2018/821

[96] I. Kim, T. Lee, J. Han, B. Sim, and D. Han, "Novel single-trace ML profiling attacks on NIST 3 round candidate Dilithium," *IACR Cryptol. ePrint Arch.*, p. 1383, 2020. [Online]. Available: https://eprint.iacr.org/2020/1383

[97] A. P. Fournaris, C. Dimopoulos, and O. G. Koufopavlou, "Profiling Dilithium digital signature traces for correlation differential side channel attacks," in *Embedded Computer Systems: Architectures, Modeling, and Simulation - 20th International Conference, SAMOS 2020, Samos, Greece, July 5-9, 2020, Proceedings*, ser. Lecture Notes in Computer Science, A. Orailoglu, M. Jung, and M. Reichenbach, Eds., vol. 12471. Springer, 2020, pp. 281–294. [Online]. Available: https://doi.org/10.1007/978-3-030-60939-9_19

[98] 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 *39th IEEE International Conference on Computer Design, ICCD 2021, Storrs, CT, USA, October 24-27, 2021*. IEEE, 2021, pp. 583–590. [Online]. Available: https://doi.org/10.1109/ICCD53106.2021.00094

[99] J. Hermelink, S. Streit, E. Strieder, and K. Thieme, "Adapting belief propagation to counter shuffling of NTTs," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 1, pp. 60–88, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i1.60-88

[100] H. M. Steffen, G. Land, L. J. Kogelheide, and T. Güneysu, "Breaking and protecting the crystal: Side-channel analysis of Dilithium in hardware," in *Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023, College Park, MD, USA, August 16-18, 2023, Proceedings*, ser. Lecture Notes in Computer Science, T. Johansson and D. Smith-Tone, Eds., vol. 14154. Springer, 2023, pp. 688–711. [Online]. Available: https://doi.org/10.1007/978-3-031-40003-2_25

[101] G. Assael, P. Elbaz-Vincent, and G. Reymond, "Improving single-trace attacks on the number-theoretic transform for Cortex-M4," in *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2023, San Jose, CA, USA, May 1-4, 2023*. IEEE, 2023, pp. 111–121. [Online]. Available: https://doi.org/10.1109/HOST55118.2023.10133270

[102] Z. Qiao, Y. Liu, Y. Zhou, M. Shao, and S. Sun, "When NTT meets SIS: efficient side-channel attacks on Dilithium and Kyber," *IACR Cryptol. ePrint Arch.*, p. 1866, 2023. [Online]. Available: https://eprint.iacr.org/2023/1866

[103] O. Bronchain, M. Azouaoui, M. ElGhamrawy, J. Renes, and T. Schneider, "Exploiting small-norm polynomial multiplication with physical attacks application to CRYSTALS-Dilithium," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2024, no. 2, pp. 359–383, 2024. [Online]. Available: https://doi.org/10.46586/tches.v2024.i2.359-383

[104] C. Mujdei, L. Wouters, A. Karmakar, A. Beckers, J. M. B. Mera, and I. Verbauwhede, "Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication," *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 27:1–27:23, 2024. [Online]. Available: https://doi.org/10.1145/3569420

[105] M. Karabulut and R. Azarderakhsh, "Efficient CPA attack on hardware implementation of ML-DSA in post-quantum root of trust," in *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2025, San Jose, CA, USA, May 5-8, 2025*. IEEE, 2025, pp. 392–402. [Online]. Available: https://doi.org/10.1109/HOST64725.2025.11050056

[106] P. Pessl and R. Primas, "More practical single-trace attacks on the number theoretic transform," in *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, ser. Lecture Notes in Computer Science, P. Schwabe and N. Thériault, Eds., vol. 11774. Springer, 2019, pp. 130–149. [Online]. Available: https://doi.org/10.1007/978-3-030-30530-7_7

[107] J. Han, T. Lee, J. Kwon, J. Lee, I. Kim, J. Cho, D. Han, and B. Sim, "Single-trace attack on NIST round 3 candidate Dilithium using machine learning-based profiling," *IEEE Access*, vol. 9, pp. 166 283–166 292, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3135600

[108] M. J. Kannwischer, P. Pessl, and R. Primas, "Single-trace attacks on Keccak," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 3, pp. 243–268, 2020. [Online]. Available: https://doi.org/10.13154/tches.v2020.i3.243-268

[109] V. Q. Ulitzsch, S. Marzougui, M. Tibouchi, and J. Seifert, "Profiling side-channel attacks on Dilithium - A small bit-fiddling leak breaks it all," in *Selected Areas in Cryptography - 29th International Conference, SAC 2022, Windsor, ON, Canada, August 24-26, 2022, Revised Selected Papers*, ser. Lecture Notes in Computer Science, B. Smith and H. Wu, Eds., vol. 13742. Springer, 2022, pp. 3–32. [Online]. Available: https://doi.org/10.1007/978-3-031-58411-4_1

[110] R. Wang, K. Ngo, J. Gärtner, and E. Dubrova, "Single-trace side-channel attacks on CRYSTALS-Dilithium: Myth or reality?" *IACR Cryptol. ePrint Arch.*, p. 1931, 2023. [Online]. Available: https://eprint.iacr.org/2023/1931

[111] A. Berzati, A. C. Viera, M. Chartouny, S. Madec, D. Vergnaud, and D. Vigilant, "Exploiting intermediate value leakage in Dilithium: A template-based approach," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 4, pp. 188–210, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i4.188-210

[112] bushing, marcan, segher, and sven, "Console Hacking 2010: PS3 Epic Fail," 2010, Presentation at the 27th Chaos Communications Congress. [Online]. Available: https://fahrplan.events.ccc.de/congress/2010/Fahrplan/events/4087.en.html

[113] J. Jancar, V. Sedlacek, P. Svenda, and M. Sýs, "Minerva: The curse of ECDSA nonces systematic analysis of lattice attacks on noisy leakage of bit-length of ECDSA nonces," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 4, pp. 281–308, 2020. [Online]. Available: https://doi.org/10.13154/tches.v2020.i4.281-308

[114] T. Roche, V. Lomné, C. Mutschler, and L. Imbert, "A side journey to Titan," in *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, M. D. Bailey and R. Greenstadt, Eds. USENIX Association, 2021, pp. 231–248. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/roche

[115] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," in *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, ser. Lecture Notes in Computer Science, Ç. K. Koç and C. Paar, Eds., vol. 1717. Springer, 1999, pp. 292–302. [Online]. Available: https://doi.org/10.1007/3-540-48059-5_25

[116] L. Batina, L. Chmielewski, B. Haase, N. Samwel, and P. Schwabe, "Sok: Sca-secure ECC in software - mission impossible?" *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 1, pp. 557–589, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i1.557-589

[117] P. Ravi, R. Poussier, S. Bhasin, and A. Chattopadhyay, "On configurable SCA countermeasures against single trace attacks for the NTT - A performance evaluation study over Kyber and Dilithium on the ARM Cortex-M4," in *Security, Privacy, and Applied Cryptography Engineering - 10th International Conference, SPACE 2020, Kolkata, India, December 17-21, 2020, Proceedings*, ser. Lecture Notes in Computer Science, L. Batina, S. Picek, and M. Mondal, Eds., vol. 12586. Springer, 2020, pp. 123–146. [Online]. Available: https://doi.org/10.1007/978-3-030-66626-2_7

[118] V. Migliore, B. Gérard, M. Tibouchi, and P. Fouque, "Masking Dilithium - Efficient implementation and side-channel evaluation," in *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings*, ser. Lecture Notes in Computer Science, R. H. Deng, V. Gauthier-Umaña, M. Ochoa, and M. Yung, Eds., vol. 11464. Springer, 2019, pp. 344–362. [Online]. Available: https://doi.org/10.1007/978-3-030-21568-2_17

[119] A. Basso, F. Aydin, D. Dinu, J. Friel, A. Varna, M. R. Sastry, and S. Ghosh, "Where Star Wars meets Star Trek: SABER and Dilithium on the same polynomial multiplier," *IACR Cryptol. ePrint Arch.*, p. 1697, 2021. [Online]. Available: https://eprint.iacr.org/2021/1697

[120] M. Azouaoui, O. Bronchain, G. Cassiers, C. Hoffmann, Y. Kuzovkova, J. Renes, T. Schneider, M. Schönauer, F. Standaert, and C. van Vredendaal, "Protecting Dilithium against leakage revisited sensitivity analysis and improved implementations," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 4, pp. 58–79, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i4.58-79

[121] I. Biehl, B. Meyer, and V. Müller, "Differential fault attacks on elliptic curve cryptosystems," in *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, ser. Lecture Notes in Computer Science, M. Bellare, Ed., vol. 1880. Springer, 2000, pp. 131–146. [Online]. Available: https://doi.org/10.1007/3-540-44598-6_8

[122] M. Ciet and M. Joye, "Elliptic curve cryptosystems in the presence of permanent and transient faults," *Des. Codes Cryptogr.*, vol. 36, no. 1, pp. 33–43, 2005. [Online]. Available: https://doi.org/10.1007/s10623-003-1160-8

[123] P.-A. Fouque, R. Lercier, D. Réal, and F. Valette, "Fault attack on elliptic curve Montgomery ladder implementation," in *2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2008, pp. 92–98.

[124] S. Yen, S. Kim, S. Lim, and S. Moon, "A countermeasure against one physical cryptanalysis may benefit another attack," in *Information Security and Cryptology - ICISC 2001, 4th International Conference Seoul, Korea, December 6-7, 2001, Proceedings*, ser. Lecture Notes in Computer Science, K. Kim, Ed., vol. 2288. Springer, 2001, pp. 414–427. [Online]. Available: https://doi.org/10.1007/3-540-45861-1_31

[125] S. Yen and M. Joye, "Checking before output may not be enough against fault-based cryptanalysis," *IEEE Trans. Computers*, vol. 49, no. 9, pp. 967–970, 2000. [Online]. Available: https://doi.org/10.1109/12.869328

[126] N. Bindel, J. Buchmann, and J. Krämer, "Lattice-based signature schemes and their sensitivity to fault attacks," in *2016 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2016, Santa Barbara, CA, USA, August 16, 2016*. IEEE Computer Society, 2016, pp. 63–77. [Online]. Available: https://doi.org/10.1109/FDTC.2016.11

[127] T. Espitau, P. Fouque, B. Gérard, and M. Tibouchi, "Loop-abort faults on lattice-based Fiat-Shamir and hash-and-sign signatures," in *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, ser. Lecture Notes in Computer Science, R. Avanzi and H. M. Heys, Eds., vol. 10532. Springer, 2016, pp. 140–158. [Online]. Available: https://doi.org/10.1007/978-3-319-69453-5_8

[128] ——, "Loop-abort faults on lattice-based signature schemes and key exchange protocols," *IEEE Trans. Computers*, vol. 67, no. 11, pp. 1535–1549, 2018. [Online]. Available: https://doi.org/10.1109/TC.2018.2833119

[129] M. ElGhamrawy, M. Azouaoui, O. Bronchain, J. Renes, T. Schneider, M. Schönauer, O. Seker, and C. van Vredendaal, "From MLWE to RLWE: A differential fault attack on randomized & deterministic dilithium," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 4, pp. 262–286, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i4.262-286

[130] V. Q. Ulitzsch, S. Marzougui, A. Bagia, M. Tibouchi, and J. Seifert, "Loop aborts strike back: Defeating fault countermeasures in lattice signatures with ILP," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 4, pp. 367–392, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i4.367-392

[131] D. Dinu, "Who's at Fault? A Look at Post-Quantum Cryptography and Fault Injection Attacks," https://www.intel.com/content/www/us/en/security/security-practices/blogs/fault-injection-attacks.html, 2025, accessed: September 25, 2025.

[132] S. Jendral, J. P. Mattsson, and E. Dubrova, "A single-trace fault injection attack on hedged module lattice digital signature algorithm (ML-DSA)," in *Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2024, Halifax, NS, Canada, September 4, 2024*. IEEE, 2024, pp. 34–43. [Online]. Available: https://doi.org/10.1109/FDTC64268.2024.00013

[133] L. Bettale, S. Montoya, and G. Renault, "Safe-error analysis of post-quantum cryptography mechanisms - short paper -," in *18th Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2021, Milan, Italy, September 17, 2021*. IEEE, 2021, pp. 39–44. [Online]. Available: https://doi.org/10.1109/FDTC53659.2021.00015

[134] S. Islam, K. Mus, R. Singh, P. Schaumont, and B. Sunar, "Signature correction attack on Dilithium signature scheme," in *7th IEEE European Symposium on Security and Privacy, EuroS&P 2022, Genoa, Italy, June 6-10, 2022*. IEEE, 2022, pp. 647–663. [Online]. Available: https://doi.org/10.1109/EuroSP53844.2022.00046

[135] P. Ravi, M. P. Jhanwar, J. Howe, A. Chattopadhyay, and S. Bhasin, "Exploiting determinism in lattice-based signatures: Practical fault attacks on pqm4 implementations of NIST candidates," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, AsiaCCS 2019, Auckland, New Zealand, July 09-12, 2019*, S. D. Galbraith, G. Russello, W. Susilo, D. Gollmann, E. Kirda, and Z. Liang, Eds. ACM, 2019, pp. 427–440. [Online]. Available: https://doi.org/10.1145/3321705.3329821

[136] P. Ravi, D. B. Roy, S. Bhasin, A. Chattopadhyay, and D. Mukhopadhyay, "Number ”not used” once - practical fault attack on pqm4 implementations of NIST candidates," in *Constructive Side-Channel Analysis and Secure Design - 10th International Workshop, COSADE 2019, Darmstadt, Germany, April 3-5, 2019, Proceedings*, ser. Lecture Notes in Computer Science, I. Polian and M. Stöttinger, Eds., vol. 11421. Springer, 2019, pp. 232–250. [Online]. Available: https://doi.org/10.1007/978-3-030-16350-1_13

[137] E. Krahmer, P. Pessl, G. Land, and T. Güneysu, "Correction fault attacks on randomized CRYSTALS-Dilithium," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2024, no. 3, pp. 174–199, 2024. [Online]. Available: https://doi.org/10.46586/tches.v2024.i3.174-199

[138] P. Ravi, B. Yang, S. Bhasin, F. Zhang, and A. Chattopadhyay, "Fiddling the twiddle constants - fault injection analysis of the number theoretic transform," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 2, pp. 447–481, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i2.447-481

[139] S. Amer, Y. Wang, H. Kippen, T. Dang, D. Genkin, A. Kwong, A. Nelson, and A. Yerukhimovich, "PQ-Hammer: End-to-end key recovery attacks on post-quantum cryptography using Rowhammer," in *IEEE Symposium on Security and Privacy, SP 2025, San Francisco, CA, USA, May 12-15, 2025*, M. Blanton, W. Enck, and C. Nita-Rotaru, Eds. IEEE, 2025, pp. 3567–3582. [Online]. Available: https://doi.org/10.1109/SP61157.2025.00048