When NTT Meets SIS: Efficient Side-channel Attacks on Dilithium and Kyber

Zehua Qiao, Yuejun Liu, Yongbin Zhou, Mingyao Shao, and Shuo Sun

***Abstract*—In 2022, NIST selected Kyber and Dilithium as post-quantum cryptographic standard algorithms. The Number Theoretic Transformation (NTT) algorithm, which facilitates polynomial multiplication, has become a primary target for side-channel attacks. In this work, we embed the NTT transformation matrix in Dilithium and Kyber into the SIS search problem, and further, we propose a divide and conquer strategy for dimensionality reduction of the SIS problem by utilizing the properties of NTT, and discuss the effectiveness of the BKZ algorithm for solving the problem by using the LLL and with different blocksize, respectively. When using BKZ-60, the time required to recover private keys s_1 for Dilithium2 after using the dimensionality reduction strategy is reduced from 82 hours to 1 minute, which is a 4,900$\times$ improvement, and the minimum number of coefficients required is reduced from 65 to 32, which is close to the theoretical lower limit value of 28. Furthermore, we propose a parameter-adjustable CPA scheme to expedite the recovery of a single coefficient in NTT domain. Combining this CPA scheme with the SIS-assisted approach, we executed practical attacks on both unprotected and masked implementations of Dilithium and Kyber on an ARM Cortex-M4. The results demonstrate that, using 5,000 power traces, we can recover complete s_1 of Dilithium2 in 2.4 minutes, which achieve a 400$\times$ speedup compared to the best-known attacks. And Kyber512 takes only 0.5 minutes, a 7.5$\times$ improvement over what's already working. Moreover, we successfully break the first-order masked implementations and explore the potential applicable to higher-order implementations.**

***Index Terms*—Lattice-based Cryptography, Number Theoretic Transformation, Side-channel Attacks, Short Integer Solution, Dilithium, Kyber.**

I. Introduction

TRADITIONAL public key cryptography relies on the computational intractability of problems such as integer factorization and discrete logarithms. However, the emergence of quantum computing has raised concerns, as it promises polynomial-time solutions to these problems, thereby compromising the security of our existing cryptographic algorithms [1]. Recognizing this threat, the National Institute of Standards and Technology (NIST) initiated a Post-Quantum Cryptography (PQC) standardization process in 2016, aiming to identify quantum-resistant cryptographic algorithms. By July 2022, NIST published its first set of post-quantum cryptographic standard algorithms [2], comprising three signature algorithms: Dilithium, FALCON, and SPHINCS+; and one key encapsulation mechanism (KEM) algorithm, Kyber. In August 2023, NIST released three draft standards except for FALCON. This paper mainly focuses on Dilithium and Kyber.

While the fundamental security of these algorithms has been widely recognized in the cryptographic community, Side-channel Attacks (SCAs) emerged as formidable adversaries, underscoring the imperative of securing cryptographic implementations against these non-traditional threats [3]. This was not a nuance lost on NIST, which duly emphasized side-channel security during its PQC evaluations [2].

Preliminary SCAs targeting Dilithium and Kyber have emerged. For the Kyber, a significant type of key recovery attack combines Chosen Ciphertext Attacks (CCA) with SCAs [4]–[7], in which adversaries constructed the Plaintext-Checking oracle with the help of SCAs to determine whether the message is successfully recovered, then extracted the information of private key. Recently, Shen *et al.* [8] developed a method that adapts to imperfect PC oracles constructed via SCAs and is still capable of recovering private keys. For the Dilithium, a typical class of attack is the randomness leakage attack, initially addressed by Liu *et al.* [9], [10]. They demonstrated that even a single-bit leakage of random polynomial per signature can be devastating for lattice-based Fiat-Shamir signatures. Later, SCAs based on similar mathematical tools were proposed by Marzougui *et al.* [11] and Berzati *et al.* [12]. These attacks require an in-depth understanding of Kyber and Dilithium. However, simpler methods are typically desirable, such as classical SCAs targeting operations in which the private key is directly involved.

Both Dilithium and Kyber operate over the cyclotomic ring $\mathbb{Z}_q[x]/(x^n+1)$, leveraging the Number Theoretic Transformation (NTT) to accelerate the polynomial multiplication. As a fundamental module of operation, the private key will almost certainly perform NTT polynomial multiplication, thereby making it vulnerable to SCAs. These algorithms primarily employ two categories of operations: NTT and Inverse NTT (INTT) operations, large number multiplication and reduction (Montgomery or Barrett) operations. For the former, existing work mainly exploited profiled side-channel methods. In 2017, Primas *et al.* [13] pioneered a single-trace attack targeting the NTT operation using the Template Attacks (TAs) and belief propagation algorithms. Building on this foundational work, Hamburg *et al.* [14] successfully recovered the private key of masked Kyber. In a related vein, Xu *et al.* [5] introduced a Simple Power Analysis (SPA) method targeting the INTT operation during Kyber's decapsulation phase. Correspondingly, Han *et al.* [15] executed the first practical attack using machine learning-based TA aimed at the NTT operation within Dilithium's signing procedure. While potent, such attacks typically usually require additional control privileges over the target system. In addition, Tosun *et al.* [16] proposed zero-value filtering to accelerate the SCA on incomplete NTT-based implementations of Dilithium and Kyber.

In contrast to the aforementioned operations, research on

attacks against multiplication and reduction operations has predominantly utilized Correlation Power Analysis (CPA) methods. Regarding Kyber, Karlov *et al.* [17] conducted practical attacks on Kyber's pqm4 open-source implementation [18]. Enhancing this approach, Yang *et al.* [19] optimized the attack by filtering ciphertexts, thereby diminishing the enumeration space and enhancing the Signal-to-Noise (SNR). In the context of Dilithium, Fournaris *et al.* [20] demonstrated the potential of CPA on its NTT polynomial multiplication procedure. Further refining this method, Chen *et al.* [21] optimized this method by employing the divide and conquer strategy, successfully reducing the time required to recover a single NTT domain coefficient from 6,357 to 818 seconds. While these attacks are conceptually simpler, they entail a substantial computational overhead.

The private keys for Kyber and Dilithium are represented as high-dimensional polynomials with independent coefficients. The ultimate objective of attacks is to deduce the entire private key, which entails significant computational costs. An emerging strategy is to recover partial coefficients via SCAs instead and recover the remaining ones using other methods, such as enumeration and lattice reduction. This strategy has paid off, for example, the combination of lattice reduction and misuse attacks reduces the cost of attacking Kyber512 by 34% [22]. Krahmer *et al.* [23] and Albrecht *et al.* [24] focused on and used the BKZ algorithm to recover the full private key of Dilithium and Kyber under a known partial NTT domain private key. However the coefficients required to complete the attack for the above work are far from the theoretical lower bound, and the time required is large for Dilithium. For example, Krahmer *et al.* [23] demonstrate that when 128 coefficients are known for Dilithium2, executing BKZ-30 takes about 90 minutes. This number of coefficients is significantly higher than the theoretical lower bound of 26 coefficients for Dilithium2. Most importantly, executing the above algorithm requires verification that the recovered coefficients are correct.

Both the Dilithium and Kyber algorithms are constructed based on the MSIS problem to ensure their security. However, through analysis of the private key NTT process, we have found that this process can be converted into an SIS search problem for resolution, and the difficulty can be further reduced through a dimension reduction strategy. Additionally, we have proposed a rapid CPA scheme that enhances attack efficiency and allows us to determine the success of SCAs in open-source, unprotected implementations. Lastly, we conducted practical attacks on both unprotected and masked protective implementations.

Our contributions are as follows:

- We demonstrated how to transform the challenge of recovering full coefficients from known partial private key NTT domain coefficients in Dilithium and Kyber into an SIS search problem. Furthermore, by leveraging the characteristics of NTT, we proposed a dimension reduction strategy that significantly enhanced the solving efficiency of the LLL and BKZ algorithms, bringing the number of required NTT domain coefficients close to the theoretical lower bound. Through theoretical analysis and practical experiments, we proved that the dimension reduction strategy enables faster solving speeds and requires fewer coefficients for solving the SIS problems extracted from Dilithium and Kyber.
- Drawing inspiration from Tunstall *et al.* [25], we have developed a parameter-adjustable CPA-based SCA scheme that targets the numerous multiplications fundamental to Dilithium. Utilizing this scheme, a coefficient of Dilithium's NTT domain private key can be recovered in under a second—achieving a speed that is 40 times faster than the methods presented by Chen *et al.* [21].
- We combined SCA with the dimensionality reduction strategy for solving SIS problems and successfully performed practical attacks on unprotected Dilithium and Kyber on the ARM Cortex-M4 platform. Using 5000 power traces, the time required to recover the complete private key $\mathbf{s}_1$ for Dilithium2, 3, and 5 was only 2.4, 3.0, and 4.1 minutes, respectively, which is more than 260 times faster than previous works. For Kyber512, 768, and 1024, the time required to recover the complete private key $\mathbf{s}_k$ was 0.5, 0.8, and 1.1 minutes, respectively, achieving a speed improvement of approximately 7 times.
- We conducted practical attacks on first-order masked Dilithium with key refreshment and first-order masked Kyber without key refreshment to demonstrate the application of our scheme in different scenarios. Using 5,000 traces, for masked Dilithium2, 3, and 5, combining higher-order CPA with SIS problem dimension reduction strategies, the time taken to recover the complete $\mathbf{s}_1$ was 89.9, 112.3, and 157.3 hours, respectively. To our knowledge, this is also the first instance of a non-template-based higher-order SCA on masked NTT domain multiplication operations. For masked Kyber512, 768, and 1024, since the private keys were not refreshed, it is possible to solve for each mask separately and combine them to recover the private keys. Using 500 traces, the time required to recover the complete private key $\mathbf{s}_k$ was 3.7, 5.7, and 7.6 minutes, respectively.

II. Preliminaries

A. Dilithium

Dilithium, a digital signature scheme based on the Module Learning with Errors (MLWE) and Module Short Integer Solution (MSIS) problems, offers different security levels through its adjustable specific parameters, thus accommodating a variety of application scenarios and device constraints. Information about these parameters is presented in Tab.I.

TABLE I: Parameters of Dilithium

NIST Security Level	2	3	4
d [dropped bits from $\mathbf{t}$]	13		
α [# of non-zero coefficients in $\mathbf{c}$]	39	49	60
γ_1 [cofficient range of $\mathbf{y}$]	131072	524288	
γ_2 [low-order rounding range]	95232	261888	
$(k \times l)$ [dimensions of $\mathbf{A}$]	(4,4)	(6,5)	(8,7)
η [private key range]	2	4	2

The computations involved in the Dilithium algorithm are carried out within the cyclotomic ring $\mathbb{R}_q^n$, wherein all co-

efficients are elements of the finite field $\mathbb{Z}_q$. The values of $q = 8380417$ and $n = 256$ are invariant at all security levels. Dilithium comprises three procedures: key generation, signing, and verification. Our study specifically focuses on the signing process, as shown in Alg.1.

Algorithm 1 Dilithium Sign(*sk,M*)

Input: $sk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0), M$
Output: $signature$
1: $\mathbf{A} \in R_q^{m \times n} :=$ ExpandA(ρ)
2: $\mu \in \{0,1\}^{384} :=$ CRH$(tr||M)$
3: $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$
4: $\rho' \in \{0,1\}^{384} :=$ CRH$(K||\mu)$ (or $\rho' \leftarrow \{0,1\}^{384}$)
5: $\hat{\mathbf{A}} =$ NTT$(\mathbf{A}), \hat{\mathbf{s}}_1 =$ NTT$(\mathbf{s}_1)$
6: $\mathbf{y} \in S^n_{\gamma_1 - 1} :=$ ExpandMask(ρ', κ)
7: $\mathbf{w} :=$ NTT$^{-1}(\hat{\mathbf{A}} \circ$ NTT$(\mathbf{y}))$
8: $\mathbf{w}_1 :=$ HighBits$_q(w, 2\gamma_2)$
9: $\tilde{\mathbf{c}} \in \{0,1\}^{256} := \mathbf{H}(\mu||\mathbf{w}_1)$
10: $\hat{\mathbf{c}} :=$ NTT(SampleInBall$(\tilde{\mathbf{c}}))$
11: $\mathbf{z} := \mathbf{y} +$ NTT$^{-1}(\hat{\mathbf{c}} \circ \hat{\mathbf{s}}_1)$
12: $\mathbf{r}_0 :=$ LowBits$_q(\mathbf{w} - \mathbf{cs}_2, 2\gamma_2)$
13: **if** $||\mathbf{z}||_\infty \geqslant \gamma_1 - \beta$ **or** $||\mathbf{r}_0||_\infty \geqslant \gamma_2 - \beta$ then $\kappa := \kappa + l$, goto 6
14: **else**
15: $\quad \mathbf{h} :=$ MakeHint$_q(-\mathbf{ct}_0, \mathbf{w} - \mathbf{cs}_2 + \mathbf{ct}_0, 2\gamma_2)$
16: $\quad$ **if** $||\mathbf{ct}_0||_\infty \geqslant \gamma_2$ **or** the # of 1$'$s in $\mathbf{h}$ is greater than ω then $\kappa := \kappa + l$, goto 6
17: **return** $signature = (\mathbf{z}, \mathbf{h}, \tilde{\mathbf{c}})$

During the signing phase, the algorithm takes the private key and a message as inputs. Following this, the Expand function generates a matrix, denoted as $\mathbf{A}$, and a masking vector of polynomials, $\mathbf{y}$, whose coefficients are less than γ_1. To expedite the computations, NTT operations are applied to the matrix $\mathbf{A}$ and the private keys $\mathbf{s}_1$. The signer then computes $\mathbf{Ay}$ and designates the higher-order bits of the coefficients of this vector to $\mathbf{w}_1$. Using the message M and $\mathbf{w}_1$, the challenge $\mathbf{c}$ is constructed, which subsequently aids in the generation of the signature $\mathbf{z}$. The algorithm also incorporates a rejection sampling loop that checks if the generated challenge $\mathbf{c}$ and signature $\mathbf{z}$ meet the prescribed output conditions. If these conditions, as detailed in lines 13-16, are satisfied, the algorithm outputs the result and the signature process is completed. If not, the algorithm revisits line 6 to regenerate the signature until a valid one is obtained.

B. Kyber

Kyber is a KEM that achieves IND-CCA security. Its security depends on the complexity of the MLWE problem. Kyber offers three distinct security levels: Kyber512, Kyber768, and Kyber1024, each affording different tiers of cryptographic strength. The parameter choices for each of these security levels are outlined in Tab.II. The values of $q = 3329$ and $n = 256$ remain unchanged at all security levels.

Kyber offers a construction known as Kyber.CCAKEM, which is derived from Kyber.CPAPKE (this CPA stands for Chosen Plaintext Attack) through a variation of the FO (Fujisaki–Okamoto) transform. The Kyber.CCAKEM includes three key phases: key generation, encapsulation, and decapsulation. Both secret keys and error vectors are sampled from a centered binomial distribution $\mathbf{B}_\eta$, expressed as $\sum_{i=1}^{\eta}(a_i - b_i)$, where each a_i and b_i are independently and randomly sampled from the set $\{0,1\}$. This paper primarily explores the decapsulation process, illustrated in Alg.2.

TABLE II: Parameters of Kyber

NIST Security Level	1	3	5
k [dimension of polynomial ring]	2	3	4
η_1 [noise of $\mathbf{s}$,$\mathbf{e}$ in KeyGen()]	3	2	2
η_2 [noise of $\mathbf{e}_1$ and $\mathbf{e}_2$ in Enc()]	2		
(d_u, d_v) [compression parameters]	(10,4)	(10,4)	(11,5)

Algorithm 2 Kyber.CCAKEM.Dec($\mathbf{c}$,sk)

Input: $\mathbf{c} = (c_1, c_2), sk$
Output: $shared\ key\ K$
1: $pk := sk + 12 \cdot k \cdot n/8$
2: $h := sk + 24 \cdot k \cdot n/8 + 32$
3: $z := sk + 24 \cdot k \cdot n/8 + 64$
4: $\mathbf{u} :=$ Decompress$_q$(Decode$_{d_u}(c_1), d_u)$
5: $v :=$ Decompress$_q$(Decode$_{d_v}(c_2), d_v)$
6: $\hat{\mathbf{s}}_k :=$ Decode$_{12}(sk)$
7: $m' :=$ Encode$_1$(Compress$_q(v -$ NTT$^{-1}(\hat{\mathbf{s}}_k^T \circ$NTT$(\mathbf{u})), 1))$
8: $(\bar{K}', r') :=$ G$(m'||h)$
9: $c' :=$ Kyber.CPAPKE.Enc(pk, m', r')
10: **if** $\mathbf{c} = \mathbf{c}'$ **then**
$\quad$ **return** $K :=$ KDF$(\bar{K}'||$H$(\mathbf{c}))$
11: **else**
$\quad$ **return** $K :=$ KDF$(z||$H$(\mathbf{c}))$
12: **end if**
13: **return** K

This algorithm takes the ciphertext and private key as input for decapsulation. The message is computed following decompression, and re-encryption is employed to verify the ciphertext's validity and deliver the final result. The algorithm's implementation also necessitates a significant number of polynomial multiplications over a finite field, with the NTT utilized to enhance the efficiency of the implementation.

C. Number Theoretic Transformations

Number Theoretic Transform (NTT), as the finite field counterpart to the Fast Fourier Transform, is instrumental in optimizing polynomial multiplication. To elaborate, consider polynomials $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ and $\mathbf{y} = (y_0, y_1, \ldots, y_{n-1})$. The multiplication process of these polynomials can be broadly divided into three stages:

1) Transforming $\mathbf{x}$ and $\mathbf{y}$ into NTT domain, *i.e.*, $\hat{\mathbf{x}} =$ NTT$(\mathbf{x})$ and $\hat{\mathbf{y}} =$ NTT$(\mathbf{y})$.
2) Perform point-wise multiplication, *i.e.*, $\hat{\mathbf{z}} = \hat{\mathbf{x}} \circ \hat{\mathbf{y}}$.
3) Apply INTT to normal domain, *i.e.*, $\mathbf{z} =$ INTT$(\hat{\mathbf{z}})$.

Applying NTT reduces the computational complexity of polynomial multiplication from $O(n^2)$ to $O(n \log n)$ when compared to the schoolbook method. Different polynomial

rings, such as $\mathbb{Z}_q[x]/(x^n-1)$ and $\mathbb{Z}_q[x]/(x^n+1)$, necessitate positive and negative convolutions respectively to accomplish NTT operations [26]. For the Dilithium and Kyber, a negative convolution is employed. The NIST reference implementation of Dilithium applies $2n$-th primitive root of unity in $\mathbb{Z}_q$, $\phi = 1753$ and $\omega = \phi^2 = 3073009$ for all security levels. The detailed calculation process is provided below and all calculations are done under $\mod q$.

$$\Phi = \begin{pmatrix} \phi^0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \phi^{n-1} \end{pmatrix} \quad \Omega = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)^2} \end{pmatrix}$$

$$\hat{\mathbf{x}} = \text{NTT}(\mathbf{x}) = \Omega\Phi\mathbf{x} \qquad \hat{\mathbf{y}} = \text{NTT}(\mathbf{y}) = \Omega\Phi\mathbf{y}$$

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \circ \hat{\mathbf{y}} \qquad \mathbf{z} = \text{INTT}(\hat{\mathbf{z}}) \tag{1}$$

Kyber also employs the NTT algorithm, however, its finite field $q = 3329$ contains merely a 256-th primitive root, denoted as ζ. Consequently, this necessitates distinct computational nuances in its implementation. The polynomial can be represented as

$$X^{256}+1 = \prod_{i=0}^{127}(X^2-\zeta^{2i+1}) = \prod_{i=0}^{127}(X^2-\zeta^{2br_7(i)+1}) \tag{2}$$

where $br_7(i)$ represents the bit-reversed order of the 7-bit unsigned integer i. The transformation process is detailed in [27]. In practice, Kyber performs seven butterfly operations on 256 coefficients in the polynomial, resulting in the final NTT domain polynomial as follows:

$$\hat{\mathbf{x}}_{2i} = \sum_{j=0}^{127} \mathbf{x}_{2j}\zeta^{(2br_7(i)+1)j}$$

$$\hat{\mathbf{x}}_{2i+1} = \sum_{j=0}^{127} \mathbf{x}_{2j+1}\zeta^{(2br_7(i)+1)j} \tag{3}$$

$$\mathbf{Z} = \begin{pmatrix} (\zeta^1)^0 & 0 & \cdots & (\zeta^1)^{127} & 0 \\ 0 & (\zeta^1)^0 & \cdots & 0 & (\zeta^1)^{127} \\ (\zeta^{129})^0 & 0 & \cdots & (\zeta^{129})^{127} & 0 \\ 0 & (\zeta^{129})^0 & \cdots & 0 & (\zeta^{129})^{127} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (\zeta^{255})^0 & 0 & \cdots & (\zeta^{255})^{127} & 0 \\ 0 & (\zeta^{255})^0 & \cdots & 0 & (\zeta^{255})^{127} \end{pmatrix}$$

$$\text{NTT}(\mathbf{x}) = \mathbf{Z}\mathbf{x} \tag{4}$$

It is also possible to split the Equation (4) into odd and even positions according to the coefficient subscript number and calculate the value of the NTT domain separately. Let $\mathbf{x}^{(e)} = (x_0, x_2, \cdots, x_{n-2})$ and $\mathbf{x}^{(o)} = (x_1, x_3, \cdots, x_{n-1})$, which can be obtained:

$$\mathbf{Z}_{\setminus\{0\}} = \begin{pmatrix} (\zeta^1)^0 & (\zeta^1)^1 & \cdots & (\zeta^1)^{127} \\ (\zeta^{129})^0 & (\zeta^{129})^1 & \cdots & (\zeta^{129})^{127} \\ \vdots & \vdots & \ddots & \vdots \\ (\zeta^{255})^0 & (\zeta^{255})^1 & \cdots & (\zeta^{255})^{127} \end{pmatrix}$$

$$\text{NTT}(\mathbf{x}^{(e)}) = \mathbf{Z}_{\setminus\{0\}}\mathbf{x}^{(e)} \qquad \text{NTT}(\mathbf{x}^{(o)}) = \mathbf{Z}_{\setminus\{0\}}\mathbf{x}^{(o)} \tag{5}$$

In the NIST reference implementation, Kyber maintains $\zeta = 17$ at all security levels. In summary, NTT offers an efficient framework for polynomial multiplication, a process integral to the performance of lattice-based cryptographic algorithms like Dilithium and Kyber. While both algorithms utilize NTT, it is important to acknowledge that their implementation details and parameters differ, with each being meticulously tailored to accommodate their unique algebraic structures.

D. Correlation Power Analysis

Correlation Power Analysis (CPA), introduced by Brier *et al.* [28], stands as a potent and refined SCA method. Since its inception, it has been adeptly leveraged to break the security of cryptographic algorithms, including DES [29] and AES [30]. The canonical procedure of CPA entails an adversary determining the Pearson Correlation Coefficient (PCC) between the observed side-channel leakage and the estimated intermediate values. Let $f(x_1, \ldots, x_p, k^*)$ represent a deterministic function indicative of the intermediate value of interest, where x_i symbolize known fluctuating parameters (such as plaintext or ciphertext), and k^* is the concealed secret key. Accumulating n traces, denoted by L, the adversary elects a pertinent leakage model M (e.g., Identity or Hamming Weight) to ascertain the hypothetical intermediate values $H_{k_j} = M(f(x_1, \ldots, x_p, k_j))$ for every conceivable key k_j within the set $\mathbb{K}$. The PCC, symbolized as $\rho(L, H_{k_j})$, is subsequently computed for each key hypothesis. The formula for PCC is given by:

$$\rho(L, H_{k_j}) = \frac{\sum_{i=1}^{n}(l_i-\bar{l})(h_i-\bar{h})}{\sqrt{\sum_{i=1}^{n}(l_i-\bar{l})^2}\sqrt{\sum_{i=1}^{n}(h_i-\bar{h})^2}} \tag{6}$$

The candidate key k_{cpa} that provides the highest correlation is chosen as the recovered key. The attack is considered successful if k_{cpa} matches the secret key k^*.

III. Incomplete NTT domain recovery methods

For Dilithium, even recover 99% of the 256 NTT domain coefficients, the search space required to recover the private key is still approximately $8380417^{2.56} \approx 2^{59}$. To address this challenge, we introduce two methods: one approach is transform the problem into overdetermined systems of equations, while the other is construct it as a SIS search problem.

A. Overdetermined system-based method

The NTT domain of Dilithium and Kyber's private key extends the range of each coefficient from $2\eta+1$ to q, with a static transformation matrix that makes the process a injective transformation, and maintains the private key space

at $(2\eta+1)^{256}$. Intuitively, when there are m coefficients of NTT domain private key unknown, the search space should be $(2\eta+1)^m$ instead of q^m. This goal is achievable by establishing and resolving overdetermined equations. Consider a vector $\mathbf{s}$ and take the NTT in Dilithium as an example. If the unknown NTT domain coefficients are the first m (this analysis applies to unknown coefficients in other positions), the specific analysis process unfolds as follows:

$$\Omega\Phi \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \end{pmatrix} \equiv \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \hat{s}_{k_m} \\ \vdots \\ \hat{s}_{k_{n-1}} \end{pmatrix} + \begin{pmatrix} \hat{s}_{un_0} \\ \vdots \\ \hat{s}_{un_{m-1}} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{7}$$

Where $\hat{\mathbf{s}}_k$ is a vector consisting of known coefficients with unknown position 0, while $\hat{\mathbf{s}}_{un}$ is a vector of unknown coefficients with known position 0. Equation (7) can be considered as an overdetermined system with m unknown coefficients and n equation relations regarding $\hat{\mathbf{s}}_{un}$ over a finite field q. Despite its complexity in solving the overdetermined system within the finite field, the constraints that Dilithium's private key within the integer range of $[-\eta,\eta]$ and the system must have a solution, allow us to propose a comparatively straightforward method. Firstly, select m fixed positions within $\mathbf{s}$ and propose a candidate value. Then, use Gaussian elimination to solve for the corresponding $\hat{\mathbf{s}}_{un}$. After substituting $\hat{\mathbf{s}}_{un}$ into Equation (7) to obtain $\mathbf{s}$, verify whether each coefficient of $\mathbf{s}$ lies within the $[-\eta,\eta]$. If all coefficients satisfy this condition, it is presumed that the correct private key has been acquired.

Although the Overdetermined system-based method is not as good as the subsequent lattice-based method, its biggest advantage is that it does not require the attacker to use sufficient cryptographic knowledge, and the enumeration space can be narrowed down from q^m to $(2\eta+1)^m$ by a simple transformation, which proves to be efficient when the number of unknown coefficients, denoted as m, is relatively small.

B. SIS-assisted method

The Small Integer Solution (SIS) problem was proposed by Ajtai in 1996 [31]. It aims to find a sufficiently short integer vector that when multiplied by a randomly selected integer matrix under an upper bound, results in a zero vector. The specific definitions are as follows:

SIS$_{n,q,\beta,m}$ **problem**: *Given m uniformly random vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$, forming the rows of a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^n$ of norm $\|\mathbf{z}\| \le \beta$ such that $\mathbf{Az} = \mathbf{0}$.*

When the elements of vector $\mathbf{z}$ are sampled from the uniform distribution over $[-\eta,\eta]$ and $\eta \ll q^{(m/n)}$, there is a high probability that a unique vector $\mathbf{z}$ exists such that $\mathbf{Az} = \mathbf{0}$ [32]. For the Inhomogeneous SIS (ISIS) problem $\mathbf{As} = \mathbf{t}$, we define $\mathbf{A}' = \mathbf{A}|\mathbf{t}$ and solve the SIS problem $\mathbf{A}'\mathbf{z} = 0$. If the solution $\mathbf{z}$ satisfies the form $(\mathbf{s}, -1)$, then we obtain the solution $\mathbf{s}$ to the ISIS problem.

We observe that the incomplete NTT domain recovery in Dilithium and Kyber can be viewed as an ISIS problem. Assume that the first m coefficients in NTT domain are known. Then $\mathbf{A}$ is the $m \times n$ matrix obtained from the transform matrix ($\Omega\Phi$ in Equation (1) for Dilithium, $\mathbf{Z}$ or $\mathbf{Z}_{\setminus\{0\}}$ in Equation (4,5) for Kyber) by removing the first m rows, $\mathbf{s}$ is the private key which is exactly a short integer vector, and $\mathbf{t}$ is the recovered NTT domain coefficients. Taking Dilithium2 as an example, when $d = 2$, $m > 128$ satisfies the condition $d \ll q^{(m/n)}$, which means that the attacker only needs to recover half of the NTT domain coefficients and the remaining one can be recovered using the SIS-assisted method.

The entire solution set of an SIS problem instance constitutes a q-ary lattice $\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{Ax} = 0 \pmod q\}$. Therefore, solving an SIS problem instance is equivalent to solving the corresponding SVP problem on the lattice $\Lambda^{\perp}(\mathbf{A})$ with volume q^m. The LLL and BKZ algorithm are commonly used to solve the SIS problem.

The LLL algorithm can be viewed as a special case of the BKZ algorithm with block size $b = 2$. The BKZ algorithm requires calling the LLL algorithm and a subroutine that solves the SVP problem on a lower dimensional lattice.

The key parameter for evaluating the BKZ algorithm is the block size b. As b increases, the quality of the reduced basis and the length of the shortest vector output by BKZ improve, but the running time also increases. In 2011, Chen and Nguyen proposed the BKZ 2.0 algorithm [33], which significantly improved the efficiency of the basis reduction algorithm in practice. Currently, in the core-SVP evaluation model, the complexity of the BKZ algorithm with block size b is approximately $O(2^{0.292b})$ in the classical computing model, and the approximate factor for the shortest vector output is

$$\lim_{n\to\infty} \delta \approx \left(\frac{b}{2\pi e}(\pi b)^{\frac{1}{b}}\right)^{\frac{1}{2(b-1)}} \tag{8}$$

Assume that the reduced basis obtained from running BKZ-b on n-dimensional lattice basis $\mathbf{B} \in \mathbb{Z}^{n\times n}$ satisfies the Geometry Series Assumption (GSA) when $b \ge 50$ [34], then we have $\|\hat{\mathbf{b}}_i^*\| = \delta^{n-2i-1} \cdot \det(\Lambda)^{1/n}$. Specifically, [35] show that when the norm of the projection of the unique shortest vector $\mathbf{z}$ into the space spanned by the last b orthogonal vectors is less than $\|\hat{\mathbf{b}}_{n-b}^*\|$, then the BKZ-$b$ basis reduction algorithm recovers $\mathbf{v}$. The norm of $\mathbf{v}$'s projection into the space spanned by b orthogonal vectors approximately equals $\sigma\sqrt{b}$, where σ is the standard deviation of the secret / error coefficients. In this case, the computational cost is mainly decided by the minimal value of b that satisfies the constraint

$$\sigma\sqrt{b} \le \delta^{2b-n-1} \cdot q^{m/n}. \tag{9}$$

Note that the specific values of m, n and q are crucial in determining the computational cost. It is straightforward to verify that the SIS problem becomes easier as either m or q increases. Conversely, it becomes harder as n increases.

For the specific SIS search problem instance corresponding to Dilithium and Kyber, we can use the .LLL and .BKZ functions in the fpylll library in the Sage software to solve the transformed SVP problem.

C. The dimensionality reduction strategy in SIS-assisted method

Directly for Dilithium and Kyber using the BKZ algorithm to solve for partial NTT domain private keys is theoretically feasible, but in the practical analysis, it is found that there is a large gap between the required NTT domain coefficients and the theoretical lower bound, and suffers from large time overhead in Dilithium. While the above problems are alleviated when the dimensionality is low, as demonstrated by Albrecht *et al.* [24], when Kyber768/1024 acquires 32 coefficients there is a high probability of recovering half of the private key. We note that the negative convolutions NTT, utilized by both Dilithium and Kyber, can be rewritten from a n-dimensional to $n/2$-dimensional NTT [24] using a divide and conquer strategy. To facilitate the subsequent formulation, let $\mathbf{s}^{(e)} = (s_0, s_2, \cdots, s_{n-2})$ and $\mathbf{s}^{(o)} = (s_1, s_3, \cdots, s_{n-1})$. In Dilithium, the transformation process is shown as follows:

$$2\mathbf{NTT}_{n/2}(\mathbf{s}^{(e)})_i = \mathbf{NTT}_n(\mathbf{s})_i + \mathbf{NTT}_n(\mathbf{s})_{i+n/2}$$

$$2\phi\omega^i\mathbf{NTT}_{n/2}(\mathbf{s}^{(o)})_i = \mathbf{NTT}_n(\mathbf{s})_i - \mathbf{NTT}_n(\mathbf{s})_{i+n/2}$$

which $i \in \{0, 1, \ldots, n/2-1\}$. Let W_{256} be the $\Omega\Phi$ in Equation (1) for example, the transformation results are shown below:

$$W_{256}^{(+)} = \begin{pmatrix} 2 & 0 & 2\phi^2 & 0 & \cdots & 2\phi^{254} & 0 \\ 2 & 0 & 2\phi^6 & 0 & \cdots & 2\phi^{250} & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 2 & 0 & -2\phi^{254} & 0 & \cdots & -2\phi^2 & 0 \end{pmatrix}$$

$$W_{128} = \begin{pmatrix} 1 & \phi^2 & \cdots & \phi^{254} \\ 1 & \phi^6 & \cdots & \phi^{250} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & -\phi^{254} & \cdots & -\phi^2 \end{pmatrix}$$

W_{128} is derived by dividing the matrix into two parts, front and back, each consisting of 128 rows. These parts are then added, all-zero columns are removed, and divide the remaining elements by 2.

Since the q used in Kyber exists only as a 256-th root of unity ζ, its conversion process is slightly different from that of Dilithium, analyzing the odd or even positions of Kyber's NTT domain private key coefficients, specifically:

$$2\mathbf{NTT}_{n/2}(\mathbf{s}^{(e)})_i = \mathbf{NTT}_n(\mathbf{s})_{2i} + \mathbf{NTT}_n(\mathbf{s})_{2i+1}$$

$$2\zeta^{2br_7(2i)+1}\mathbf{NTT}_{n/2}(\mathbf{s}^{(o)})_i = \mathbf{NTT}_n(\mathbf{s})_{2i} - \mathbf{NTT}_n(\mathbf{s})_{2i+1}$$

which $i \in \{0, 1, \ldots, n/2-1\}$. Taking the NTT matrix of $\mathbf{Z}_{\setminus\{0\}}$ in Equation (5) as an example, the result is shown below:

$$W_{128}^{(+)} = \begin{pmatrix} 2 & 0 & 2\zeta^2 & 0 & \cdots & 2\zeta^{126} & 0 \\ 2 & 0 & -2\zeta^2 & 0 & \cdots & -2\zeta^{126} & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 2 & 0 & -2\zeta^{126} & 0 & \cdots & -2\zeta^2 & 0 \end{pmatrix}$$

$$W_{64} = \begin{pmatrix} 1 & \zeta^2 & \cdots & \zeta^{126} \\ 1 & -\zeta^2 & \cdots & -\zeta^{126} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & -\zeta^{126} & \cdots & -\zeta^2 \end{pmatrix}$$

W_{64} is derived by splitting the matrix into two parts based on the parity of the rows. After adding these parts, all-zero columns are removed, and divide the elements by 2.

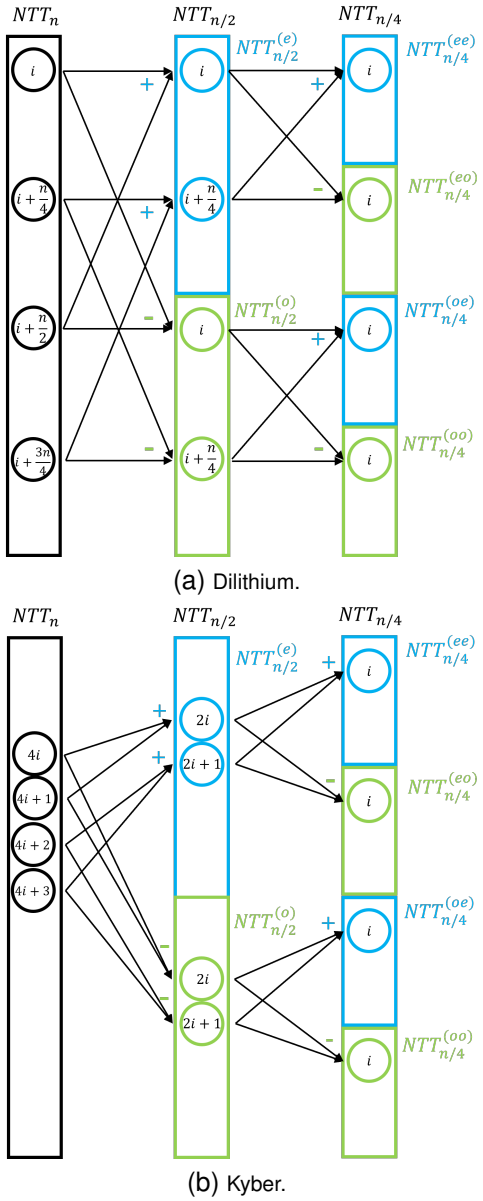


Fig. 1: Transform process of the values under NTT domain.

With the help of this transformation, the original NTT process can be split and converts into low-dimensional NTT. In order to understand more intuitively, Fig.1 shows the transformation process of the values under NTT domain, when the value of a specific position under NTT domain is obtained, the value corresponding to a specific position of the low-dimensional matrix can be calculated. At this time, the dimension of the SIS search problem that the attacker needs to solve is reduced to $n/4$. According to the analysis in SecIII-B, using LLL or BKZ algorithm to solve it at this time will achieve better performance. If the minimum coefficients required to solve the SIS search problem corresponding to $\mathrm{NTT}_{n/4}$ is k, then the normal domain private key can be

recovered after obtaining $4k$ NTT domain coefficients that satisfy the specific coefficients.

TABLE III: Theoretical minimum coefficients required for each dimension of Dilithium

	NTT_{256}	NTT_{128}	NTT_{64}	NTT_{32}
Dilithium2,5	26	13	7	4
Dilithium3	36	18	9	5
Kyber512	-	31	16	8
Kyber768,1024	-	26	13	7

For any dimensional NTT_i computation process, it is theoretically necessary to ensure that the known NTT domain coefficients m satisfy the condition $q^m > (2\eta+1)^i$ in order to perform the inverse transformation correctly. Tab.III gives the theoretical minimum coefficients required for each dimension to be able to perform the inverse transformation correctly under different partitioning strategies of Dilithium and Kyber. We wish to solve the problem using the number of coefficients in the NTT domain as close as possible to the theoretical limit.

The results shown in Sec.V-C also indicate that, for solving the original SIS problems extracted from Dilithium and Kyber, the minimum required number of coefficients m and the time overhead t are significantly greater than those for the reduced-dimension SIS problems using the same algorithm. The introduction of the dimension reduction strategy allows an attacker to quickly recover the complete private key with only a small number of NTT domain private key coefficients, greatly expanding practical application scenarios. Furthermore, under more stringent conditions—where NTT domain coefficients are obtained with a success rate of p but cannot be verified for correctness—we can use an enumeration method to randomly select known NTT coefficients for solving. The probability of obtaining the correct result is p^m, and the average time required to obtain the complete coefficients is t/p^m.

IV. Recovering NTT domain private keys using SCAs

Attackers may extract sensitive information from a cryptographic system during the execution through various means, including SCAs, cache attacks, and cold-boot attacks. In this section, we will discuss the application of SCAs to swiftly recover the NTT domain private keys for Dilithium and Kyber.

A. CPA for Dilithium and Kyber

Dilithium and Kyber employ point-wise operations using NTT domain private keys, as illustrated in Alg.1 line 11 and Alg.2 line 7, respectively. The reference implementation of $\hat{\mathbf{c}} \circ \hat{\mathbf{s}}_1$ for Dilithium, submitted to NIST, is depicted in Fig.2, Kyber's implementation is fundamentally identical. Conducting SCAs on these operations using CPA is theoretically possible. In fact, Yang *et al.* [19], working with Kyber where $||\mathbb{K}|| = 3329$, successfully recovered the private key in a matter of minutes using CPA. However, the task becomes computationally very difficult for Dilithium, where $||\mathbb{K}|| = 8380417$, as this renders exhaustive enumeration of the entire key space impractical.

```
1 int32_t montgomery_reduce(int64_t a) {
2   int32_t t;
3   t = (int32_t)a*qinv;
4   t = (a - (int64_t)t*q) >> 32;
5   return t;}
6 void poly_point-wise_montgomery(poly *c, const poly
      *a, const poly *b) {
7   unsigned int i;
8   for(i = 0; i < N; ++i)
9     c->coeffs[i] = montgomery_reduce((int64_t)a->
          coeffs[i] * b->coeffs[i]);}
```

Fig. 2: Dilithium $\hat{\mathbf{c}} \circ \hat{\mathbf{s}}_1$ reference implementation.

Tunstall *et al.* [25] outlined the use of CPA to attack large word sizes, successfully extracting the DES key implemented on a 32-bit platform. The approach involves segmenting large byte intermediate values, and then using CPA to recover each segment independently. Given an intermediate value $f(x_1, \ldots, x_p, k)$, where k is the l-bit unknown fixed value and x_i is a known change value, we divide k into consecutive blocks $b_{n-1}, \ldots, b_0$ from the most significant the least significant bit, with each block being l_{b_i} bits. Consequently, the PCC for each block can be computed as follows:

$$\rho(L, H_{k^*/b_i}) = \rho(L, H_{k^*})\sqrt{\frac{l_{b_i}}{l}} \tag{10}$$

The practical attack procedure is as follows:

(1) Compute the PCCs for all possible values of b_0, rank them in descending order, and select the top h_0 as candidates.
(2) Merge the h_0 candidate values with b_1 to generate a new set of candidates, and then calculate the PCCs and sort them in descending order. Finally, pick the top h_1 candidate values.
(3) Recursively perform step (2) for the remaining b_i ($2 \leq i < n-2$).
(4) Combine the h_{n-2} candidate values with b_{n-1}, calculate all possible PCCs, and choose the candidate that corresponds to the highest value as the result.

This strategy reduces the enumeration requirement from 2^l values to $2^{b_0} + h_0 \cdot 2^{b_1} + \cdots + h_{n-2} \cdot 2^{b_{n-1}}$. For instance, with $l = 32$, $h_i = 8$, and each b_i being 8-bit, the enumeration space is trimmed from 2^{32} to under 2^{13}. This approach can also be employed for analyzing large number multiplication operations. The private key can be recovered from low to high blocks in turn, while the carry operation within the multiplication makes it easier to distinguish the correct candidate value of the high block. While choosing smaller blocks might seem to expedite the attack, Equation (10) indicates that this also results in a smaller PCC calculated in step (1).

Chen *et al.* [21] noted that the output from the Montgomery reduction (as seen in Fig.2, line 5) constitutes a distinct leakage. Utilizing CPA with this leakage, dubbed the Conservative scheme, they achieved a 100% success rate using 157 power traces. Since this calculation process involves a shift operation, it precludes the direct application of Tunstall *et al.*'s method. Consequently, Chen *et al.* opted to divide the NTT domain into high and low two blocks. They employed CPA on the Montgomery reduction input (Fig.2, line 9) to recover the

lower block, then used the output to recover the NTT domain private key. When the results are not accurate, the Conservative scheme is enacted to guarantee 100% accuracy in the NTT domain results, leading to the naming of this approach as the Hybrid scheme. The work of Chen *et al.* still uses the traditional scheme in order to achieve 100%, which leads to a huge time overhead.

B. Efficient SCA scheme for NTT domain private keys

The introduction of an incomplete NTT domain recovery method allows us to relax the requirement for correctness in exchange for efficient attacks. We design efficient SCA schemes against Dilithium and Kyber. Our scheme selects linear operations as the target, giving lower-order bits candidates with the help of Tunstall *et al.* [25]'s method, and later selects nonlinear operations with more significant leakage in order to recover the complete private key.

Algorithm 3 CPA in NIST Reference Implementations of Dilithium and Kyber

Require: th_{sel}, th_{fin}, mcn, num_{blocks},L_m
1: Initialize CS to an empty set ▷ Candidate Set
2: **for** $i \leftarrow 0$ **to** $num_{blocks} - 1$ **do**
3: $CS \leftarrow \text{combine}(CS, [2^{i \cdot m}, 2^{(i+1) \cdot m}])$ ▷ Combine lower-order bits
4: $temp \leftarrow \text{sort}(\text{CPA}(H_{middle}(CS)), L_m)$ ▷ Sort by CPA of middle part
5: $CS \leftarrow temp \geq th_{sel} \cdot \max(temp)$
6: $CZS \leftarrow \text{combine}(CS, 0)$ ▷ Candidate with Zero Set
7: **end for**
8: **for each** k **in** (CS, CZS) **do**
9: **if** $\text{CPA}(H_{final}(k), L_m) > th_{fin}$ **then**
10: **return** k and **break**
11: **end if**
12: **end for**
13: **return** -1 ▷ Indicate failure in key recovery

Alg.3 gives the attack process. Designed for practical attack scenarios, this algorithm introduces selection thresholds (th_{sel}), a maximum candidate number (mcn) for the Candidate Set (CS) size, and specifies the number of lower-order bit blocks (num_{blocks}), each block containing m bits. We establish a (CS) for the key by applying CPA to these values. The candidate key is iteratively computed by selecting the intermediate value of the linear transformation in the Montgomery reduction, though false positives can occur in this process. For instance, when the lowest block is zero ($b_0 = 0$), as in 0x23450, the result would erroneously be $b_{3\sim0} =$ 0x2345. This issue is addressed by zero-padding the existing candidate values during low-order bit attacks and storing the data in a Candidate with Zero Set (CZS). For the higher-order bits, the CS and CZS are merged, and the most distinct leakage is identified through CPA to accurately reconstruct the entire key.

Practical situations may necessitate a balance between success rate and attack speed, adjustable through mcn and th_{sel}. If traces are limited, increasing mcn or decreasing th_{sel} enhances the success rate. Alternatively, abundant traces allow for attack speed optimization by reducing mcn or increasing th_{sel}. During practical attacks, we noticed that the Montgomery reduction's shift and storage operations made the PCC of the correct key substantially higher than that of incorrect candidates (see line 5 in Fig.2). This characteristic enables the setting of a threshold, th_{fin}, to expedite computation.

It should be noted that when using overdetermined system-based and SIS-assisted methods, both of them need to know which NTT domain private key coefficients are correct, while our method can easily determine the current coefficients to attack successfully by setting a threshold. At this point, the attacker only needs to quickly recover the number of correct coefficients that satisfy the condition and then call the above mathematical method to complete the attack.

V. Experiments and Results

A. Masked implementation of Dilithium and Kyber

There has been some work proposing protected implementations of Dilithium and Kyber. For Dilithium, Azouaoui *et al.* [36] proposed a protection strategy for intermediate computations, which considers the physical security requirements and classifies intermediate values into three categories: resist Differential Power Analysis (DPA), resist SPA, and publicly known. Fig.3 depicts a first-order masking scheme of polynomial multiplication, where each coefficient of the $\mathbf{s}$ is split into two arithmetic shares, $\mathbf{s}^0$ and $\mathbf{s}^1$, during the key generation procedure. The NTT operation is then independently applied to $\mathbf{s}^0$ and $\mathbf{s}^1$. This is followed by point-wise multiplication with the signature $\mathbf{c}$ in the NTT domain. Coron *et al.* [37] proposed some gadgets to improve the efficiency of the algorithm based on [36] and gave open source code, which is used in this experiment for the implementation.

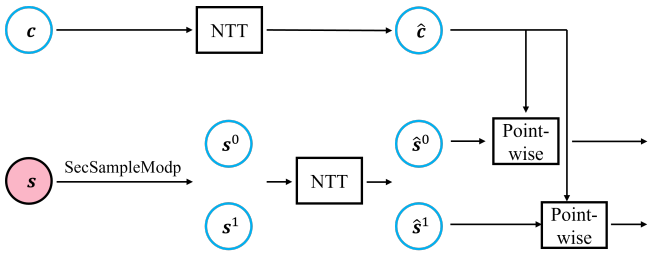


Fig. 3: Procedure for masked protection implementation of polynomial multiplication in $\mathbf{cs}$. Red: DPA protection is required. Blue: No side-channel protection is needed.

For Kyber, since the scheme proposed by Bos *et al.* [38] is not open-source yet, this experiment utilizes the implementation by Heinz *et al.* [39]. They provided a mask-protected complete implementation based on the unprotected Cortex-M4 optimized implementation in the PQM4 project [18]. For the polynomial multiplication of the private key $\mathbf{s}_k$ with the ciphertext $\mathbf{c}$ during the decapsulation process, the implementation follows the same structure as shown in Fig.3.

It is important to note that the private key should be refreshed each time it is called. However, the open-source implementation in [39] does not include this operation. We

did not modify the implementation, and our subsequent results highlight the necessity of the refreshing operation.

B. Set up

The experiments conducted on unprotected algorithms employed the open-source, C-based implementation that was submitted to NIST. Notably, the open-source implementation of Kyber, optimized for the ARM Cortex-M4 platform using assembly language, does not have separate experimental results presented herein due to its polynomial multiplication process being nearly identical to that of masked Kyber [39].

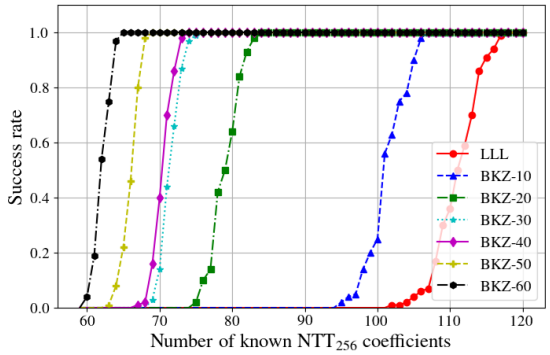
Our experimental setup includes a WR610Zi oscilloscope connected to a BLP1.9+ low-pass filter and a PA303 preamplifier, enabling the capture of power traces at a sampling rate of 100 MSa/s. The compiled code was executed on a ChipWhisperer UFO development board fitted with an STM32F405GTx microcontroller. For data analysis, we used servers equipped with Intel(R) Xeon(R) Platinum 8268 CPUs. Analytical tools were implemented using Python 3.9 and SageMath 9.7. Each experiment was repeated 100 times, and the average value of these repetitions was taken as the result.

The high-dimensional characteristics of the lattice allow many attacks to be executed in parallel to improve efficiency. However, to provide a more straightforward comparison, our practical attacks were conducted without parallelism.

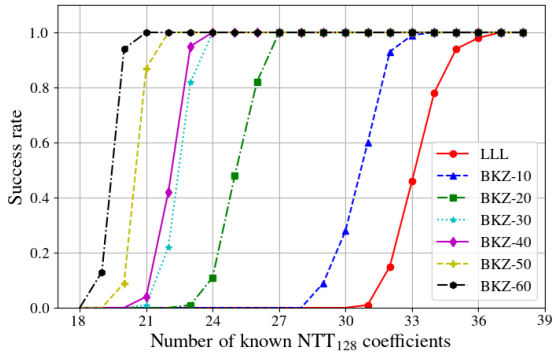
C. Results of incomplete NTT domain recovery methods

While the time overhead for the overdetermined equation-based method is exponential, it is efficient when there are few unknown coefficients. For schemes with a normal domain $\eta = 2$ (e.g., Dilithium2, 5, and Kyber768, 1024), the recovery of the private key takes under 10 minutes when there are no more than 10 unknown coefficients and less than 1 second when the number is below 6. For schemes with a normal domain $\eta = 3$ (e.g., Dilithium3 and Kyber512), the time overhead ranges between 2 to 5 minutes for 7 unknown coefficients, escalating exponentially as the number increases.

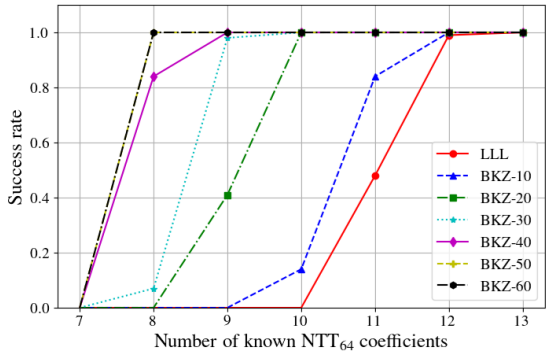
For the SIS-assisted method, we experimented on Dilithium2, 3, 5 using LLL and BKZ algorithms with different block sizes under various dimensionality reduction strategies. Fig.4 and 5 show the SR of correctly solving the SIS problem extracted from the NTT process in Dilithium. The results indicate that, BKZ requires fewer NTT domain coefficients than LLL with a 100% SR. Moreover, the required coefficients decrease as the block size increases. For recovering a set of normal domain private keys of Dilithium2 and 5, LLL requires 118 NTT domain coefficients, whereas BKZ-40 and BKZ-60 require only 75 and 66 coefficients, reducing the number by 36% and 44%, respectively. Using the same algorithm, the required coefficients are also significantly reduced after applying the dimensionality reduction strategy. For example, with BKZ-60, compared to needing 77 coefficients for NTT_{256} to recover a set of normal domain private keys of Dilithium3, the NTT_{128} and NTT_{64} schemes require only 25 and 10 known coefficients, corresponding to 50 and 40 NTT domain coefficients, reducing by 29% and 48%, respectively. Overall, after using the dimensionality reduction strategy, the number of known NTT domain coefficients required for solving is significantly reduced. For NTT_{64}, the number of coefficients required by BKZ-60 is already very close to the theoretical lower bound listed in Tab.III. The number of NTT domain private keys required to recover the complete private keys of Dilithium2 and 5 is less than that for Dilithium3, consistent with our analysis in Sec.III-B.



(a) Result of NTT_{256}.



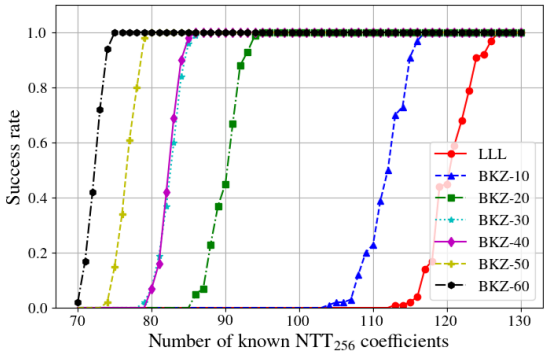
(b) Result of NTT_{128}.



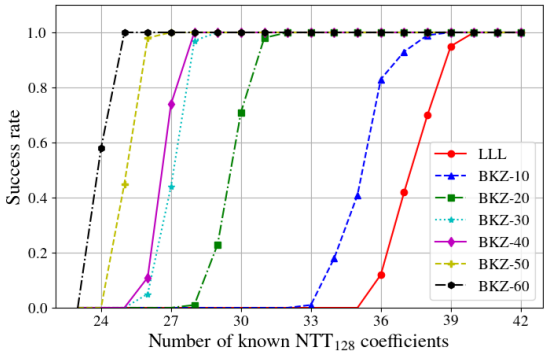
(c) Result of NTT_{64}.

Fig. 4: SR of solving the SIS problem in Dilithium2,5 NTT.

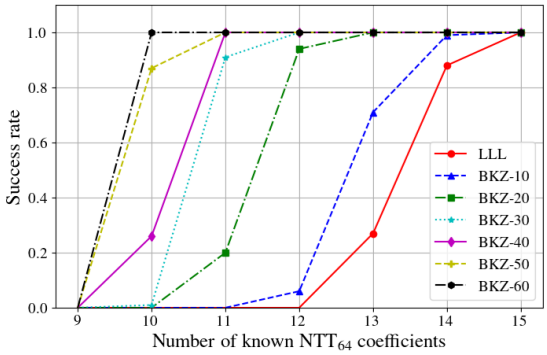
Fig.6 illustrates the time costs for solving the SIS search problem of Dilithium2, 5 using the LLL and BKZ, without employing the dimensionality reduction strategy. The time overhead is generally higher with fewer known coefficients for both algorithms. As the number of known coefficients increases beyond 190, the time overhead decreases. LLL consistently requires less time than BKZ, whose execution time escalates with increasing block size. Fig.6 only displays scenarios solvable within 20 hours; however, the time overhead for BKZ-60 can extend up to 82 hours. Although larger block



(a) Result of NTT_{256}.



(b) Result of NTT_{128}.



(c) Result of NTT_{64}.

Fig. 5: SR of solving the SIS problem in Dilithium3 NTT.

TABLE IV: Time required for SIS-assisted method in Dilithium (s)

	SIS-assist	NTT_{256}	NTT_{128}	NTT_{64}
Dilithium2,5	LLL	1,073.5	8.1	3.4
	BKZ-40	9,936.4	13.8	3.5
	BKZ-60	295,451.7	662.6	62.2
Dilithium3	LLL	1,167.2	8.4	3.0
	BKZ-40	11,006.7	14.3	3.5
	BKZ-60	280,130.7	677.7	64.2

sizes reduce the number of required known coefficients, they may result in prohibitively high time costs.

Tab.IV illustrates the time required to achieve a 100% SR in fully recovering the $\mathbf{s}_1$ for Dilithium2, 3, and 5 using LLL, BKZ-40, and BKZ-60 under various dimensionality reduction strategies, with the minimal number of coefficients needed. The results for NTT_{128} and NTT_{64} include the cumulative overhead of executing the solving algorithm twice

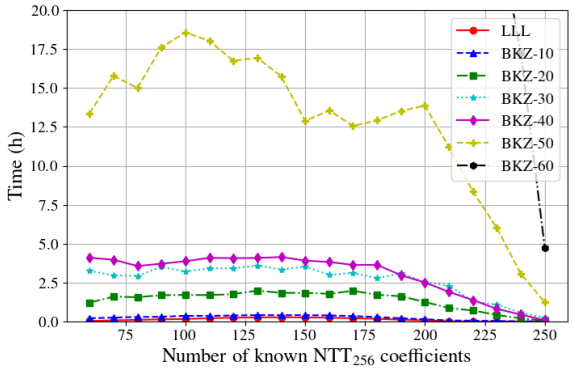
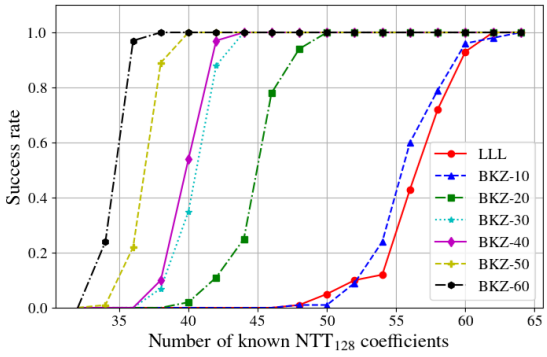


Fig. 6: Time cost of Dilithium2,5 with LLL and BKZ.

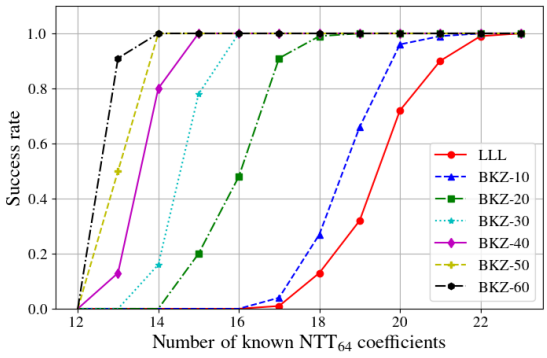
and four times, respectively. Notably, the time required for all three algorithms significantly decreases. For example, under the NTT_{128} and NTT_{64} configurations, BKZ-40 only requires 13.8 and 3.5 seconds, respectively, to complete the attack—compared to about 2.7 hours for the original NTT_{256} scheme. These reductions represent speed improvements of approximately 720 and 2,800 times, respectively, significantly enhancing attack efficiency.

We also conducted similar experiments on Kyber. Sec.II-C describes how Kyber can be divided into lower 128-dimensional SIS problems. We applied the LLL and BKZ algorithms on NTT_{128} and NTT_{64} configurations for Kyber512, 768, and 1024, with results shown in Fig.7 and 8. As with Dilithium, BKZ outperforms LLL. Particularly with BKZ-60, for NTT_{128}, Kyber512 and Kyber768/1024 require only 38 and 34 coefficients to recover the half private key, which represents a reduction of 41% and 45% compared to the 64 and 62 coefficients required by LLL. The dimensionality reduction strategy further enhanced results: for Kyber512 and Kyber768/1024 using BKZ-60 under the NTT_{64} strategy, only 28 and 24 coefficients are needed to recover half of the private key, marking a decrease of 26% and 29% compared to NTT_{128}. Remarkably, the results for NTT_{64} are already below the $q^m > 5^i$ case. This may due to the fact that Kyber's private key distribution follows a centered binomial distribution.

Due to the smaller dimension of the SIS problem, Kyber's time overhead is significantly lower than Dilithium's. Fig.9 presents the time required to execute the LLL and BKZ algorithms at different block sizes for Kyber768/1024. As the block size of BKZ increases, so does the time overhead. However, once the number of known coefficients surpasses 105, the time overhead for any algorithm begins to decrease. The maximum time overhead for BKZ-60 is approximately 6 minutes. When the block size was further increased, under NTT_{128}, BKZ-70 required only 31 known coefficients to recover half of the private key set for Kyber768/1024, but the required time extended to 1.5 hours. Tab.V displays the time overhead needed to achieve a 100% SR in recovering $\mathbf{s}_k$ of Kyber with the minimum number of coefficients. The application of the dimensionality reduction strategy significantly reduces the required time. For instance, BKZ-60 under the NTT_{64} configuration for Kyber512 necessitates only 64.6 seconds, compared to about 664.7 seconds for NTT_{128}, marking a

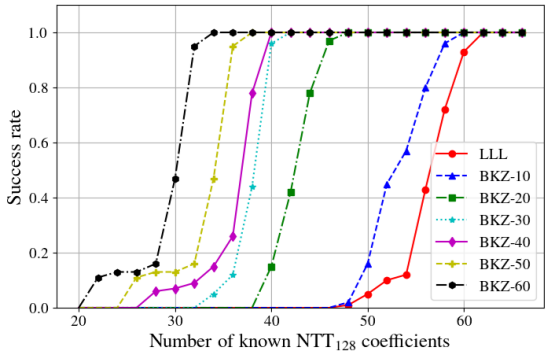


(a) Result of NTT_{128}.

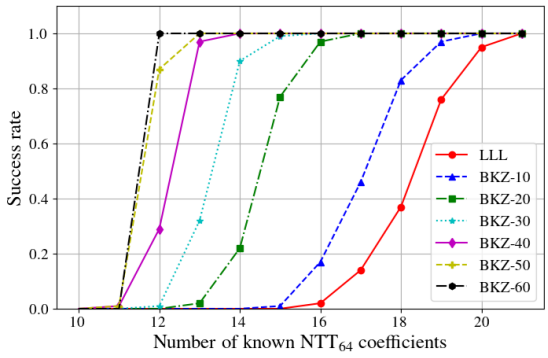


(b) Result of NTT_{64}.

Fig. 7: SR of solving the SIS problem in Kyber512 NTT.



(a) Result of NTT_{128}.



(b) Result of NTT_{64}.

Fig. 8: SR of solving the SIS problem in Kyber768,1024 NTT.

tenfold increase in speed.

TABLE V: Time cost for SIS-assisted method in Kyber (s)

	SIS method	NTT_{128}	NTT_{64}
Kyber512	LLL	8.6	4.3
	BKZ-40	14.3	3.6
	BKZ-60	664.7	64.6
Kyber768,1024	LLL	7.5	4.1
	BKZ-40	14.3	3.4
	BKZ-60	641.1	61.8

Overall, using BKZ yields better results compared to LLL, and the advantage of BKZ becomes more pronounced as the block size increases. However, the corresponding time overhead also significantly increases. Even using BKZ-60, Kyber can be attacked in just a few minutes. In contrast, for the higher-dimensional Dilithium, executing BKZ-60 takes over 80 hours, and the minimum number of NTT domain coefficients required is far greater than the theoretical minimum. The dimensionality reduction strategy can significantly reduce time overhead and the required number of coefficients, approaching theoretical limits, thereby proving the effectiveness of the dimensionality reduction strategy.

D. SCAs of unprotected Dilithium and Kyber

In this experiment, we perform practical SCAs on unprotected implementations of Dilithium and Kyber. Both algorithms employ the Montgomery reduction operation, which is executed in a similar manner in their respective C reference implementations for consistency. For clarity, Fig.10 illustrates the specific process used in Dilithium, serving as an example.

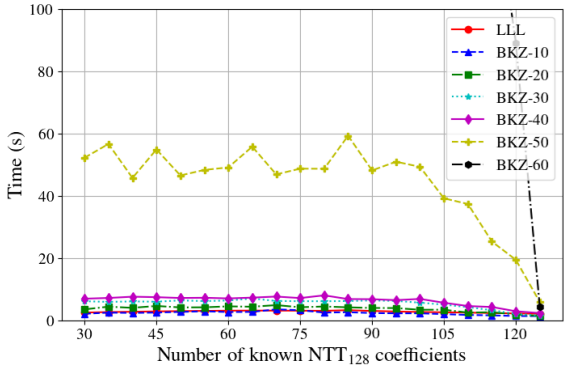


Fig. 9: Time cost of Kyber768,1024 with LLL and BKZ.

Using the HW model, we analyze the Pearson correlation coefficients (PCCs) between intermediate values and power traces, as shown in Fig.11. With a dataset of 5,000 traces, both temp1 and temp2 exhibit PCC peaks at sample index 300, attributed to their shared lower 12 bits. However, temp1's leakage peak appears slightly earlier and is less pronounced. Since the last 32 bits of temp3 are identical of temp1, they are omitted in Fig.10. The highest PCC value for temp4, nearing 1, is observed at sample index 380, likely resulting from the nonlinear shift operation followed by an STR operation that stores the result in memory. In our methodology, outlined in Alg.3, temp2 is used to calculate the last 20 bits, while temp4 is utilized to recover the final result. In this setup, we allocate 4 bits per block, set a final threshold (th_{fin}) of 0.8, and identify

```
temp1 = int64(NTT(c)*NTT(s1))
temp2 = int32(temp1*qinv)
temp3 = int64(temp2*q)
temp4 = (temp1-temp3)>>32
```

Fig. 10: Intermediate values of Montgomery reduction operation in Dilithium.

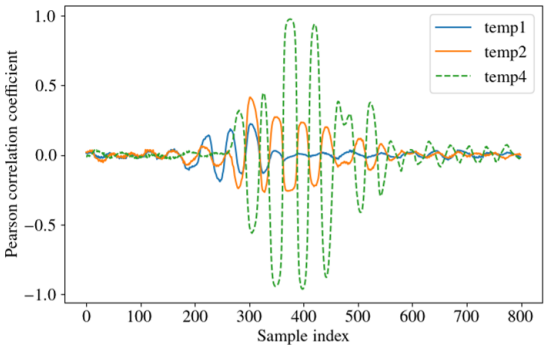


Fig. 11: PCCs of the temp1 and temp2 in Dilithium2.

20 points near the highest PCC as points of interest (PoI).

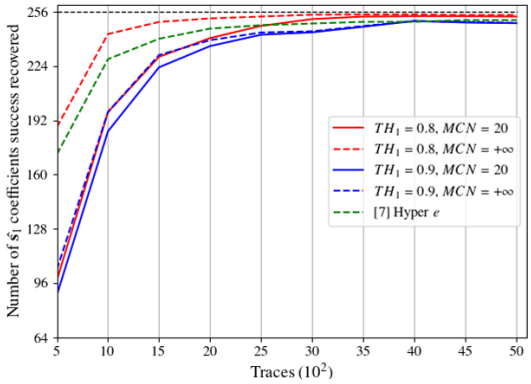


Fig. 12: $\hat{\mathbf{s}}_1$ coefficients recovered.

Following the CPA scheme detailed in Sec.IV-B, we set $th_{sel} = 0.8$ or 0.9, and $mcn = 20$ or $+\infty$. Additionally, we replicate the optimal Hyper e scheme by Chen *et al.* [21] for comparison. Fig.12 displays the number of recovered $\hat{\mathbf{s}}_1$ coefficients using varying trace counts. At $th_{sel} = 0.9$, different mcn values have a marginal effect on the number of coefficients recovered; around 180 coefficients are retrieved with 1,000 traces, increasing to approximately 250 with 4,000 traces. For $th_{sel} = 0.8$ with $mcn = 20$, the trend is similar to $th_{sel} = 0.9$ for fewer traces, but it facilitates the recovery of more coefficients when using over 2,000 traces, stabilizing at 253 coefficients with 4,000 traces. When $mcn = +\infty$, recovery is notably more efficient with smaller trace counts—about 190 coefficients with 500 traces and over 250 with 2,000 traces. In comparison, Chen *et al.*'s method recovers approximately 170 coefficients with 500 traces, increasing to 251 with 5,000 traces. The SR of all schemes increases sharply with fewer than 2,000 traces and stabilizes beyond 4,000.

Tab.VI shows the SCAs result for Dilithium2 using different CPA parameters, compared with the optimized Enumeration and Hyper e strategies by Chen *et al.* [21], including the SR and time for recovering a single coefficient. Additionally, it provides the time overhead for recovering the complete $\mathbf{s}_1$ using only SCA. In this process, the attacker first uses the optimized scheme to attack all coefficients and then performs the enumeration scheme for incorrectly recovered coefficients. Using 5,000 traces, our SCA SR is comparable to or even better than Chen *et al.*'s [21] Hyper e method. Importantly, our method significantly reduces the time overhead, with the speed of attacking a single coefficient improved by 7 to 40 times. With a minimum of just 1.7 hours, we can recover the complete $\mathbf{s}_1$, whereas the Hyper e strategy requires 11.3 hours. Although the Enumeration method can ensure a 100% SR with only 200 traces, its time cost is prohibitively high, taking about 6 days.

TABLE VI: SCAs result of Dilithium2

Method	#Traces	SCA SR	Time[1](s)	Time[2](h)
$t_{sel} = 0.8$ $mcn = 20$	5,000	99.1	1.7	1.7
$t_{sel} = 0.8$ $mcn = \infty$		99.2	4.1	2.2
$t_{sel} = 0.9$ $mcn = 20$		97.4	0.7	3.5
$t_{sel} = 0.9$ $mcn = \infty$		97.7	0.8	3.3
[21] Hyper e		98.1	28.3	10.6
[21] Enumerate	200	100	463.7	131.9

[1] The time overhead to recover a single NTT domain coefficient.
[2] The time overhead to recover the complete $\mathbf{s}_1$ of Dilithium.

TABLE VII: Time to recover Dilithium $\mathbf{s}_1$ (min)

Security level	SIS method	NTT_{256}	NTT_{128}	NTT_{64}
Dilithium2	LLL	12.6	4.9	3.4
	BKZ-40	80.7	4.2	2.4
Dilithium3	LLL	16.2	7.3	4.3
	BKZ-40	84.5	5.3	3.0
Dilithium5	LLL	23.2	8.3	5.9
	BKZ-40	134.2	7.2	4.1

Tab.VII shows the minimum time required to recover the complete private key $\mathbf{s}_1$ of Dilithium using SCAs combined with the SIS-assisted method. The attacker first performs SCAs with parameters $t_{sel} = 0.9$ and $mcn = 20$ to recover partial coefficients. Then, different SIS-assisted methods such as LLL and BKZ-40, along with various dimensionality reduction strategies, are used to recover $\mathbf{s}_1$. It is important to note that the number of coefficients recovered via SCA varies depending on the algorithm and strategy to achieve the minimum time.

For Dilithium2, with a $\mathbf{s}_1$ subset of 256 coefficients, using the LLL algorithm without the dimensionality reduction strategy, 220 coefficients need to be recovered via SCA, it takes 12.6 minutes to get $\mathbf{s}_1$. With the dimensionality reduction strategy, NTT_{128} and NTT_{64} require only 80 and 60 coefficients, respectively, reducing the time for a complete attack to4.9 and 3.4 minutes. Compared to Chen *et al.*'s [21] 11.3 hours using only SCA, these methods are 53.8, 138.4, and 199 times faster.

The time overhead of solving the high-dimensional SIS problem using BKZ is high, and the no-dimensionality reduction strategy requires obtaining as many coefficients as possible in the SCA phase in order to achieve the minimum total attack time.. However, with NTT_{128} and NTT_{64} dimensionality reduction strategies, SCA needs to recover only 60

and 40 coefficients, respectively, with total times of 4.2 and 2.4 minutes. Similar results are seen for Dilithium3 and 5. Using LLL or BKZ algorithms improves attack efficiency, and incorporating the dimensionality reduction strategy further enhances this advantage.

```
1 void basemul(int16_t r[2], const int16_t a[2], const
      int16_t b[2], int16_t zeta){
2   r[0]  = fqmul(a[1], b[1]);
3   r[0]  = fqmul(r[0], zeta);
4   r[0] += fqmul(a[0], b[0]);
5   r[1]  = fqmul(a[0], b[1]);
6   r[1] += fqmul(a[1], b[0]);}
```

Fig. 13: Kyber $\hat{\mathbf{c}} \circ \hat{\mathbf{s}}_k$ reference implementation.

Kyber undergoes seven rounds of butterfly transformations, resulting in five Montgomery reductions for each coefficient pair, as depicted in Fig.13. Xu *et al.* [5] analyzed the impact of selecting different operations for attack. For consistency in evaluating the efficiency of the attack under the same experimental setup, we opted to target the operation corresponding to coefficient multiplication for SCAs, specifically, lines 2 and 4 in Fig.13. The parameter choices for the CPA scheme are identical to those used for Dilithium, with the modification of determining the lower 8 bits.

TABLE VIII: SCAs result of Kyber512

Method	#Traces	SCA SR	Time[1](s)	Time[1](s)
$t_{sel} = 0.8$ $mcn = 20$	5,000	98.4	0.2	145.3
$t_{sel} = 0.8$ $mcn = \infty$		99.2	0.3	164.1
$t_{sel} = 0.9$ $mcn = 20$		95.3	0.2	95.6
$t_{sel} = 0.9$ $mcn = \infty$		95.7	0.2	89.8
[5] Random ciphertext	200	100	0.4	213.2

[1] The time overhead to recover a single NTT domain coefficient .
[2] The time overhead to recover the complete $\mathbf{s}_k$ of Kyber.

Tab.VIII shows the results of recovering a single coefficient and the complete private key of Kyber512 using only SCAs. With parameters $th_{sel} = 0.9$ or 0.8, about 175 and 210 coefficients were recovered using 500 traces, respectively. The number of recovered coefficients increased with the number of traces and stabilized around 4000 traces. Under the same conditions, we replicated the random ciphertext attack of [5] and included the results in Tab.VIII. Although Xu *et al.*'s [5] method is highly efficient, recovering coefficients in 0.4 seconds, our method doubled the recovery speed while maintaining a high SR.

We also conducted practical attacks to recover the complete $\mathbf{s}_k$ of different Kyber schemes, including SCAs and solving SIS problems. Tab.IX shows the results using LLL and BKZ-40 algorithms without the dimensionality reduction strategy and with NTT_{64}. Due to the lower dimension of the SIS problem, the time overhead for LLL and BKZ-40 is relatively small, with the SCA time being the main factor affecting the total time overhead. After applying the dimensionality reduction strategy, the attack time for both algorithms was significantly reduced. For example, for Kyber512, compared to Xu *et al.* [5], our method reduced the time overhead from 213.2 seconds to 28.7 seconds.

TABLE IX: Time to recover Kyber $\mathbf{s}_1$ (s)

Security level	SIS method	NTT_{128}	NTT_{64}
Kbyer512	LLL	67.6	43.4
	BKZ-40	75.7	28.7
Kbyer768	LLL	88.8	66.9
	BKZ-40	88.7	47.6
Kbyer1024	LLL	118.3	89.3
	BKZ-40	116.3	63.5

Our practical attacks on Dilithium and Kyber show that while Kyber can be feasibly attacked using only side-channel analysis, the time overhead for Dilithium is enormous. Moreover, this process requires obtaining all NTT domain coefficients, which may be challenging in attacks such as fault or cache attacks. The introduction of SIS-assisted methods and the dimensionality reduction strategy not only reduces the SCA requirements for the adversary but also significantly enhances the attack speed. In our implementation, the $\mathbf{s}_1$ of any security level of Dilithium can be recovered within 6 minutes.

E. SCAs of masked Dilithium and Kyber

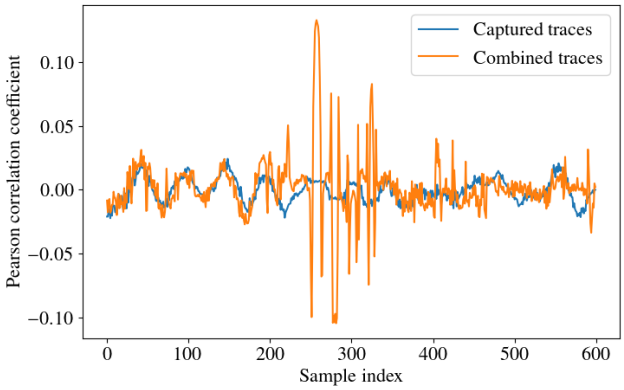


Fig. 14: PCCs for the captured and combined traces of masked Dilithium2.

For the implementation of first-order masked Dilithium with key refreshing [36], we conducted practical attacks using higher-order CPA. By combining the leaks L_0 and L_1 corresponding to the two shares $\mathbf{s}^0$ and $\mathbf{s}^1$, we obtained the higher-order leakage $L^* = (L_0 + L_1)^2$ for the attack. Fig.14 shows the PCC between the original traces, the combined leakage L^*, and the sensitive intermediate value temp4 using 5,000 traces. The original traces did not show significant correlation, while the combined L^* had a max correlation coefficient of 0.13, indicating that this combination method is effective and can be used for SCA. We also attempted to detect the combined traces for temp1-3 but did not find significant leakage, so we had to use the enumeration method.

Tab.X shows the SR, Guess Entropy (GE), and time overhead for recovering a single coefficient of the Dilithium NTT domain $\mathbf{s}_1$ with different numbers of traces. As the number of traces increases, the SR of the SCA gradually improves, reaching 100% at 5,000 traces. However, the time overhead also reaches its maximum at this point, with approximately

TABLE X: Time to recover one NTT domain coefficient of masked Dilithium (min)

Traces	1,000	2,000	3,000	4,000	5,000
SR (%)	9.7%	41.2%	90.3%	99.2%	100%
GE	39,964.1	2,577.9	1.5	1.0	1.0
Time (min)	19.7	27.6	33.7	39.7	47.8

47.8 minutes required to recover a single coefficient. It was observed that using 3,000 traces, the SR exceeds 90%, and the GE is 1.5, indicating that most coefficients can be correctly recovered, and the rankings of unrecovered coefficients are high. For each coefficient, we first use 3,000 traces to obtain a set of candidates (50 candidates in this experiment), and then use 5,000 traces to determine the final value. Despite this, the time cost is still unacceptable. For masked Dilithium2, recovering the complete $\mathbf{s}_1$ using only SCAs would take approximately 600 hours.

TABLE XI: Time to recover masked Dilithium $\mathbf{s}_1$ (h)

Security level	SIS method	NTT_{256}	NTT_{128}	NTT_{64}
masked Dilithium2	LLL	270.6	179.7	123.6
	BKZ-40	191.4	134.8	89.9
masked Dilithium3	LLL	366.6	280.8	168.5
	BKZ-40	240.9	168.5	112.3
masked Dilithium5	LLL	513.2	314.5	235.9
	BKZ-40	335.0	216.3	157.3

Tab.XI shows the time overhead for recovering the complete $\mathbf{s}_1$ of masked Dilithium using SCAs and different dimensionality reduction strategies combined with LLL and BKZ-40. For attacks on masked implementations, the main time overhead is concentrated in the SCAs. Thus, after applying the SIS-assisted methods, the time overhead is significantly reduced. BKZ-40, which can solve the corresponding SIS problem with fewer coefficients, outperforms the LLL algorithm. Furthermore, the time for each algorithm is significantly reduced under the dimensionality reduction strategy. For masked Dilithium2, the time is reduced to 90 hours, improving efficiency by 6 times.

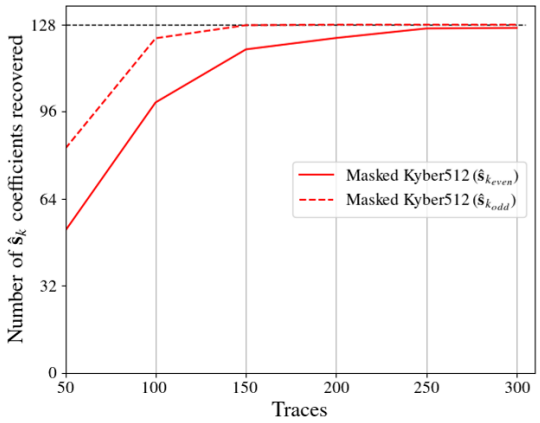


Fig. 15: Number of $\hat{\mathbf{s}}_k$ coefficients recovered of Masked Kyber512.

For Kyber, we chose the implementation without key refreshing to more comprehensively demonstrate our attack.

TABLE XII: Time to recover masked Kyber $\mathbf{s}_1$ (s)

Security level	SIS method	NTT_{128}	NTT_{64}
Kbyer512	LLL	489.5	359.3
	BKZ-40	430.1	220.9
Kbyer768	LLL	708.6	542.8
	BKZ-40	504.5	340.7
Kbyer1024	LLL	945.5	723.9
	BKZ-40	672.7	457.2

When the private key is not refreshed, as shown in Fig.15, the attacker only needs to recover two fixed mask values and then combine them to complete the attack. The implementation of Masked Kyber in [39] optimized polynomial multiplication and Montgomery reduction in assembly. A pair of coefficients now requires 3 Montgomery reductions instead of 5, making it necessary for the attacker to recover the correct even-position coefficients before the odd-position coefficients. Additionally, the optimized Montgomery reduction requires only two assembly instructions. Previous schemes that could recover all intermediate values of the lower bits are difficult to utilize. Therefore, our CPA scheme cannot be used to determine whether the coefficients are correctly recovered.

In the practical attack of first-order masked Kyber512, the entire attack process is systematically divided into four stages. 1) Employing SCAs to recover both $\hat{\mathbf{s}}^0_{k_{odd}}$ and $\hat{\mathbf{s}}^1_{k_{odd}}$; 2) Calculating $\mathbf{s}_{k_{odd}}$; 3) Utilizing SCAs to recover both $\hat{\mathbf{s}}^0_{k_{even}}$ and $\hat{\mathbf{s}}^1_{k_{even}}$; 4) Computing to determine $\mathbf{s}_{k_{even}}$. Fig.15 showcases the results of the SCA corresponding to stages 2) and 4). Overall, attacking coefficients in odd positions works better, requiring only 200 traces to recover all coefficients accurately. However, only 123 coefficients can be retrieved for the even positions at this juncture. For this attack, it is possible to keep increasing the number of traces until all coefficients can be recovered correctly, and this experiment achieves that goal at 500 traces in 3 minutes.

Regardless of whether key refreshing is used, side-channel analysis can recover the complete private keys of masked Dilithium and Kyber. The introduction of SIS-assisted methods significantly enhances the attack speed. Additionally, in practical attacks, it may not be possible to achieve a 100% success rate for side-channel attacks due to low signal-to-noise ratio or a limited number of signals collected. However, as long as there are only a few errors, as analyzed in III-C, the complete private key can be recovered by using the LLL and BKZ algorithms with repeated random selection of NTT domain coefficients.

VI. Conclusion and Future Work

In this work, we conducted a detailed analysis of the private key NTT process in the post-quantum algorithms Dilithium and Kyber, clarifying how to convert the problem of recovering the full private key from known partial NTT domain coefficients into an SIS problem. By further utilizing the properties of NTT for dimension reduction, we significantly decreased the difficulty of solving the problem and greatly increased the solving speed. With the newly proposed CPA scheme, we also conducted practical attacks, confirming that the dimension reduction strategy is more efficient and requires

fewer NTT domain coefficients, allowing us to recover the complete $\mathbf{s}_1$ of any security level of Dilithium within 6 minutes. We also successfully attacked first-order masking implementations, significantly reducing the required time with the help of the dimension reduction strategy.

Theoretically, the multi-dimensional nature of the lattice ensures the security of the lattice-based cryptosystem. With current computing power, it is almost impossible to brute force a private key through an exhaustive search. However, each coefficient is calculated independently when the algorithm is executed on a cryptographic platform. From an SCA perspective, this independence means that as the dimensionality increases, attackers will only face repeated challenges. A significant difficulty in performing SCA on lattice-based algorithms such as Dilithium is the requirement of recovering all the coefficients correctly, which is difficult to guarantee. However, due to the introduction of NTT operations to improve efficiency, the requirement on the SCA success rate can be greatly relaxed using SIS-assisted method.

In terms of protective strategies, the experimental results from masked Kyber have already highlighted the importance of key refreshment. Additionally, we suggest adopting a shuffle strategy for polynomial multiplication to counter our attacks; without being able to determine the correct positions of NTT domain coefficients, constructing the SIS problem becomes impossible. In the future, we will also explore how to solve the aforementioned problem when the correct positions of the coefficients are unknown.

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, 20-22 November 1994*. IEEE Computer Society, 1994, pp. 124–134. [Online]. Available: https://doi.org/10.1109/SFCS.1994.365700

[2] G. Alagic, D. Cooper, Q. Dang, T. Dang, J. M. Kelsey, J. Lichtinger, Y.-K. Liu, C. A. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, and D. Apon, "Status report on the third round of the nist post-quantum cryptography standardization process," 2022-07-05 04:07:00 2022. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=934458

[3] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, ser. Lecture Notes in Computer Science, M. J. Wiener, Ed., vol. 1666. Springer, 1999, pp. 388–397. [Online]. Available: https://doi.org/10.1007/3-540-48405-1_25

[4] P. Ravi, S. S. Roy, A. Chattopadhyay, and S. Bhasin, "Generic side-channel attacks on cca-secure lattice-based PKE and kems," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 3, pp. 307–335, 2020. [Online]. Available: https://doi.org/10.13154/tches.v2020.i3.307-335

[5] Z. Xu, O. Pemberton, S. S. Roy, D. F. Oswald, W. Yao, and Z. Zheng, "Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber," *IEEE Trans. Computers*, vol. 71, no. 9, pp. 2163–2176, 2022. [Online]. Available: https://doi.org/10.1109/TC.2021.3122997

[6] G. Rajendran, P. Ravi, J. D'Anvers, S. Bhasin, and A. Chattopadhyay, "Pushing the limits of generic side-channel attacks on lwe-based kems - parallel PC oracle attacks on kyber KEM and beyond," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 2, pp. 418–446, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i2.418-446

[7] Y. Tanaka, R. Ueno, K. Xagawa, A. Ito, J. Takahashi, and N. Homma, "Multiple-valued plaintext-checking side-channel attacks on post-quantum kems," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 3, pp. 473–503, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i3.473-503

[8] M. Shen, C. Cheng, X. Zhang, Q. Guo, and T. Jiang, "Find the bad apples: An efficient method for perfect key recovery under imperfect SCA oracles - A case study of kyber," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 1, pp. 89–112, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i1.89-112

[9] Y. Liu, Y. Zhou, S. Sun, T. Wang, R. Zhang, and J. Ming, "On the security of lattice-based fiat-shamir signatures in the presence of randomness leakage," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1868–1879, 2021. [Online]. Available: https://doi.org/10.1109/TIFS.2020.3045904

[10] Z. Qiao, Y. Liu, Y. Zhou, J. Ming, C. Jin, and H. Li, "Practical public template attack attacks on crystals-dilithium with randomness leakages," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 1–14, 2023. [Online]. Available: https://doi.org/10.1109/TIFS.2022.3215913

[11] S. Marzougui, V. Ulitzsch, M. Tibouchi, and J. Seifert, "Profiling side-channel attacks on dilithium: A small bit-fiddling leak breaks it all," *IACR Cryptol. ePrint Arch.*, p. 106, 2022. [Online]. Available: https://eprint.iacr.org/2022/106

[12] 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

[13] 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*, ser. Lecture Notes in Computer Science, W. Fischer and N. Homma, Eds., vol. 10529. Springer, 2017, pp. 513–533. [Online]. Available: https://doi.org/10.1007/978-3-319-66787-4_25

[14] M. Hamburg, J. Hermelink, R. Primas, S. Samardjiska, T. Schamberger, S. Streit, E. Strieder, and C. van Vredendaal, "Chosen ciphertext k-trace attacks on masked CCA2 secure kyber," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 4, pp. 88–113, 2021. [Online]. Available: https://doi.org/10.46586/tches.v2021.i4.88-113

[15] 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

[16] T. Tosun and E. Savas, "Zero-value filtering for accelerating non-profiled side-channel attack on incomplete ntt-based implementations of lattice-based cryptography," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 3353–3365, 2024. [Online]. Available: https://doi.org/10.1109/TIFS.2024.3359890

[17] A. Karlov and N. L. de Guertechin, "Power analysis attack on kyber," *IACR Cryptol. ePrint Arch.*, p. 1311, 2021. [Online]. Available: https://eprint.iacr.org/2021/1311

[18] 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.

[19] Y. Yang, Z. Wang, J. Ye, J. Fan, S. Chen, H. Li, X. Li, and Y. Cao, "Chosen ciphertext correlation power analysis on kyber," *Integr.*, vol. 91, pp. 10–22, 2023. [Online]. Available: https://doi.org/10.1016/j.vlsi.2023.02.012

[20] 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

[21] 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

[22] R. Mi, H. Jiang, and Z. Zhang, "Lattice reduction meets key-mismatch: New misuse attack on lattice-based nist candidate kems," Cryptology ePrint Archive, Paper 2022/1064, 2022, https://eprint.iacr.org/2022/1064. [Online]. Available: https://eprint.iacr.org/2022/1064

[23] E. Krahmer, P. Pessl, G. Land, and T. Güneysu, "Correction fault attacks on randomized crystals-dilithium," *IACR Cryptol. ePrint Arch.*, p. 138, 2024. [Online]. Available: https://eprint.iacr.org/2024/138

[24] M. R. Albrecht, A. Deo, and K. G. Paterson, "Cold boot attacks on ring and module LWE keys under the NTT," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 173–213, 2018. [Online]. Available: https://doi.org/10.13154/tches.v2018.i3.173-213

[25] M. Tunstall, N. Hanley, R. P. McEvoy, C. Whelan, C. C. Murphy, and W. P. Marnane, "Correlation power analysis of large word sizes," in *IETIrish Signals and Systems Conf - ISSC*, 2007, pp. 145–150.

[26] T. Pöppelmann and T. Güneysu, "Towards efficient arithmetic for lattice-based cryptography on reconfigurable hardware," in *Progress in Cryptology - LATINCRYPT 2012 - 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings*, ser. Lecture Notes in Computer Science, A. Hevia and G. Neven, Eds., vol. 7533. Springer, 2012, pp. 139–158. [Online]. Available: https://doi.org/10.1007/978-3-642-33481-8_8

[27] 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)," 2020.

[28] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, ser. Lecture Notes in Computer Science, M. Joye and J. Quisquater, Eds., vol. 3156. Springer, 2004, pp. 16–29. [Online]. Available: https://doi.org/10.1007/978-3-540-28632-5_2

[29] T. Le, J. Clédière, C. Canovas, B. Robisson, C. Servière, and J. Lacoume, "A proposition for correlation power analysis enhancement," in *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, ser. Lecture Notes in Computer Science, L. Goubin and M. Matsui, Eds., vol. 4249. Springer, 2006, pp. 174–186. [Online]. Available: https://doi.org/10.1007/11894063_14

[30] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, and V. Verneuil, "Improved collision-correlation power analysis on first order protected AES," in *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer, 2011, pp. 49–62. [Online]. Available: https://doi.org/10.1007/978-3-642-23951-9_4

[31] M. Ajtai, "Generating hard instances of lattice problems," *Electron. Colloquium Comput. Complex.*, vol. TR96, 1996.

[32] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds., vol. 7237. Springer, 2012, pp. 738–755. [Online]. Available: https://doi.org/10.1007/978-3-642-29011-4_43

[33] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, "New directions in nearest neighbor searching with applications to lattice sieving," in *IACR Cryptology ePrint Archive*, 2016.

[34] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, "Revisiting the expected cost of solving usvp and applications to LWE," in *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, ser. Lecture Notes in Computer Science, T. Takagi and T. Peyrin, Eds., vol. 10624. Springer, 2017, pp. 297–322. [Online]. Available: https://doi.org/10.1007/978-3-319-70694-8_11

[35] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange - A new hope," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, T. Holz and S. Savage, Eds. USENIX Association, 2016, pp. 327–343. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

[36] 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

[37] J. Coron, F. Gérard, M. Trannoy, and R. Zeitoun, "Improved gadgets for the high-order masking of dilithium," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 4, pp. 110–145, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i4.110-145

[38] J. W. Bos, M. Gourjon, J. Renes, T. Schneider, and C. van Vredendaal, "Masking kyber: First- and higher-order implementations," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 4, pp. 173–214, 2021. [Online]. Available: https://doi.org/10.46586/tches.v2021.i4.173-214

[39] D. Heinz, M. J. Kannwischer, G. Land, T. Pöppelmann, P. Schwabe, and A. Sprenkels, "First-order masked kyber on ARM cortex-m4," *IACR Cryptol. ePrint Arch.*, p. 58, 2022. [Online]. Available: https://eprint.iacr.org/2022/058