Zero-Value Filtering for Accelerating Non-Profiled Side-Channel Attack on Incomplete NTT based Implementations of Lattice-based Cryptography

Tolun Tosun[1,2], Erkay Savas[1]
[1]*Sabanci University Computer Science and Engineering* [2]*Analog Devices*

***Abstract*—Lattice-based cryptographic schemes such as Crystals-Kyber and Dilithium are post-quantum algorithms selected to be standardized by NIST as they are considered to be secure against quantum computing attacks. The multiplication in polynomial rings is the most time-consuming operation in many lattice-based cryptographic schemes, which is also subject to side-channel attacks. While NTT-based polynomial multiplication is almost a norm in a wide range of implementations, a relatively new method, incomplete NTT is preferred to accelerate lattice-based cryptography, especially on some computing platforms that feature special instructions. In this paper, we present a novel, efficient and non-profiled power/EM side-channel attack targeting polynomial multiplication based on the incomplete NTT algorithm. We apply the attack on the Crystals-Dilithium signature algorithm and Crystals-Kyber KEM. We demonstrate that the method accelerates attack run-time when compared to the existing approaches. While a conventional non-profiled side-channel attack tests a much larger hypothesis set because it needs to predict two coefficients of secret polynomials together, we propose a much faster *zero-value filtering attack* (ZV-FA), which reduces the size of the hypothesis set by targeting the coefficients individually. We also propose an effective and efficient validation and correction technique employing the inverse NTT to estimate and modify the mispredicted coefficients. Our experimental results show that we can achieve a speed-up of $\mathbf{1915}\times$over brute-force.**

***Index Terms*—post-quantum cryptography; side-channel attack; correlation power analysis; multivariate mutual information analysis; crsytals dilithium; crsytals kyber;**

I. Introduction

SECURITY of public-key cryptosystems relies on the hardness of well-known mathematical problems such as the discrete logarithm problem for the elliptic curve cryptography (ECC) [1], [2] and the digital signature algorithm (DSA) [3] or the integer factorization problem for RSA [4]. While those hard problems are conjectured to be secure against known cryptanalytic algorithms running on classical computers, it has been shown that Shor's algorithm [5] can solve them in polynomial time on a large-scale quantum computer.

To address the quantum threat, the National Institute of Standards and Technology (NIST) has announced the standardization process for post-quantum public-key cryptographic algorithms (PQC) in 2016. The standardization process covers quantum-resistant digital signature schemes, and public-key encryption and key-establishment algorithms. Currently, the contest is at the fourth round with already standardized algorithms. Lattice-based schemes, based on various hard lattice problems, facilitate the construction of quantum resilient public-key cryptography with a promising level of efficiency. Among the winners, the lattice-based digital signature algorithm Crystals-Dilithium [6] is based on the Module-LWE [7] and Module-SIS (MSIS) problems, while Crystals-Kyber [8] is MLWE based key encapsulation mechanism (KEM). As Kyber and Dilithium are members of the same family, Crystals, they have several building blocks in common.

In cryptoanalysis, side-channel attacks (SCA) are the ones that target the weaknesses in implementations rather than algorithm specifications, by collecting side information such as running time or power consumption that can leak sensitive (intermediate) information during the execution of the targeted cryptographic operation. Side-channel attacks are considered as one of the main threats, particularly for embedded devices because of the simplicity of side-information collection, such as IoT chips, which sign sensor data before transmission. The Correlation Power Analysis (CPA) is proposed in [9], which models the power consumption of the device under test and measures the correlation of the model with real-world data to test secret value hypotheses. The power leakage of the device/implementation is often modeled with the Hamming Weight (HW)/Hamming Distance (HD) of/between the intermediate data. The Mutual Information Analysis (MIA) [10] is another efficient side-channel *distinguisher*, based on information theory and Shannon entropy. The Electromagnetic (EM) side-channels [11] are similar to power analysis as any attack suit designed for power leakage can be practiced with EM leakage while it can supply more precise information about the sensitive intermediate data. Masking is one of the most promising countermeasures against power/EM attacks, which randomizes the intermediate data with secret sharing so that characteristics of sensitive data are not reflected in power consumption.

JIL Rating [12] is a widely used metric to assess the complexity of side-channel attacks; the higher the JIL score, the harder to perform the attack. As the time needed to apply the attack is a factor in the overall rating, both the number of traces and the attack run-time affect the rating of the attack, constituting an important motivation for this work.

Needless to say, the side-channel security of post-quantum public cryptography is essential as well since post-quantum algorithms are intended to replace the existing public-key standards soon and the usage of public-key cryptography in embedded devices will be potentially more extensive. For example, a secure firmware update on an embedded device relies on the security of the employed digital signature while embedded devices are open to timing, and power/EM attacks by nature. With increasing interest, several attacks and countermeasures have been proposed for PQC candidates in the literature.

Polynomial multiplication is the core operation for practical constructions of lattice-based cryptography, which are based on ring-learning with errors (R-LWE) problem [13]. Most implementations utilize number theoretic transform (NTT) for efficient polynomial arithmetic [14]. A technique referred as *incomplete NTT* is introduced to handle rings of special structures as well as for efficiency [15]–[17] in implementations of various lattice-based cryptography algorithms. Our study targets the incomplete NTT operation specifically.

Table I summarizes the related side-channel attacks against lattice-based schemes from the literature. Among the attack types, the profiled class forms the majority, where we require a device identical to the one targeted by the attacker, who tries to characterize the leakage when executing a cryptographic algorithm with a known secret key. In other words, the attacker needs to have more capability in a profiled attack compared to the non-profiled class. Machine learning-based approaches are gaining popularity in the design of profiled attacks for lattice-based cryptography [18], [19]. In addition, Primas et al. [20] present a notable study by combining the side-channel leakage of NTT computation with the belief propagation algorithm to conduct a single-trace profiled attack.

As for the non-profiled class, the polynomial multiplication is the most attractive target operation [21]–[23]. Steffen et al. [23] conduct an attack on a hardware implementation of Dilithium. Chen et al. [22] target the reference implementation of Dilithium, concentrating on improving the runtime performance of non-profiled attack, as the conventional approach requires brute-force effort over 23-bit secrets based on the coefficient modulus length of Dilithium. Mujdei et al. [21] attack ARM M4 implementation of Kyber [17], which has a very similar NTT implementation with Dilithium, based on [16]. The authors of [21] show that the secret coefficients must be predicted in pairs since the incomplete NTT algorithm is used in the polynomial multiplication of the targeted implementation.

In our study, we present a more efficient non-profiled attack on the incomplete NTT implementation, which facilitates that the coefficients of the secret polynomials can be predicted individually. To show its efficacy, we use the very recent and fast implementation of Dilithium and Kyber on ARM M4 [17] as in [21]. We present several non-profiled side-channel attacks targeting the multiplication in the incomplete NTT domain and finally develop a much more efficient approach exploiting the zero-valued coefficients of the known operand of the incomplete NTT multiplication, with application to Dilithium and Kyber schemes.

A. Main Contributions

We can list our contributions as follows:

- We present a novel non-profiled power/EM attack against incomplete NTT-based implementations of polynomial multiplication in Lattice-based Cryptography, referred to as Zero-Value Filtering Attack (ZV-FA). Our approach is efficient as it decreases the number of hypotheses significantly by introducing a filtering technique based on zero-value coefficients in the known input/output polynomials of the operation targeted by the side-channel attack.
- We present an efficient validation technique for estimating and correcting mispredicted values for attacking secret polynomials with short coefficients. The method not only ensures full accuracy on the estimated secret polynomials but also accelerates the attack run time by trading off the number of traces.
- We show that using short secret key polynomials in lattice-based cryptography can be exploited to accelerate side-channel attacks.
- We implement the ZV-FA with the validation technique on the pqm4 [17] implementations of Dilithium and Kyber [16]. Our experiments demonstrate that a moderate increase in the number of traces can decrease the attack run-time significantly. It is experimentally shown through EM side-channel that, a speed-up of up to three orders of magnitude in attack run-time can be achieved over a conventional CPA targeting the polynomial multiplication.
- We experimentally show that our approach is also favorable in the presence of masking, by applying ZV-FA to a protected implementation of Kyber.

II. Notation

Matrices are represented by bold uppercase letters, such as $\mathbf{A}$, while vectors are represented by bold lowercase letters, such as $\mathbf{b}$. Sets are denoted by uppercase calligraphic letters, such as $\mathcal{A}$. Polynomials are denoted by lowercase italic letters, such as f. Depending on the context, polynomials may be represented together with their indeterminate, such as $f(x)$. Subscripts together with square brackets are used to denote element(s) of matrices and vectors, such as $\mathbf{A}_{[i,j]}$ and $\mathbf{s}_{[i]}$; elements of sets, such as $\mathcal{A}_{[i]}$; coefficients of polynomials, such as $f_{[i]}$. The notation $\mathbf{A}_{[:,j]}$ denotes the j-th column vector of $\mathbf{A}$. Similarly, $\mathcal{A}_{[:i]}$ denotes the first i elements of the set $\mathcal{A}$.

Modular reductions are performed in a centered manner. Specifically, given an integer i and a modulus q, the operation $i' = i \pmod{^{\pm}q}$ maps i to a unique integer i' in the range of $[-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$ for odd q. The set of integers modulo q with centered reduction is denoted by $\mathbb{Z}_q$. On the other hand, the notation $\mathbb{Z}_q^+$ is used to denote the set of integers modulo q using the positive range, namely $[0, q-1]$. The ring of polynomials $\mathbb{Z}_q[x]/(X^n+1)$, where elements are polynomials of the maximum degree of $n-1$, whose coefficients are modulo-q reduced, is denoted by R_q.

The dimensionality of matrices and vectors is shown in the superscript. For example, $R_q^{k \times l}$ represents a matrix of dimensions $k \times l$, whose elements are in R_q. Similarly, superscript is used for the matrices and vectors to express their dimensionality, such as $\mathbf{A}^{N \times M}$ and $\mathbf{b}^N$. The cardinality of the set $\mathcal{A}$ is denoted by $|\mathcal{A}|$. The matrix-vector multiplication of the operands $\mathbf{A}$ and $\mathbf{b}$ is denoted by $\mathbf{Ab}$. Similarly, multiplication of the vector $\mathbf{b}$ with scalar a is denoted by $a\mathbf{b}$.

Attack	Class	Algorithm	Implementation	Target Operation
this work	Non-profiled	Dilithium / Kyber	ARM M4 (masked)	polynomial multiplication
[22]	Non-profiled	Dilithium	Reference C	polynomial multiplication
[23]	Non-Profiled	Dilithium	HW	polynomial multiplication
[21]	Non-profiled	Kyber	ARM M4	polynomial multiplication
[18]	Profiled	Dilithium	Reference C	NTT
[19]	Profiled	Dilithium	Reference C	bit-unpacking
[23]	Profiled	Dilithium	HW	decoding / NTT
[24]	Profiled	Dilithium	Reference C / ARM M4	small polynomial sampling
[25]	Profiled	Dilithium	Reference C	decompose
[26]	Profiled	Kyber	Reference C / ARM M4	NTT
[20]	Profiled	R-LWE Encryption	ARM M4 (masked) [27]	NTT

TABLE I: Related Side-Channel Attacks from the Literature

Polynomial multiplication is denoted by the standard symbol $\cdot$, while element-wise multiplication of two vectors is denoted by $\odot$. In some cases, the symbol $\cdot$ is used to represent integer multiplication to support the narrative. The symbol $\times$ denotes the Cartesian product between sets, such as $\mathcal{A} \times \mathcal{B}$.

The notation (also referred to as infinity norm) $||s||_\infty$ is used to represent the maximum coefficient of the polynomial s in absolute value, whose elements are reduced in a centered manner. Similarly, $||\mathbf{s}||_\infty$ is the maximum of the maximum absolute values of coefficients of the polynomials in the vector $\mathbf{s}$. The set S_η consists of polynomials $w \in R_q$ with $||w||_\infty \leq \eta$, where η is a (relatively small) positive integer, referred to as the set of *short polynomials*. $\mathcal{B}_\eta$ is denotes the central binomial distribution over R_q, where $||w||_\infty \leq \eta$ as well for $w \leftarrow \mathcal{B}_\eta$. Another subset of R_q is denoted by B_τ, which consists of polynomials with exactly τ coefficients that are either -1 or 1, and the rest is zero. $\{0,1\}^N$ denotes the set of N-bit strings. The operator $\leftarrow$ denotes uniformly random sampling from the set on the right-hand side, such as $\rho \leftarrow \{0,1\}^N$. A prediction to a secret a is denoted by *wide hat*, such as $\widehat{a}$.

III. Side-channel Attack Overview and Distinguishers

Main steps of a non-profiled side-channel attack can be summarized as follows:

- The attacker observes N cryptographic operations involving the secret key and records the power consumption of the victim device. M points are sampled in time at each observation. Power samples are stored in the matrix $\mathbf{T}^{N\times M}$ while $\mathbf{p}^N$ is the vector of known variables.
- A point of interest (PoI) is selected by the adversary. The PoI should be a function of a known variable that changes at each experiment and the attacked secret that remains the same for all experiments.
- A set of predictions is prepared, denoted by $\mathcal{K}$. Then, intermediate value matrix $\mathbf{V}^{N\times|\mathcal{K}|}$ is computed w.r.t. each hypothesis; *i.e.*, $\mathbf{V}_{[i,j]}$ is the value of the PoI computed using $\mathbf{p}_{[i]}$ and $\mathcal{K}_{[j]}$.
- $\mathbf{V}$ is mapped to the hypothetical power consumption matrix, $\mathbf{H}^{N\times|\mathcal{K}|}$ by applying a power consumption model. In other words, $\mathbf{H}_{[i,j]} = \text{HW}(\mathbf{V}_{[i,j]})$, in case HW is chosen.
- Each hypothesis $\mathcal{K}_{[i]}$ is tested, *i.e.* scored, with the selected distinguisher, by considering the relationship between $\mathbf{H}_{[:,i]}$ and $\mathbf{T}_{[:,j_1]}$ (and $\mathbf{T}_{[:,j_2]}$ for second-order case). Usually, the maximum score over all j_1 (and j_2) is assigned.

In side-channel literature, distinguishers are statistical tools that are used to rank predictions (*i.e.*, hypotheses) for secret keys, by exploiting the dependency between the data processed by a cryptographic implementation and its power consumption. The distinguishers employed in this study are presented in the following subsections.

A. Correlation Power Analysis (CPA)

CPA [9] is a widely-used side-channel distinguisher, based on Pearson's correlation. For random variables X and Y, the correlation coefficient is estimated by

$$\hat{\rho}(X,Y) = \frac{\hat{\text{cov}}(X,Y)}{\hat{\text{std}}(X)\cdot\hat{\text{std}}(Y)}. \tag{1}$$

To utilize correlation as a side-channel distinguisher, $\hat{\rho}(\mathbf{H}_{[:,i]}, \mathbf{T}_{[:,j_1]})$ is computed.

B. Higher-Order CPA (HOCPA)

To perform a higher-order attack using CPA, multiple samples must be combined using a pre-processing function. The state-of-art pre-processing function is the normalized product [28], defined as follows for the second-order case

$$\mathbf{T}'_{[\kappa,j_1\times j_2]} = (\mathbf{T}_{[\kappa,j_1]} - \overline{\mathbf{T}_{[:,j_1]}})\cdot(\mathbf{T}_{[\kappa,j_2]} - \overline{\mathbf{T}_{[:,j_2]}}) \text{ for } 0 \leq \kappa < N \tag{2}$$

which combines the leakage at time samples j_1 and j_2. Then, $\hat{\rho}(\mathbf{H}_{[:,i]}, \mathbf{T}'_{[:,j_1\times j_2]})$ is computed.

C. Mutual Information Analysis (MIA)

MIA [10] is an information-theoretic side-channel distinguisher. Let $\text{H}(X)$ denote the entropy of a random variable X on a discrete space $\mathcal{X}$, and let $\text{H}(X|Y)$ denote the conditional entropy for X and another random variable Y on a discrete space $\mathcal{Y}$. $\text{P}(\cdot)$ denotes the probability function. The mutual information between X and Y is defined as

$$\begin{aligned} \text{I}(X;Y) &= \text{H}(X) - \text{H}(X|Y) \\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \text{P}(X=x,Y=y)\cdot \\ &\quad \log_2\Big(\frac{\text{P}(X=x,Y=y)}{\text{P}(X=x)\text{P}(Y=y)}\Big) \end{aligned} \tag{3}$$

	NIST Security Level	2	3	5
Parameter	Meaning			
τ	number of ±1 in c	39	49	60
k,l	dimensions of $\mathbf{A}$	(4,4)	(6,5)	(8,7)
η	coefficient range of $\mathbf{s}_1$,$\mathbf{s}_2$	2	4	2
	Expected #repetitions	4.25	5.1	3.85

TABLE II: Dilithium parameter set

Verbally, the entropy in X not covered by Y, namely $\mathrm{H}(X|Y)$, is subtracted from $\mathrm{H}(X)$ to formulate the mutual information in between. Similar to Section III-A, $\mathrm{I}(\mathbf{H}_{[:,i]}; \mathbf{T}_{[:,j_1]})$ is computed in side-channel analysis.

D. Multivariate Mutual Information Analysis (MMIA)

MMIA [29] is a generalization of MIA to high-order side-channel attacks. The multivariate mutual information between three random variables X, Y, and Z is given by

$$\mathrm{I}(X;Y;Z) = \mathrm{I}(Y;Z) - \mathrm{I}(Y;Z|X) \tag{4}$$

where

$$\mathrm{I}(Y;Z|X) = \sum_{x\in\mathcal{X}} \mathrm{P}(X=x)\cdot \mathrm{I}(Y;Z|X=x) \tag{5}$$

By plugging two power samples in time into the above equation, a second-order side-channel distinguisher is achieved, namely $\mathrm{I}(\mathbf{H}_{[:,i]}; \mathbf{T}_{[:,j_1]}; \mathbf{T}_{[:,j_2]})$.

IV. Lattice-based Post-quantum Cryptography

In this section, we present the lattice-based post-quantum cryptography algorithms, Dilithium and Kyber.

A. Dilithium

Crystals: Dilithium [6] is a lattice-based post-quantum digital signature scheme based on the hardness of MLWE and MSIS problems. It operates over the ring of polynomials R_q with dimension $n = 256$ and $q = 8380417 = 2^{23} - 2^{13} + 1$. The rest of the parameter set used by Dilithium can be found in Table II. Dilithium's key generation creates an MLWE instance by the equation $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$. The matrix of polynomials $\mathbf{A} \in R_q^{k\times l}$ is part of the public key as well as the vector of polynomials $\mathbf{t} \in R_q^k$ (in the full scheme, lower bits of coefficients of polynomials in $\mathbf{t}$ are kept secret). The vectors of short polynomials $\mathbf{s}_1 \in S_\eta^l$, $\mathbf{s}_2 \in S_\eta^k$ forms the secret key.

The template of Dilithium's signature generation, given in Algorithm 1, applies the rejection sampling idea. Most of the signature procedure is implemented in a loop, which iterates until a valid and secure signature is found. The parameters are chosen such that the expected number of repetitions is small, as presented in Table II. Inside the signature loop, a challenge polynomial $c \in B_\tau$ is generated. The candidate signature $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$ is rejected and restarted in case the security and correctness checks at line 7 fail. We would like to note that, there is no difference between the template and the standard Dilithium from the perspective of this work.

Algorithm 1 Dilithium.Sign(sk, M)

```
1: $\mathbf{z} := \perp$
2: while $\mathbf{z} = \perp$ do
3:     $\mathbf{y} \leftarrow \widetilde{S}_{\gamma_1}^l$
4:     $\mathbf{w}_1 := \mathrm{HighBits}(\mathbf{Ay}, 2\gamma_2)$
5:     $c \in B_\tau := \mathrm{H}(\mu \mid\mid \mathbf{w}_1)$
6:     $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$
7:     if $||\mathbf{z}||_\infty \geq \gamma_1 - \beta$ or $||\mathrm{LowBits}(\mathbf{Ay} - c\mathbf{s}_2)||_\infty \geq \gamma_2 - \beta$ then
8:         $(\mathbf{z}) := \perp$
9: return $\sigma = (\mathbf{z}, c)$
```

	NIST Security Level	1	3	5
Parameter	Meaning			
k	dimensions of $\mathbf{A}$	2	3	4
η	Parameter of CBD $\mathcal{B}_\eta$	2	4	2

TABLE III: Kyber parameter set

B. Kyber

Crystals: Kyber [6] is a lattice-based post-quantum KEM signature scheme based whose security relies on the computational difficulty of the MLWE problem. Kyber's ring dimension $n = 256$ and the coefficient modulus $q = 3329 = 2^{11} + 2^{10} + 2^8 + 1$ for the operated ring of polynomials R_q. The rest of the related parameter set can be found in Table III. Similar to Dilithium, the public-private key pair of Kyber is generated by the MLWE equation $\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e}$, where $\mathbf{s} \in R_q^k$ is the secret key, $\mathbf{t} \in R_q^k$ and $\mathbf{A} \in R_q^{k\times k}$ forms the public key and $\mathbf{e} \in R_q^k$ is the noise vector. $\mathbf{s}$ and $\mathbf{e}$ are short polynomials, whose coefficients are sampled from the central binomial distribution $\mathcal{B}_\eta$.

Algorithm 2 Kyber.CPAPKE.Dec($\mathbf{s}$, ct=($\mathbf{u}$,v))

```
return $\mathrm{Compress}_q(v - \mathbf{s}^T\mathbf{u}, 1)$
```

Algorithm 2 presents a simplified version of Kyber's key decapsulation. The ciphertext ct is a composition of two parts; $\mathbf{u} \in R_q^k$ and $v \in R_q$. The function $\mathrm{Compress}_q$ maps from R_q to $\{0,1\}^n$.

V. Number Theoretic Transform (NTT)

Number Theoretic Transform (NTT) is a special form of Fast-Fourier Transform (FFT) that operates over $\mathbb{Z}_q$ instead of complex numbers $\mathbb{C}$. NTT allows efficient multiplication of polynomials over R_q. Representing a polynomial $a(x) \in R_q$ in the NTT domain can be viewed as an application of Chinese Remainder Theorem (CRT). Polynomial multiplication is achieved by element-wise multiplying the NTT representations of the operands:

$$a(x)\cdot b(x) = \mathrm{NTT}^{-1}(\mathrm{NTT}(a(x)) \odot \mathrm{NTT}(b(x))) \tag{6}$$

Most lattice-based crypto systems, including Dilithium, operates over the ring R_q. NTT in R_q requires $q \equiv 1 \pmod{2n}$, which ensures a primitive $2n$-th root of unity ζ_{2n} exists in $\mathbb{Z}_q$ for which $\zeta_{2n}^n = -1 \mod q$, referred to as *negacyclic NTT*. In case n is a power of 2, NTT can be computed efficiently by splitting the polynomial to half of its size in recursive manner until linear degree is reached. The transformation at each layer can be efficiently implemented using *Cooley-Tukey* (CT) butterfly circuit [30]. For degree-$n/2^i$ polynomial $a(x) = a_0(x) + a_1(x) \cdot x^{n/2^{i+1}}$, the CT butterfly is defined by the map

$$a_0(x)+a_1(x)\cdot x^{n/2^{i+1}} \rightarrow (a_0(x)-\delta\cdot a_1(x), a_0(x)+\delta\cdot a_1(x)).$$

where δ is an odd power of ζ_{2n}, called the *twiddle factor*. In this manner, full NTT computation requires $\log n$ layers. Most applications use *Gentleman-Sande* (GS) butterfly for computing the inverse NTT [31], although it is not necessarily required. Using CT or GS when n is a power of 2, computing NTT / inverse NTT takes $\Theta(n \log n)$ steps.

A. Incomplete NTT

For efficiency NTT can be computed for $m < \log n$ layers so polynomials are recursively splitted to degree-$n/2^m$ polynomial components, denoted by $\mathrm{NTT}_m(a(x))$. The prerequisite for NTT_m is to have $q \equiv 1 \pmod{\frac{n}{2^{\log n - m - 1}}}$ for the negacyclic NTT [32].

Let $\mathbf{a} = \mathrm{NTT}_m(a(x))$ and $\mathbf{b} = \mathrm{NTT}_m(b(x))$. $\mathbf{a}$ and $\mathbf{b}$ are 2^m-dimensional vectors and $\mathbf{a}_{[i]}, \mathbf{b}_{[i]} \in \mathbb{Z}_q[x]/(x^{n/2^m} - \delta)$ for $i < 2^m$, where δ is some power of ζ_{2n}. Then, $a(x) \cdot b(x)$ can be computed through $\mathbf{a} \odot \mathbf{b}$ as in Equation 6. In this case, element-wise multiplication refers to the multiplication of polynomials of degree $n/2^m - 1$, which is mostly implemented by the school-book algorithm, and there are 2^m such multiplications. For instance, when $m = \log(n/2)$, $n/2$ multiplications of degree-1 polynomials is performed, as presented in Algorithm 3.

Algorithm 3 Incomplete NTT Multiplication($\mathbf{s}'$, $\mathbf{c}'$)

```
for $i \leftarrow 0$ until $n/2$ do
    $\mathbf{r}_{[i,0]} \leftarrow \mathbf{s}'_{[i,0]} \cdot \mathbf{c}'_{[i,0]} + \mathbf{s}'_{[i,1]} \cdot \mathbf{c}'_{[i,1]} \cdot \delta_i \mod q$
    $\mathbf{r}_{[i,1]} \leftarrow \mathbf{s}'_{[i,0]} \cdot \mathbf{c}'_{[i,1]} + \mathbf{s}'_{[i,1]} \cdot \mathbf{c}'_{[i,0]} \mod q$
return r
```

VI. Target Implementation and Point of Interest (PoI)

We focus on the side-channel attack against incomplete NTT-based polynomial multiplication (Algorithm 3). One of the inputs is secret, and the attacker wants to learn it through a side-channel attack while one of the inputs is public. This case is directly applicable to Kyber and Dilithium, regarding the operations $\mathbf{s}^T\mathbf{u}$ and $c\mathbf{s}_1$ (and $c\mathbf{s}_2$), respectively. Needless to say, the attack aims to retrieve $\mathbf{s}$ for Kyber and $c\mathbf{s}_1$ and $c\mathbf{s}_2$ for Dilithium. To simplify notation, we denote the attacked polynomial and its NTT representation by s' and $\mathbf{s}'$, respectively. Note that, our attack is identical over the polynomials in the vector of polynomials $\mathbf{s}$, as well as the polynomials in $\mathbf{s}_1$ and $\mathbf{s}_2$ and it is simply repeated to retrieve all polynomials in those secret vectors. Similarly, c' denotes the known polynomial, and $\mathbf{c}'$ denotes its NTT representation. It corresponds to the challenge polynomial c, which is among the output of the signature (Algorithm 1) for Dilithium. On the other hand, the public polynomial is any element of the $\mathbf{u}$ part of the ciphertext for Kyber, which is an input of the key decapsulation (Algorithm 2). Again, our methodology is identical for any polynomial in $\mathbf{u}$.

Recall that, coefficient modulus $q = 3329$ and the ring dimension $n = 256$ for Kyber permits incomplete NTT of maximum $\log(n/2)$ layers. On the other hand, the specified coefficient modulus of Dilithium does not require the NTT to be incomplete. However, incomplete NTT can be preferred by the implementers to enhance performance [16]. Specifically, the multiplications $c\mathbf{s}_1$ and $c\mathbf{s}_2$ are referred to as *small NTT* and performed using a *carrier prime*. Although carrier primes are usually denoted by q', we will use q to unify our notation. The small NTT operates with the prime $q = 257$ for Dilithium2 and Dilithium5 while $q = 769$ is chosen for Dilithium3. The rationale behind the small NTT is that, coefficients of $c\mathbf{s}_1$ and $c\mathbf{s}_2$ does not exceed $\tau\eta$ in absolute value. Recall that coefficient range of the polynomials in $\mathbf{s}_1$ and $\mathbf{s}_2$ is $[-\eta, \eta]$ while c has exactly τ coefficients equal to ± 1 and the rest of the coefficients are 0.

The target implementation is the pqm4 library [17], which is mostly based on the work [16] In the mentioned implementation, $\mathbf{r}_{[i,j]}$ (see Algorithm 3) are written to the memory while the intermediate steps of the computation remain in the processor. We consider the store instructions for $\mathbf{r}_{[i,j]}$ as the PoI, assuming a memory operation leads to a power leakage with a greater signal-to-noise-ratio (SNR) compared to the register updates or combinational logic. We use the HW model for the hypothetical power consumption computation.

VII. Proposed Side-Channel Attacks

In this section, we first show a straightforward baseline attack and then give the details of a more efficient zero-value filtering attack. Figure 1 presents a high-level overview of the side-channel attacks discussed in this section.

A. The Baseline Attack

As each output coefficient $\mathbf{r}_{[i,j]}$, chosen as the PoI, depends on both $\mathbf{s}'_{[i,0]}$ and $\mathbf{s}'_{[i,1]}$, the *baseline scheme* is formed as a conventional non-profiled attack using one of the distinguishers presented in Section III that predicts $\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}$, simultaneously [21]. Consequently, q^2 hypotheses are tested by the baseline scheme.

We would like to make a note regarding the number of hypotheses for the attack on Dilithium. When the NTT multiplication function (Algorithm 3) is called, the coefficients $\mathbf{s}'_{[i,j]}$ can be larger than the modulus q, due to the so-called lazy reductions. In particular, they are in the range $[-7q-\eta, 7q+\eta]$, [16], [22], [33]. Fortunately, it is sufficient to predict in the set of residues, $\mathbb{Z}_q$, due to two main factors [21], [22]: 1) A trivial mathematical fact is that the coefficients are indeed in $\mathbb{Z}_q$, which is sufficient to compute the inverse NTT to obtain the coefficients of the secret polynomial. 2) For the selected PoI function, the integers in $[-7q-\eta, 7q+\eta]$, that are in the same congruence class modulo q, mostly result in the same

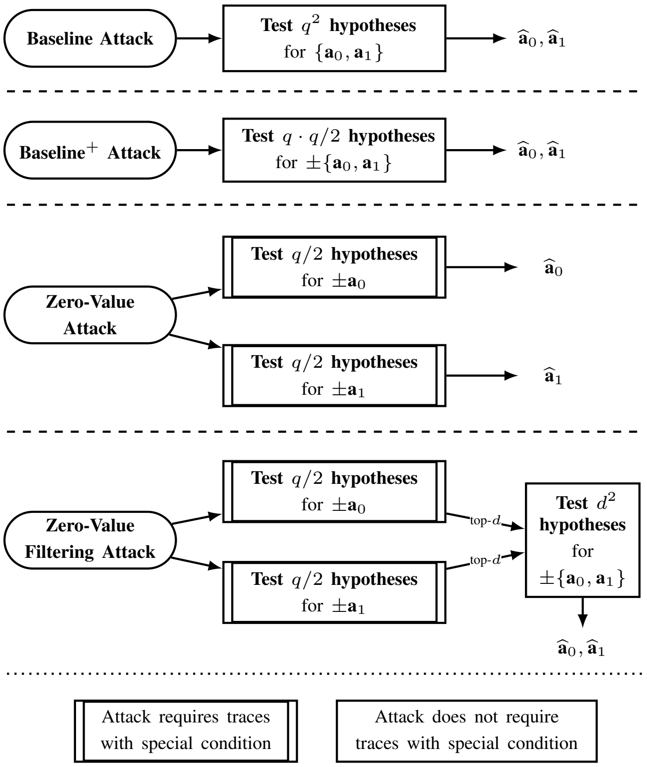

Baseline Attack
Test q^2 hypotheses for $\{\mathbf{a}_0, \mathbf{a}_1\}$
$\widehat{\mathbf{a}}_0, \widehat{\mathbf{a}}_1$
Baseline$^+$ Attack
Test $q \cdot q/2$ hypotheses for $\pm\{\mathbf{a}_0, \mathbf{a}_1\}$
$\widehat{\mathbf{a}}_0, \widehat{\mathbf{a}}_1$
Zero-Value Attack
Test $q/2$ hypotheses for $\pm\mathbf{a}_0$
$\widehat{\mathbf{a}}_0$
Test $q/2$ hypotheses for $\pm\mathbf{a}_1$
$\widehat{\mathbf{a}}_1$
Zero-Value Filtering Attack
Test $q/2$ hypotheses for $\pm\mathbf{a}_0$
Test $q/2$ hypotheses for $\pm\mathbf{a}_1$
top-d
top-d
Test d^2 hypotheses for $\pm\{\mathbf{a}_0, \mathbf{a}_1\}$
$\widehat{\mathbf{a}}_0, \widehat{\mathbf{a}}_1$
Attack requires traces with special condition
Attack does not require traces with special condition

Fig. 1: Overview of the presented attacks. $\{\mathbf{a}_0, \mathbf{a}_1\} \in \mathbb{Z}_q^2$ denotes the attacked pair of secret coefficients, $\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}$.

output. Therefore, the PoI that are calculated for integers of the same congruence class are correlated with each other. This situation does not exist in Kyber as the secret polynomials are stored in the NTT domain by algorithm definition so the coefficients are precisely in $\mathbb{Z}_q$.

The computational complexity of the baseline scheme is, therefore, $\Theta(q^2(n/2))$. q^2 is approximately 16.01-bit for Dilithium2 and Dilithium5, 19.17-bit for Dilithium3, while it is 23.4-bit for all security levels of Kyber. The baseline is an accurate, yet inefficient attacking scheme in terms of the attack run-time. Therefore, we seek more efficient methods to accelerate the attack time in the next section.

B. Using Negative Correlation

It is possible to further narrow down the hypothesis set presented in the preceding section by taking advantage of the negative correlation [22]. This is due to the fact that the Hamming weights of an integer and its additive inverse in 2's complement notation are inversely correlated.

Note that, $\mathbf{V}_{:,\{-\alpha_0,-\alpha_1\}} = -\mathbf{V}_{:,\{\alpha_0,\alpha_1\}}$, for any $\{\alpha_0, \alpha_1\} \in \mathbb{Z}_q \times \mathbb{Z}_q$; recalling that $\mathbf{V}_{:,\{\alpha_0,\alpha_1\}}$ is the intermediate value vector computed w.r.t. $\{\alpha_0, \alpha_1\}$. The statistical properties of Hamming Weight suggest that $\mathbf{H}_{:,\{-\alpha_0,-\alpha_1\}}$ correlates with $\mathbf{H}_{:,\{\alpha_0,\alpha_1\}}$. Therefore, $\pm\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}$ is retrieved by testing either the hypothesis set $\mathbb{Z}_q \times \mathbb{Z}^+_{\lceil q/2 \rceil}$ or $\mathbb{Z}^+_{\lceil q/2 \rceil} \times \mathbb{Z}_q$, leading to $q \cdot q/2$ predictions.

Lastly, we need a distinguisher to tell the difference between the actual key and its additive inverse as the attacker can get either one of them. If the sign of the peak on correlation scores is positive we conclude that the actual key is found. Otherwise, the additive inverse of the hypothesis is computed as the output of the attack. We would like to note that, the actual device leakage inversely correlates with the HW of intermediate data in some cases such as when the data-bus is pre-charged with all 1's. Then, the behavior of the explained distinguisher is reversed.

The improved baseline scheme that takes advantage of negative correlation is denoted by baseline$^+$, dropping the attack complexity by 1-bit to $\Theta(q(q/2)(n/2))$.

C. Decreasing the Number of Hypotheses: Zero-Value Attack

A more practical scheme in terms of the attack run-time can be constructed by attacking the coefficients $\mathbf{s}'_{[i,0]}$ and $\mathbf{s}'_{[i,1]}$ individually, referred here as *Zero-Value (ZV) Attack*. To achieve this, we need to eliminate the effect of one of the secret coefficients from the reduction step during the NTT multiplication, whose output constitutes the chosen PoI. This can be accomplished by including only the traces to the attack that contain zeros in their coefficients, which multiply one of the secret coefficients. Consider line 2 of Algorithm 3 to develop intuition for the proposed method. Assume $\mathbf{c}'_{[i,1]} = 0 \mod q$ for some $0 \le i < n/2$, then $\mathbf{s}'_{[i,1]} \cdot \mathbf{c}'_{[i,1]} \cdot \delta_i \mod q$ becomes 0 and $\mathbf{r}_{[i,0]} = \mathbf{s}'_{[i,0]} \cdot \mathbf{c}'_{[i,0]} \mod q$. With sufficient number of traces meeting the condition $\mathbf{c}'_{[i,1]} = 0 \mod q$, predictions on $\mathbf{s}'_{[i,0]}$ can be made independently from $\mathbf{s}'_{[i,1]}$ for the specific value of i. The prerequisites for the ZV attack are referred to as *zero-value condition*s. Table IV lists the four attacking scenarios that can be adopted. For instance, to attack $\mathbf{s}'_{[i,0]}$, we need the condition $\mathbf{c}'_{[i,1]} = 0 \mod q$ and use $\mathbf{c}'_{[i,0]}$ or $\mathbf{c}'_{[i,0]} = 0 \mod q$ and use $\mathbf{c}'_{[i,1]}$. As the conditions are identical with attack scenarios 1 and 4, both $\mathbf{s}'_{[i,0]}$ and $\mathbf{s}'_{[i,1]}$ will show peaks in the results.

Attacking Scenario	Target	Condition	Used Meta	Probability
1	$\mathbf{s}'_{[i,0]}$	$\mathbf{c}'_{[i,1]} = 0$	$\mathbf{c}'_{[i,0]}$	0.0039
2	$\mathbf{s}'_{[i,0]}$	$\mathbf{c}'_{[i,0]} = 0$	$\mathbf{c}'_{[i,1]}$	0.0039
3	$\mathbf{s}'_{[i,1]}$	$\mathbf{c}'_{[i,0]} = 0$	$\delta_i \cdot \mathbf{c}'_{[i,1]}$	0.0039
4	$\mathbf{s}'_{[i,1]}$	$\mathbf{c}'_{[i,1]} = 0$	$\mathbf{c}'_{[i,0]}$	0.0039

TABLE IV: ZV-Attack Scenarios. Probabilities are equal to $1/n$.

This approach leads to an attack complexity of $\Theta(qn)$ and $\Theta((q/2)n)$ without and with the negative correlation trick, respectively, from the previous section. The drawback of this method is the hardness of finding traces meeting the mentioned conditions. We mark a trace and the corresponding known polynomial c as valid for the attack if at least one coefficient in $\mathbf{c}$ is 0; namely,

$$\mathbf{c}'_{[0,0]} = 0 \lor \mathbf{c}'_{[0,1]} = 0 \lor .. \\ .. \lor \mathbf{c}'_{[n/2-1,0]} = 0 \lor \mathbf{c}'_{[n/2-1,1]} = 0 \mod q \qquad (7)$$

The probability for a random $\mathbf{c}'$ to be valid depends on q and n, and can be computed by the following

$$1 - \Big(\frac{q-1}{q}\Big)^n, \qquad (8)$$

which leads to 0.283 for Dilithium and 0.074 for Kyber. Recall that, Dilithium's signature function applies rejection sampling, which means the target operation $c\mathbf{s}_1$ is performed several times for each signature generation, with challenge polynomials that are thrown away since the corresponding signature is rejected. However, one can retrieve the unused challenge polynomials through another side-channel attack and include them in the attack to $c\mathbf{s}_1$. We would like to note that, c is usually unprotected in the existing works from literature [34] [35] and the same assumption is made by [36]. As the NTT of c is computed, it can be attacked as presented in [20], which makes use of the belief-propagation algorithm to attack NTT transformation. Moreover, the coefficients of c are in $\{-1, 0, 1\}$ so it would be relatively easier to distinguish between those. We leave this application as a future work. By multiplying the expected number of iterations presented in Table II by the probability 0.283, we find out that each signature operation contains 1.2, 1.44, 1.09 valid challenges, *i.e.* known polynomials, on average. On the other hand, $\mathbf{u}$ is produced by the key encapsulation function of Kyber, whose inputs are publicly known. Therefore, an attacker can brute-force the seeds used by key encapsulation to find $\mathbf{u}$ that satisfies the validity condition given in Equation 7.

The individual probabilities for the coefficients $\mathbf{c}'_{[i,j]}$ being $0 \mod q$, for a valid c' are another crucial factor of attack performance. Table IV lists the probabilities for the aforementioned conditions, which is $1/n = 0.0039$ for any i and j. Note that, each $\mathbf{s}'_{[i,j]}$ is attacked with the ones ensuring the corresponding zero-value conditions among the collected traces. The listed probabilities suggest that the conditions are not met very often. Intuitively, assuming the SNR in $\mathbf{T}$ requires 200 traces for the attack to converge, then the attacker must perform approximately 51.2K measurements on the victim's device considering the probabilities of conditions in Table IV. Although the attacking phase of the presented scheme is significantly faster than the baseline by a factor of $q/2$, a more optimal strategy exists, in terms of both the number of traces and attack run-time, as presented in the next section.

D. Decreasing the Number of Traces: Zero-Value Filtering

While the ZV scheme introduced in Section VII-C requires a large number of traces to retrieve the correct key exactly, alternatively having the correct key fall in top-d candidate list is relatively inexpensive in terms of the number of traces, depending on the value of d. Therefore, the ZV attack method can be used as a filtering mechanism for the following hypothesis testing, forming a two-stage attacking scheme, referred to as *Zero-Value Filtering Attack* (ZV-FA), which is formalized in Figure 2.

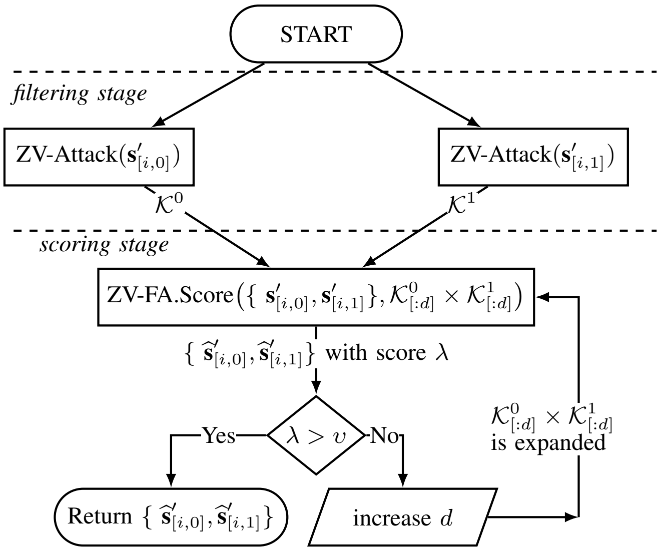

START
filtering stage
ZV-Attack($\mathbf{s}'_{[i,0]}$)
ZV-Attack($\mathbf{s}'_{[i,1]}$)
$\mathcal{K}^0$
$\mathcal{K}^1$
scoring stage
ZV-FA.Score($\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}, \mathcal{K}^0_{[:d]} \times \mathcal{K}^1_{[:d]}$)
$\{\widehat{\mathbf{s}}'_{[i,0]}, \widehat{\mathbf{s}}'_{[i,1]}\}$ with score λ
$\lambda > \upsilon$
Yes
No
$\mathcal{K}^0_{[:d]} \times \mathcal{K}^1_{[:d]}$ is expanded
Return $\{\widehat{\mathbf{s}}'_{[i,0]}, \widehat{\mathbf{s}}'_{[i,1]}\}$
increase d

Fig. 2: Flowchart of ZV-FA($\mathbf{s}'_{[i]}$)

In the first stage (a.k.a. *filtering stage*), $\mathbf{s}'_{[i,0]}$ and $\mathbf{s}'_{[i,1]}$ are attacked individually using the ZV attack scheme as presented in the preceding section. The outcomes of these ZV attacks are denoted by $\mathcal{K}^0$ and $\mathcal{K}^1$, the sorted set of predictions for $\mathbf{s}'_{[i,0]}$ and $\mathbf{s}'_{[i,1]}$, respectively, based on the scores assigned by the employed distinguisher in ZV attacks. Then, in the second stage (a.k.a. *scoring stage*), a set of predictions $\mathcal{K} = \mathcal{K}^0_{[:d]} \times \mathcal{K}^1_{[:d]}$ of size d^2 is formed for the pair $\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}$. Notice that $\mathcal{K}^0_{[:d]}$ ($\mathcal{K}^1_{[:d]}$) stands for the top scoring d predictions in $\mathcal{K}^0$ ($\mathcal{K}^1$). Afterward, $\mathcal{K}$ is scored by one of the distinguishers presented in Section III. Compared to the baseline scheme, a relatively small number of hypotheses, d^2 is used for attacking $\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}$, as opposed to q^2.

By the filtering stage, this method assumes that the correct key is in the top-d list of predictions of the highest scores for the ZV attack. A threshold mechanism, denoted by υ, validates the assumption through the attack output. The value of d is iteratively increased and naturally $\mathcal{K}^0_{[:d]}$ and $\mathcal{K}^1_{[:d]}$ get larger, until a prediction scoring greater than υ is found. By increasing d, more candidates are evaluated by the second stage, which increases the probability of having the correct key in the top-d list; naturally, the second stage takes longer to evaluate more candidates. A trivial strategy for increasing d can be doubling it. Intuitively, doubling d is acceptable for Dilithium as q is relatively small, regarding the RAM usage and response times from scoring each set of candidates. However, reducing the rate of increasing d for Kyber can be desirable, as the search space can grow quite large, considering q^2 is 23.4-bit. We explicitly state how d is updated in our experiments in Section VIII. Needless to say, the evaluated candidates from previous trials are not included during the attack. The attack becomes identical to the baseline attack, for which the threshold is not taken into account if the scores remain below υ until d covers the whole search space.

The negative correlation trick presented in Section VII-B is also applied to ZV-FA both in the filtering stage and the scoring stage. Therefore, the number of hypotheses to be tested by ZV attacks in the filtering stage is $q/2$. Then, for each $\alpha \in \mathcal{K}^0$, we insert $-\alpha$ to $\mathcal{K}^0$, with the same rank as α. In this manner, either $\{\mathbf{s}'_{[i,0]}, \mathbf{s}'_{[i,1]}\}$ or its additive inverse $\{-\mathbf{s}'_{[i,0]}, -\mathbf{s}'_{[i,1]}\}$ is retrieved through the hypothesis testing in the scoring stage. The sign is corrected by observing the sign of the peak in correlation results as in the baseline$^+$ scheme.

Compared to the ZV attack scheme, the new ZV-filtering attack is more effective with a significantly smaller number of traces. The number of traces included in the filtering stage is denoted by N_f, while the second stage can be carried out without the zero-value conditions. As a result, it can be carried out with a sufficient number of traces to ensure that its output is reliable rather than using the entire set of valid traces, which is excessive for evaluating the score.

E. Improving ZV-FA by Using Inverse NTT to Validate Predictions

The zero-value filtering attack introduced in Section VII-D relies on the assumption that a precise threshold can be found for all attacks on $\mathbf{s}'_{[i]}$ for $0 \leq i < n/2$, which, however, may not hold in practice as a non-profiled attack is performed blindly. A possible solution to this problem is to use a conservative threshold. However, this approach is computationally expensive, and a conservative threshold can still result in false positives, albeit with a lower probability. Therefore, the attacker needs to verify the found secrets, $\mathbf{s}_1$, $\mathbf{s}_2$ for Dilithium, $\mathbf{s}$ for Kyber, by the MLWE equation presented in Section IV-A and Section IV-B, respectively. Note that this verification can only be performed after all the mentioned secrets have been attacked in all vector indices.

A more reasonable strategy for the attacker is to make use of the fact that $\text{NTT}^{-1}(\mathbf{s}')$ is a short polynomial. Recall that, for Dilithium, $\mathbf{s}_1 \in S_\eta^l$, $\mathbf{s}_2 \in S_\eta^k$, whose coefficients are in the range $[-\eta, \eta]$. Similarly, for Kyber, $\mathbf{s}$ is sampled from $\mathcal{B}_\eta$, for which the coefficients are in the range $[-\eta, \eta]$, as well. Therefore, a small mistake in the prediction will diffuse through the inverse NTT computation and ruin the coefficients of the output polynomial, empowering the attacker to efficiently validate the attack output.

Figure 3 illustrates the flowchart of the ZV-FA from a higher-level perspective with validation using the inverse NTT. Let $\widehat{\mathbf{s}}'$ denote the vector of polynomials, a prediction to $\mathbf{s}'$, formed after completing the individual ZV attacks to $\mathbf{s}'_{[i,0]}$ and $\mathbf{s}'_{[i,1]}$ for all i at Step 1. To validate $\widehat{\mathbf{s}}'$, $\text{NTT}^{-1}(\widehat{\mathbf{s}}')$ is computed and the shortness property is sought in the resulting polynomial. If the found polynomial is not validated the mispredicted pair of coefficients in $\widehat{\mathbf{s}}'$ is approximated and re-attacked to correct it. The approximation is performed by selecting the pair of coefficients with the minimum score. The current score for the prediction $\widehat{\mathbf{s}}'_{[i]}$ is denoted by λ_i. Observe that the outputs of ZV attacks are immediately scored at Step 2 if the shortness check fails. This initial scoring step is needed to be able to compare the ZV scores with ZV-FA scores, which are originally in distinct scales. The same distinguisher with ZV-FA.Score is applied to $\widehat{\mathbf{s}}'_{[i]}$ using the top scorer candidates from Step 1, $\mathcal{K}^{i,0}_{[0]}$ and $\mathcal{K}^{i,1}_{[0]}$, with the same number of traces to get a comparable score with ZV-FA. In this manner, the attack terminates without re-attacking predictions which are already correctly predicted by ZV. Note that $\mathcal{K}^{i,0}$ ($\mathcal{K}^{i,1}$) is the set of predictions for $\mathbf{s}'_{[i,0]}$ ($\mathbf{s}'_{[i,1]}$) sorted for ZV attack scores, by slightly modifying our notation from the previous section as the coefficient index i is added to superscript.

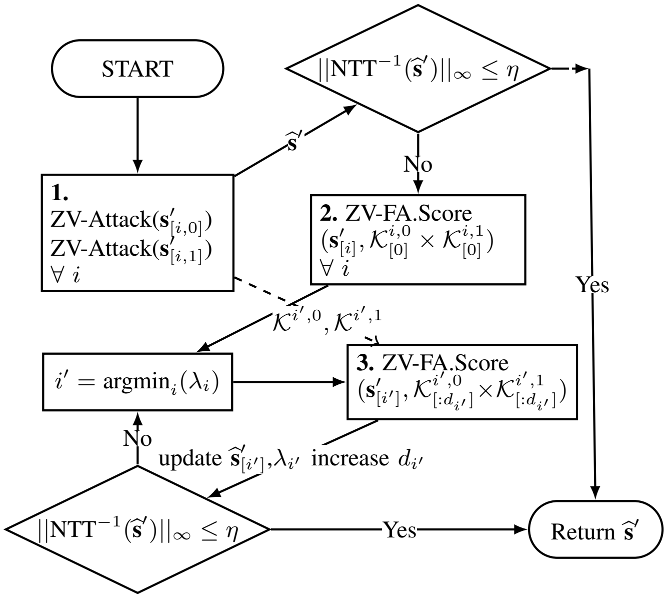

START
$||\text{NTT}^{-1}(\widehat{\mathbf{s}}')||_\infty \leq \eta$
$\widehat{\mathbf{s}}'$
No
Yes
1. ZV-Attack($\mathbf{s}'_{[i,0]}$) ZV-Attack($\mathbf{s}'_{[i,1]}$) $\forall\ i$
2. ZV-FA.Score ($\mathbf{s}'_{[i]}, \mathcal{K}^{i,0}_{[0]} \times \mathcal{K}^{i,1}_{[0]}$) $\forall\ i$
$\mathcal{K}^{i',0}, \mathcal{K}^{i',1}$
$i' = \text{argmin}_i(\lambda_i)$
3. ZV-FA.Score ($\mathbf{s}'_{[i']}, \mathcal{K}^{i',0}_{[:d_{i'}]} \times \mathcal{K}^{i',1}_{[:d_{i'}]}$)
No
update $\widehat{\mathbf{s}}'_{[i']}$,$\lambda_{i'}$ increase $d_{i'}$
$||\text{NTT}^{-1}(\widehat{\mathbf{s}}')||_\infty \leq \eta$
Yes
Return $\widehat{\mathbf{s}}'$

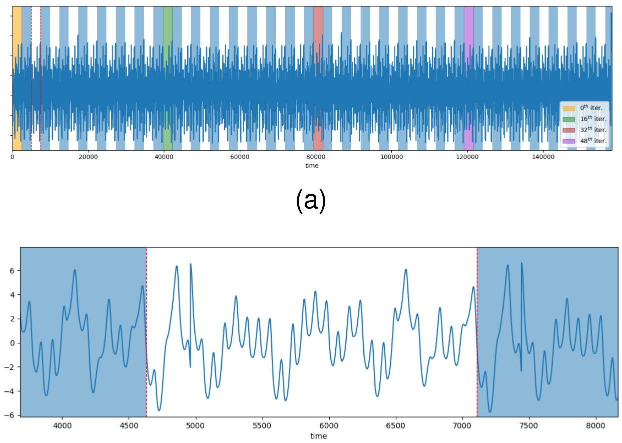
Fig. 3: ZV-FA for the whole vector of polynomials $\mathbf{s}'$ with the application of inverse NTT validation

The index of the minimum scoring pair from $\widehat{\mathbf{s}}'$ is found by computing $i' = \text{argmin}_i(\lambda_i)$ and ZV-FA.Score is performed on $\mathbf{s}'_{[i']}$ at Step 3 to replace $\widehat{\mathbf{s}}'_{[i']}$. Here, we skip the filtering stage of ZV-FA (see Figure 2) because it is indeed performed at Step 1 and therefore $\mathcal{K}^{i',0}$ and $\mathcal{K}^{i',1}$ are known. $\widehat{\mathbf{s}}'_{[i']}$ is updated if a better scoring prediction is found at the current invocation of ZV-FA.Score. At the same time, $d_{i'}$ is updated as in Figure 2 (Again, the coefficient index i is included in the notation to differentiate between d for $\mathbf{s}'_{[i]}$). Notice that the ZV-FA.Score can be performed for the same $\mathbf{s}'_{[i]}$ multiple times. However, the scoring is performed with distinct d_i at each invocation of ZV-FA.Score to expand the search for the actual secret. Recall that from the previous section, predictions for $\mathbf{s}'_{[i]}$ that are evaluated previously are not included in the attack. The prediction $\widehat{\mathbf{s}}'_{[i]}$ is guaranteed to be corrected by subsequent applications of ZV-FA.Score if the correct value for $\mathbf{s}'_{[i]}$, is the top-scorer among $q \cdot q/2$ candidates.

The scheme is equivalent to ZV attack in terms of run-time if the found polynomial is validated after Step 1. On the other hand, the scheme evaluates $O(q(q/2)(n/2))$ predictions in the worst case, via iterations of Step 3, thus it becoming equivalent to the baseline$^+$. The differentiation between both directions depends on the number of filtering traces, N_f. Note that the threshold υ presented in the previous section is not needed in the improved scheme thanks to inverse-NTT validation. The application of inverse NTT as a reliable and efficient method of verification renders the ZV-FA fault-tolerant. This method ensures the preservation of accuracy regardless of the choice of N_f, enhancing the attack performance.

VIII. Results

In this section, we present the results obtained after implementing the above-mentioned attacks in a realistic experimental setting, against Dilithium-3 and Kyber-768. Moreover, we apply our attack against a masked version of Kyber-768.



(a)

(b)

Fig. 4: Iterations of `_asymmetric_mul_16_loop`, implementation of Algorithm 3 for Dilithium, highlighted over mean trace

A. Experimental Setup

We employ Analog Devices' MAX32520[1] as the victim device to run pqm4's Dilithium signature and Kyber key decapsulation implementations. The first-order masked implementation of Kyber is developed on top of the pqm4's implementation by randomly splitting $\mathbf{s}'$. The MAX32520 incorporates a 120 MHz ARM Cortex-M4F core that can sign random 32 B messages in 65.52 ms, on average, while it can perform Kyber's decapsulation in 7.2 ms. For EM trace collection, LeCroy WavePro HD oscilloscope[2] and Langer ICR HH500-6[3] nearfield micro-probe are used. Sampling rate of the oscilloscope is set to 10 GS/s and 1 GS/s, yielding 83.33 and 8.33 samples per clock, for unprotected and protected implementations, respectively. We set up a trigger at the beginning of the implementation of Algorithm 3 from the target implementations for both Dilithium and Kyber to record the relevant time samples because we focus on the polynomial multiplication. The Scared library[4] is used for analysis and attack, running on a computer equipped with 64GB RAM and AMD Ryzen 9 5900X 12-Core Processor clocked at 3.70 GHz.

B. Pre-processing and Analysis

To cope with the adverse effects of misalignment over time, we performed the following pre-processing steps for both algorithms: 1) pattern detection, 2) signal filtering, and 3) extraction around peaks. A reference pattern is set by band-pass filtering the first trace between 100 MHz and 140 MHz and applying moving variance to it. The traces are aligned based on the reference pattern. Then, 64 peaks (128 for first-order protected Kyber), which correspond to iterations of Algorithm 3 (loop is unrolled by a factor of two), are detected and sequential points after each peak are combined. Figure 4 highlights the iterations over the average of pre-processed traces for Dilithium, conforming with the pre-knowledge on the implementation. On the other hand, iterations of Algorithm 3 for (masked) Kyber, are observed (for both shares) in Figure 5. Given the clear visibility of the iterations of Algorithm 3 over time samples, it is possible to conduct individual attacks on $\mathbf{s}'_{[i]}$ in time regions associated with each iteration. Note that, partitioning the attack range over time is critical for the presented performance results of all schemes.

Upon observation, we use the concatenation of $\mathbf{r}_{[i,0]}$ and $\mathbf{r}_{[i,1]}$ (see Algorithm 3) as the PoI for attacking Dilithium, while we use $\mathbf{r}_{[i,0]}$ for Kyber. As an initial analysis, we performed the baseline$_{\text{CPA}}$ on $\mathbf{s}'_{[0]}$ and $\mathbf{s}'_{[1]}$. The convergence patterns of the retrieved secrets are presented in Figure 6.

C. Attack and Performance

1) First-order: For the first-order attacks that target unprotected implementations of Dilithium and Kyber, we use CPA as the distinguisher. We start the evaluation by the performance of the baseline$_{\text{CPA}}$ and baseline$^{+}_{\text{CPA}}$ schemes. Figure 7 illustrates the distribution of the required number of traces $\mathbf{s}'_{[i]}$ needed to converge in our experiments. Note that, the histograms are computed based on a single $\mathbf{s}'$ and varying i. Accordingly, 220 and 150 traces are needed the retrieve whole $\mathbf{s}'$ for Dilithium and Kyber, respectively. Notice that, these numbers are the maximums of the values presented in the histograms. However, a significant portion of the secret coefficients, $\mathbf{s}'_{[i]}$, are indeed revealed with fewer traces, around 50. The performance of the baseline$_{\text{CPA}}$ and baseline$^{+}_{\text{CPA}}$ schemes is reported in Table V. In terms of accuracy, both schemes exhibit flawless performance and do not pose any concerns regardin accuracy. However, in terms of run-time, the performance of the attacks is moderate. As expected, the baseline^{+} scheme improves the performance of the baseline scheme by a factor of $\mathbf{2}\times$ while preserving accuracy, which supports the correctness of the attack methodology. Nevertheless, even with the baseline^{+} scheme, retrieving $\mathbf{s}_1$ and $\mathbf{s}_2$ requires approximately 4.37 hours for Dilithium while it takes 22.4 hours to attack $\mathbf{s}$ for the Kyber case. Note that the baseline$_{\text{CPA}}$ is equivalent to the attack presented by [21] against the same implementation of Kyber. Our application of the conventional approach is slightly better than that work both in terms of attack run-time and number of traces.

For the application of ZV-FA to Dilithium (to Kyber), the attack scenarios 1 and 2 (scenario 1 for Kyber) from Table IV are employed for attacking the lower degree coefficients of the polynomials, specifically $\mathbf{s}'_{[i,0]}$ for any $0 \leq i < 128$, while the scenarios 3 and 4 (scenario 3) are utilized for the higher degree coefficients $\mathbf{s}'_{[i,1]}$. It should be noted that scenarios 1 and 4 are not independently executed, as they represent the same attack. The outcomes of different scenarios are combined by multiplying their respective results. For both schemes, we use $N = 500$ traces for ZV-FA.Score, and d_i is doubled to increase it after each call to ZV-FA.Score (see Figure 3). Note that, we use slightly more traces compared to the convergence of the baselines (see Figure 6). This is needed to have the score

[1] https://www.analog.com/en/products/max32520.html

[2] https://teledynelecroy.com/oscilloscope/wavepro-hd-oscilloscope

[3] https://www.langer-emv.de/en/product/near-field-microprobes-icr-hh-h-field/26/icr-hh500-6-near-field-microprobe-2-mhz-to-6-ghz/108

[4] https://pypi.org/project/scared/

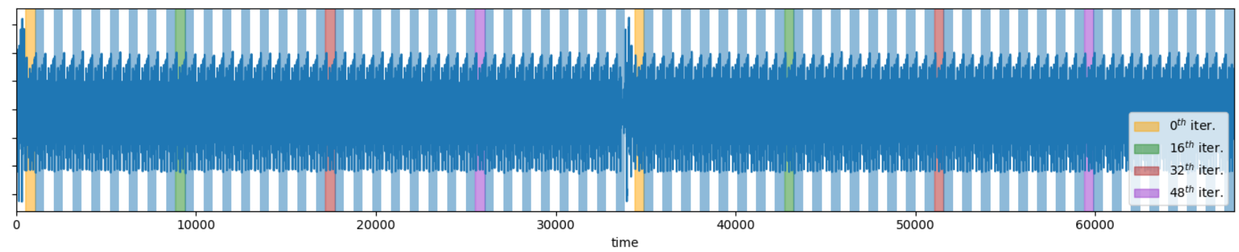


Fig. 5: Iterations of `frombytes_mul_asm_acc_32_16` for two shares, implementation of Algorithm 3 for Kyber, highlighted over mean trace.

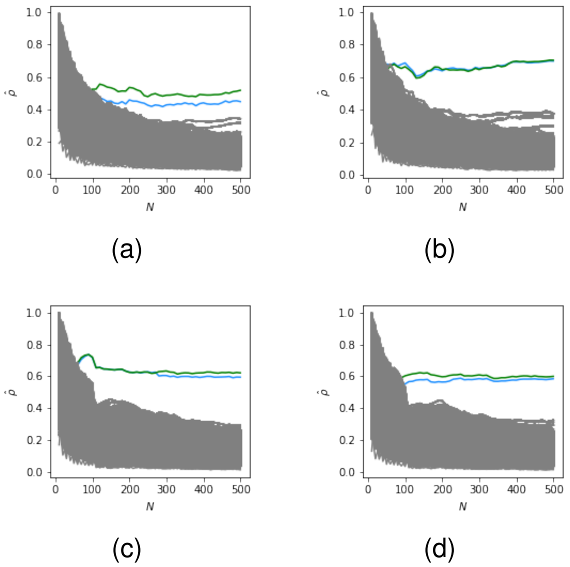


Fig. 6: Key convergence of Baseline$_{\text{CPA}}$, for $\mathbf{s}'_{[0]}$ (a,c) $\mathbf{s}'_{[1]}$ (b,d) for Dilithium (a,b) and Kyber (c,d). Green lines denote the correct hypothesis while the blue line denotes its additive inverse

of $\mathbf{s}'_{[i]}$ converged and become comparable with the scores of predictions to other secret coefficients. Observe from Figure 6 that the SNR of even and odd coefficients are different in Dilitihum. Therefore we scale the scores onto the same range by multiplying the odd coefficients by $\lambda_0/\lambda_1 \approx 5/7$.

Thanks to the inverse NTT validation and correction mechanism presented in Section VII-E, The ZVFA's accuracy is independent of the number of traces used for filtering N_f, which, however, determines its performance. Recall that, the performance of the ZV-FA scheme strongly depends on the effectiveness of the filtering stage. Figure 8 shows that, even with moderate values of N_f, the correct value for the attacked secret pair is discovered within the top-d for a significant portion of the secret coefficients and practical values of d in terms of performance. Particularly for Dilithium and $N_f = 5$K, 203 of 256 $(\%80)$ secret coefficients are retrieved in top-64, which corresponds to $(769-64)/769$ $(\%92)$ reduction in the search space from the baseline to ZV-FA.Score. Similarly for Kyber and the same value of N_f, 202 coefficients are in the top-256, leading to $(\%92)$ reduction in the search space.

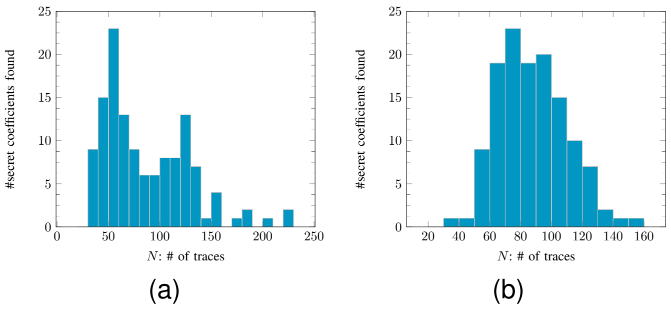
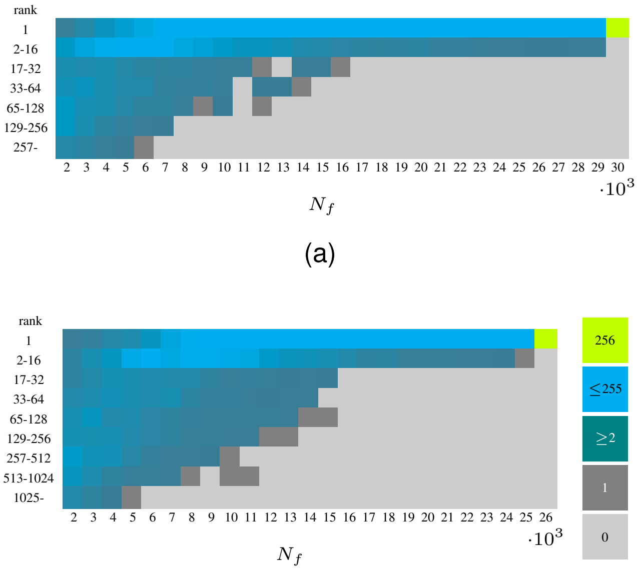


Fig. 7: Histograms of the required number of traces N for the baseline$_{\text{CPA}}$ to succeed against $\mathbf{s}'_{[i]}$ for Dilithium (a) and Kyber (b)

Experimental results indicate that the ZV filtering can substantially decrease the attack response time up to three orders of magnitude, depending on the number of filtering traces N_f available in the system. The trade-off between N_f and speed-up is illustrated in Figure 9 for both attacked algorithms. Observe that the trade-off suggests the same pattern for Dilithium and Kyber. The ZV-FA approaches to the ZV attack as N_f increases and approaches to the baseline as N_f decreases. Notably, even a small number of traces can significantly improve baseline performance. For example with

Algorithm	Method	N	Runtime($\mathbf{s}'_{[i]}$)	Runtime($\mathbf{s}'$)	
					Runtime($\mathbf{s}_1$,$\mathbf{s}_2$)
Dilithium-3	Baseline$_{\text{CPA}}$	220	22.35s	48m	≈8.74h
Dilithium-3	Baseline$^{+}_{\text{CPA}}$	220	11.18s	24m	≈4.37h
					Runtime($\mathbf{s}$)
Kyber-768	Baseline$_{\text{CPA}}$	150	7m	14.9h	≈44.8h
Kyber-768	Baseline$^{+}_{\text{CPA}}$	150	3.5m	7.45h	≈22.4h
Kyber	Baseline$_{\text{CPA}}$ [21]	200	5m	10.7h	

TABLE V: Performance of Baseline and Baseline^{+} Attacks



(a)

(b)

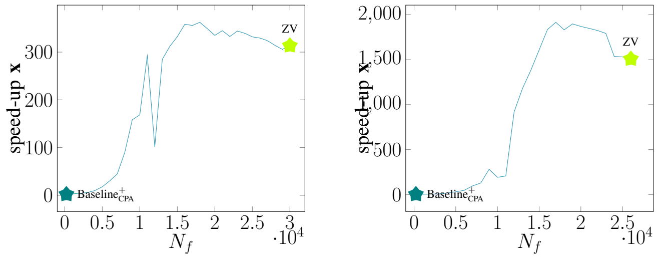
Fig. 8: Histograms of the ranks of correct hypotheses for $\mathbf{s}'_{[i,j]}$ in $\mathcal{K}^{i,j}$ during filtering stage of ZV-FA, *i.e.* the correct secret among results of ZV attacks, w.r.t. N_f for Dilithium (a) and Kyber (b)

$N_f = 5K$ the collection of which is feasible, the ZV-FA provides a speed-up of **17**× and **30**× for Dilithium and Kyber, respectively.

When more valid traces are available in the system, particularly with $N_f = 18K$, ZV-FA achieves a speed-up of **362**× for Dilithium over the baseline$_{\text{CPA}}$. We underline that, the achieved speed-up can save approximately 523 minutes ($\approx$ 8.71 hours) considering the retrieval of whole $\mathbf{s}_1$ and $\mathbf{s}_2$. Recall that $k = 6$ and $l = 5$ for Dilithium-3, which means $\mathbf{s}_1$ and $\mathbf{s}_2$ consist of 5 and 6 secret polynomials, respectively. As for Kyber (recall that $k = 3$ for Kyber-768, which means $\mathbf{s}$ has 3 elements.), our scheme is more favorable as q is roughly 2-bit larger compared to Dilithium-3. With $N_f = 17K$, ZV-FA achieves **1915**× speed-up over the baseline$_{\text{CPA}}$. It saves roughly 44.77 hours of computation time, considering all the elements of Kyber's secret vector of polynomials $\mathbf{s}$.

On the other hand, ZV-FA reduces the number of traces needed for the ZV attack while the run-time performance is slightly improved. Observe that the ZV attack is successful with $N_f = 30K$ for Dilithium which brings up a speed-up of 313×. The same speed-up is achieved by ZV-FA with $N_f \approx 14K$. Similarly for Kyber, the ZV attack is successful with $N_f = 26K$ accelerating the baseline$_{\text{CPA}}$ by 1508× while ZV-FA reaches the same speed-up with $N_f \approx 14.5K$.

Recall also that another study [22] targets Dilithium with a non-profiled attack. However, the target implementation uses the original 23-bit coefficient modulus of Dilithium. Therefore, our study is not comparable to [22] as the target implementations are different. On the other hand, while their method accelerates the baseline approach about 16 times, ours provides a speedup of more than two orders of magnitude. Consequently, we expect their method would not be as effective as ours when an incomplete NTT method is used in the implementation.

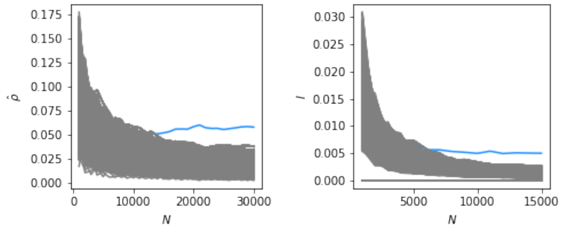


(a) (b)

Fig. 9: Speed-up of ZV-FA w.r.t. N_f for Dilithium (a) and Kyber (b). The performance of Baseline$^+_{\text{CPA}}$ and ZV attack are marked.

Method	i	N	Runtime($\mathbf{s}'_{[i]}$)	Runtime($\mathbf{s}'_{[i]}$)$\cdot n/2 \cdot k$ $\approx$Runtime($\mathbf{s}$)
Baseline$^+_{\text{HOCPA}}$	0	14K	78m	$\approx$21d
Baseline$^+_{\text{HOCPA}}$	1	23K	129m	$\approx$34d
Baseline$^+_{\text{MMIA}}$	0	7K	182m	$\approx$48.5d

TABLE VI: Performance of Baseline and Baseline$^+$ Attacks against first-order protected Kyber-768



(a) (b)

Fig. 10: Key convergence for $\mathbf{s}'_{[0]}$ Baseline$^+_{\text{HOCPA}}$ (a) Baseline$^+_{\text{MMIA}}$ (b) for masked Kyber. Blue line denotes the additive inverse of the correct hypothesis

2) Second-order: To discuss the efficiency of our attack in the protected case, we first present the performance of the baseline schemes thereof in Table VI. Due to long running times, we perform the evaluation only for $\mathbf{s}'_{[0]}$ and $\mathbf{s}'_{[1]}$. The last column approximates the required amount of time to break whole secret key $\mathbf{s}$, based on the statistics of the attacks to $\mathbf{s}'_{[0]}$ and $\mathbf{s}'_{[1]}$. For instance, retrieving $\mathbf{s}$ would roughly take 68 days if all coefficients were retrieved by 23K traces as $\mathbf{s}'_{[1]}$. Therefore the baseline scheme stands as an impractical option. The speed-up of ZV-FA is computed based on the average number of traces needed by Baseline$_{\text{HOCPA}}$ over the analyzed coefficients, $(23 + 14)/2 = 18.5$K.

Observe from Table VI and Figure 10 that MMIA is superior to HOCPA in terms of the number of traces while HOCPA runs faster. Therefore, we use MMIA as the distinguisher in the filtering stage, differently from the unprotected case. We

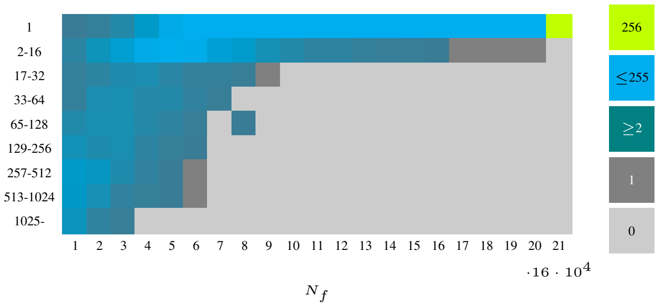


Fig. 11: Histograms of the ranks of the correct hypotheses for $\mathbf{s}'_{[i,j]}$ in $\mathcal{K}^{i,j}$ during filtering stage of ZV-FA, w.r.t. N_f, against masked Kyber

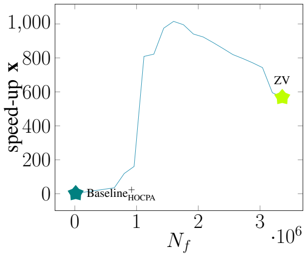


Fig. 12: Speed-up of ZV-FA w.r.t. N_f against Masked Kyber.

utilize MMIA with two bins and select the bins such that it partitions the samples into equal parts, *i.e.* halves. As the iterations of Algorithm 3 are easily distinguishable through the mean trace as depicted by Figure 5, we use constant offset for combining points over time samples. We would like to note that we observe that the MMIA is quite efficient in this setting. However, we leave a comprehensive study on MMIA in comparison with HOCPA regarding arithmetic masking as a future work. On the other hand, we use HOCPA in the scoring stage, considering its superior run-time performance and we employ $N = 50K$ traces for scoring. Recall that, significantly more traces are available during scoring as zero-value conditions are not sought therein.

Figure 11 demonstrates the distribution of the ranking of the correct hypothesis for $\mathbf{s}'_{[i,j]}$, among the sorted results from the filtering stage of ZV-FA, $\mathcal{K}^{i,j}$. We observe that the filtering stage is quite effective as in the unprotected case. Particularly with $N_f = 480K$, for 213 out of 256 secret coefficients, the correct hypothesis is in top-256, *i.e.* $\mathcal{K}^{i,j}_{[:256]}$, leading to %92 reduction in search space. Figure 12 depicts the speed-up values achieved over the Baseline$_{\text{HOCPA}}$. We observe that ZV-FA reaches **1015**$\times$ speed-up over Baseline$_{\text{HOCPA}}$, with $N_f = 1.6M$. If those many traces are not available to the attacker, ZV-FA is still more practical compared to the Baseline$_{\text{HOCPA}}$. For instance, a speed-up of 35× is observed with a relatively less number of filtering traces, $N_f = 640K$.

IX. Conclusion and Future Work

This paper presents a series of non-profiled side-channel attacks against the incomplete NTT-based polynomial multiplication (Algorithm 3), which is widely adopted in lattice-based cryptography. We exercised our approach in the proposed attacks against the signature algorithm Dilithium and the KEM Kyber [16], [17]. Specifically, the attacks focus on the NTT-based polynomial multiplications $c\mathbf{s}_1$ and $\mathbf{s}^T\mathbf{u}$. The target implementation for Dilithium operates with a carrier prime $q' = 769$, which restricts the NTT to be incomplete. On the other hand, Kyber also needs to employ 7 layers of incomplete NTT by algorithm definition, albeit with a different prime $q = 3329$. The baseline and baseline$^+$ schemes are conventional methods that rely on brute-force methods in the sets of cardinality q^2 and $q^2/2$, respectively, as two coefficients of the incomplete NTT representation must be predicted together.

To mitigate the search costs, we introduced the zero-value attack, which reduces the size of the set of hypotheses to q in the brute force attack by taking advantage of multiplication by 0 to eliminate one of the attacked pair of coefficients from the equation. However, this approach requires a significantly higher number of traces. Next, we presented the zero-value filtering attack, which represents a trade-off between the number of traces and attack run-time. With an appropriate number of traces, this attack can achieve a speed-up of two orders of magnitude over the baseline. Finally, we proposed an efficient way of verification of predictions on short polynomials, utilizing the inverse NTT transformation. It makes the proposed scheme accurate independent of the number of filtering traces. Experiments suggest that the ZV-FA is favorable even with moderate parameters. Moreover, we show that our attack is effective in the presence of masking by applying it against a first-order protected implementation of Kyber. Our approach is also generalizable to higher orders as long as the attacker can combine leaky samples over time that correspond to the different shares. During the study, we found out that MMIA outperforms HOCPA in terms of the number of traces and we leave this study as a future work.

Acknowledgments

This work is partially supported by the European Union's Horizon Europe research and innovation program under grant agreement No: 101079319 and by TUBITAK under Grant Number 122E222.

References

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

[2] V. S. Miller, *Use of elliptic curves in cryptography*. Springer, 1986.

[3] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology—CRYPTO'89 Proceedings 9*. Springer, 1990, pp. 239–252.

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[5] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[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 (version 3.1)," *NIST Post-Quantum Cryptography Standardization Round*, vol. 3, 2021.

[7] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.

[8] J. 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)*. IEEE, 2018, pp. 353–367.

[9] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2004, pp. 16–29.

[10] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis: A generic side-channel distinguisher," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2008, pp. 426–442.

[11] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side—channel (s)," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2002, pp. 29–45.

[12] J. I. Librabry, "Application of attack potential to smartcards (version 3.1)," 2020.

[13] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *J. ACM*, vol. 60, no. 6, pp. 43:1–43:35, 2013. [Online]. Available: https://doi.org/10.1145/2535925

[14] R. Agarwal and C. Burrus, "Fast convolution using fermat number transforms with applications to digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 2, pp. 87–97, 1974.

[15] V. Lyubashevsky and G. Seiler, "NTTRU: truly fast NTRU using NTT," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 3, pp. 180–201, 2019. [Online]. Available: https://doi.org/10.13154/tches.v2019.i3.180-201

[16] A. Abdulrahman, V. Hwang, M. J. Kannwischer, and A. Sprenkels, "Faster kyber and dilithium on the cortex-m4," in *Applied Cryptography and Network Security - 20th International Conference, ACNS 2022, Rome, Italy, June 20-23, 2022, Proceedings*, ser. Lecture Notes in Computer Science, G. Ateniese and D. Venturi, Eds., vol. 13269. Springer, 2022, pp. 853–871. [Online]. Available: https://doi.org/10.1007/978-3-031-09234-3_42

[17] 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.

[18] I.-J. Kim, T.-H. Lee, J. Han, B.-Y. Sim, and D.-G. Han, "Novel single-trace ml profiling attacks on nist 3 round candidate dilithium," *Cryptology ePrint Archive*, 2020.

[19] S. Marzougui, V. Ulitzsch, M. Tibouchi, and J.-P. Seifert, "Profiling side-channel attacks on dilithium: A small bit-fiddling leak breaks it all," *Cryptology ePrint Archive*, 2022.

[20] R. Primas, P. Pessl, and S. Mangard, "Single-trace side-channel attacks on masked lattice-based encryption," in *Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*. Springer, 2017, pp. 513–533.

[21] 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 Transactions on Embedded Computing Systems*, 2022.

[22] 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 *2021 IEEE 39th International Conference on Computer Design (ICCD)*. IEEE, 2021, pp. 583–590.

[23] H. Steffen, G. Land, L. Kogelheide, and T. Güneysu, "Breaking and protecting the crystal: Side-channel analysis of dilithium in hardware," *Cryptology ePrint Archive*, 2022.

[24] 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 *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2021, pp. 35–45.

[25] A. Berzati, A. C. Viera, M. Chartouni, S. Madec, D. Vergnaud, and D. Vigilant, "A practical template attack on crystals-dilithium," *Cryptology ePrint Archive*, 2023.

[26] Z. Xu, O. Pemberton, S. S. Roy, D. Oswald, W. Yao, and Z. Zheng, "Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber," *IEEE Transactions on Computers*, vol. 71, no. 9, pp. 2163–2176, 2021.

[27] O. Reparaz, S. S. Roy, R. De Clercq, F. Vercauteren, and I. Verbauwhede, "Masking ring-lwe," *Journal of Cryptographic Engineering*, vol. 6, no. 2, pp. 139–153, 2016.

[28] E. Prouff, M. Rivain, and R. Bevan, "Statistical analysis of second order differential power analysis," *IEEE Transactions on computers*, vol. 58, no. 6, pp. 799–811, 2009.

[29] B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede, "Revisiting higher-order dpa attacks: Multivariate mutual information analysis," in *Topics in Cryptology-CT-RSA 2010: The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*. Springer, 2010, pp. 221–234.

[30] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[31] W. M. Gentleman and G. Sande, "Fast fourier transforms: for fun and profit," in *Proceedings of the November 7-10, 1966, fall joint computer conference*, 1966, pp. 563–578.

[32] V. Lyubashevsky and G. Seiler, "Nttru: truly fast ntru using ntt," *Cryptology ePrint Archive*, 2019.

[33] G. Seiler, "Faster avx2 optimized ntt multiplication for ring-lwe lattice cryptography," *Cryptology ePrint Archive*, 2018.

[34] V. Migliore, B. Gérard, M. Tibouchi, and P.-A. 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 17*. Springer, 2019, pp. 344–362.

[35] M. Azouaoui, O. Bronchain, G. Cassiers, C. Hoffmann, Y. Kuzovkova, J. Renes, T. Schneider, M. Schönauer, F.-X. Standaert, and C. van Vredendaal, "Protecting dilithium against leakage: Revisited sensitivity analysis and improved implementations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2023, no. 4, pp. 58–79, 2023.

[36] O. Bronchain, M. Azouaoui, M. ElGhamrawy, J. Renes, and T. Schneider, "Exploiting small-norm polynomial multiplication with physical attacks: Application to crystals-dilithium," *Cryptology ePrint Archive*, 2023.

X. Biography Section

Tolun Tosun received the B.S. and M.S. degrees in computer science and engineering from the Faculty of Natural Science and Engineering, Sabanci University in 2016 and 2019, respectively. He is a Ph.D. student in Sabanci University starting from 2019. His research interests include applied cryptography; side-channel attacks and countermeasures, lattice-based cryptography, applications of (fully) homomorphic encryption, privacy preserving machine learning. He is also working in the industry since 2019 as a cryptography engineer.

Erkay Savas received the BS (1990) and MS (1994) degrees in electrical engineering from the Electronics and Communications Engineering Department at Istanbul Technical University. He completed the PhD degree in the Department of Electrical and Computer Engineering (ECE) at Oregon State University in June 2000. He had worked for various companies and research institutions before he joined Sabanci University in 2002. He has been the dean of Faculty of Engineering and Natural Science, Sabanci University, since July 1, 2020. His research interests include applied cryptography, data and communication security, privacy in biometrics, security and privacy in data mining applications, embedded systems security, and distributed systems. He is a member of IEEE, ACM, the IEEE Computer Society, and the International Association of Cryptologic Research (IACR).