Single Trace is All It Takes: Efficient Side-channel Attack on Dilithium

Zehua Qiao[1,2], Yuejun Liu[3], Yongbin Zhou[1,3], Yuhan Zhao[3] and Shuyi Chen[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{qiaozehua}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
liuyuejun@njust.edu.cn

Abstract.
As the National Institute of Standards and Technology (NIST) concludes its post-quantum cryptography (PQC) competition, the winning algorithm, Dilithium, enters the deployment phase in 2024. This phase underscores the importance of conducting thorough practical security evaluations. Our study offers an in-depth side-channel analysis of Dilithium, showcasing the ability to recover the complete private key, $\boldsymbol{s}_1$, within ten minutes using just two signatures and achieving a 60% success rate with a single signature. We focus on analyzing the polynomial addition in Dilithium, $z = y + \boldsymbol{cs}_1$, by breaking down the attack into two main phases: the recovery of y and $\boldsymbol{cs}_1$ through side-channel attacks, followed by the resolution of a system of error-prone equations related to $\boldsymbol{cs}_1$. Employing Linear Regression-based profiled attacks enables the successful recovery of the full y value with a 40% success rate without the necessity for initial filtering. The extraction of $\boldsymbol{cs}_1$ is further improved using a CNN model, which boasts an average success rate of 75%. A significant innovation of our research is the development of a constrained optimization-based residual analysis technique. This method efficiently recovers $\boldsymbol{s}_1$ from a large set of error-containing equations concerning $\boldsymbol{cs}_1$, proving effective even when only 10% of the equations are accurate. We conduct a practical attack on the Dilithium2 implementation on an STM32F4 platform, demonstrating that typically two signatures are sufficient for complete private key recovery, with a single signature sufficing in optimal conditions. Using a general-purpose PC, the full private key can be reconstructed in ten minutes.

Keywords: Lattice-based Cryptography · CNN · Side-channel Attacks · Dilithium

1 Introduction

Rapid advancements in quantum computing present a significant threat to cryptographic algorithms that rely on the computational difficulty of problems such as integer factorization and discrete logarithms. Should a general-purpose quantum computer be successfully developed, it is expected that the quantum algorithm proposed by Shor [Sho94] in 1994 would render these cryptographic algorithms vulnerable to being broken in polynomial time. In response, the National Institute of Standards and Technology (NIST) has initiated the PQC competition, which has led to the identification of CRYSTALS-Dilithium (abbr.Dilithium) as a digital signature candidate.

Dilithium [DKL+18] is a digital signature scheme based on the hardness of lattice problems, utilizing the Fiat-Shamir paradigm within the polynomial ring $Z_{q[x]}/(x^n+1)$. Its

Parts of this work was submitted to Fincryptography at 15:00pm Mar 30, 2024 AOE.

recognition among experts is due to its comprehensive performance that marries operational efficiency with theoretical safety. This balance positions Dilithium as a prominent candidate in the field of post-quantum cryptography, selected for its capability to secure digital communications against the quantum computing threat.

Despite the theoretical resilience of PQC algorithms against both quantum and classical computational attacks, their real-world implementations remain vulnerable to side-channel analysis. This form of attack exploits unintentional leakages from cryptographic operations, such as power consumption [KJJ99], electromagnetic emissions [QS01], and execution timing [Koc96], to extract sensitive information. Over nearly three decades, side-channel analysis has matured significantly within the field of cryptanalysis, marking substantial achievements. It has successfully facilitated the practical analysis of cryptographic algorithms, including those targeting DES [KJJ99], AES [BCO04], RSA [BJL+14], and ECC [Ols04], among others.

To date, numerous studies have been conducted to evaluate the security of Dilithium implementations, yielding significant findings. In the realm of side-channel analysis, particularly the non-template class exemplified by Correlation Power Analysis (CPA), such investigations have predominantly focused on the polynomial multiplication operation $\boldsymbol{cs}_1$ for conducting attacks. Chen *et al.* [BCO04], building upon the methodology proposed by Fournaris *et al.* [FDK20] for the Montgomery reduction operation, have demonstrated that the CPA technique, aimed at the exhaustive recovery of the private key from 157 traces, can achieve this in 6,357 seconds for one Number Theoretic Transform (NTT) domain coefficient. Furthermore, the integration of a partitioning method has been shown to accelerate the attack by a factor of 7.77. Qiao *et al.* have developed a CPA-based attack, enhanced with the LLL algorithm, to accomplish full Dilithium private key recovery in less than one minute. Additionally, Liu *et al.* [QLZ+23b] have introduced an innovative random leakage attack strategy, leveraging public template attacks to extract lower-bit polynomial coefficient information. This approach significantly streamlines the private key recovery process, reducing it to a solvable integer LWE problem within polynomial time.

The exploration of profiling side-channel attacks, especially those integrating machine learning methodologies, has provided significant advancements. Han *et al.* [HLK+21] initiated the recovery of all Dilithium private keys by focusing on the NTT's initial butterfly operations during the signature generation phase, employing a machine learning-based template attack. Following this, Marzougui *et al.* [MUTS22] devised an attack targeting the sensitive random number y, correlating sensitive parameters with specific values (e.g., $y = 0$) and constructing a system of linear equations about $\boldsymbol{cs}_1$ to resolve the Dilithium private key $\boldsymbol{s}_1$. Berzati *et al.* [BVC+23] honed in on a narrower spectrum of sensitive intermediate values, particularly $\boldsymbol{w}_0$, using a filtering algorithm specifically designed for Dilithium's parameter traits. Wang *et al.* [WNGD23] launched an attack on the secret key unpacking phase of the signing algorithm, leveraging deep learning-assisted profiled power analysis. This approach harbors a slim chance of completing the private key recovery with a single trace, boasting a success rate nearing 100% after 74 signatures. A common challenge among several of these works is the task of solving a system of equations concerning the error-prone $\boldsymbol{cs}_1$, predominantly utilizing integer linear programming (ILP) for this purpose. Bronchain *et al.* [BAE+23] applied the Belief Propagation (BP) algorithm to solve for polynomial multiplication $\boldsymbol{cs}_1$ and conducted simulation experiments to attack y. Their optimal finding indicates that recovering the private key $\boldsymbol{s}_1$ can be achieved with four signatures using the Hamming model at a signal-to-noise ratio (SNR) of 700.

In profiling attacks, particularly those aimed at values such as y and $\boldsymbol{w}_0$, the fundamental strategy is to choose specific numbers to enhance the success rate of side-channel attacks. This approach, however, requires a substantial number of power traces for both the construction and matching of templates due to the large candidate space of the above targets. Moreover, given that these target values are regenerated randomly with each

signature, the attack typically hinges on a single trace, which frequently fails to secure a high success rate. When faced with numerous errors, the application of lattice basis reduction algorithms presents its own set of challenges, and alternative approaches, such as integer ILP, demand considerable time. Bronchain *et al.* [BAE+23] suggested the use of BP algorithm to resolve the system of error-prone equations in Dilithium related to $\boldsymbol{cs}_1$, though their investigations were confined to simulation experiments. Furthermore, the low accuracy of side-channel information necessitates an increased number of equations, leading to an excessively large graph for the BP algorithm and significantly elevating the risk of computational overflow. Due to the need to determine the probability distribution of all candidate values for the target value, the feasibility of the BP algorithm in attacks such as fault injection, where a probability distribution is not available, remains to be determined.

This paper introduces a novel method for swiftly recovering the Dilithium private key with a reduced number of signatures. We propose an approach for the side-channel attack aspect that enables the complete recovery of the random number y with a high success rate, eliminating the need for any filtering. Additionally, we explore an attack on $\boldsymbol{cs}_1$ targeting a significantly narrowed value space. Moreover, we unveil a constrained optimization-based residual analysis technique tailored for efficiently solving error-prone linear equations associated with $\boldsymbol{cs}_1$. These methodologies were deployed in a practical attack scenario against the open-source implementation of Dilithium2, conducted on an STM32F4 platform, which substantiates the feasibility and efficacy of our proposed techniques. The specific contributions of our research are as follows:

- Building upon Qiao *et al.*'s [QLZ+23a] discovery that higher bit information of y can be deduced from the signature z, we introduce a Linear Regression-based side-channel attack. This method effectively mitigates the influence of higher bits, thereby enhancing the attack's success rate. In practical experiments, this approach achieves a 40% success rate in attacking y.

- We exploit the characteristic that $\boldsymbol{cs}_1$ operates within a limited value range, with each coefficient being independently calculated. For Dilithium2, a single signature unveils a minimum of 1024 data leaks, which are conducive to machine learning pre-training. Employing a Convolutional Neural Networks (CNNs) model that accounts for potential alignment discrepancies, we attain a 75% success rate in completely recovering the value of $\boldsymbol{cs}_1$.

- Given that z is known in $z = y + \boldsymbol{cs}_1$, we utilize the relatively more precise Hamming weight information from side-channel results for y to augment the success rate of $\boldsymbol{cs}_1$ recovery. By amalgamating the attack outcomes on y with those on $\boldsymbol{cs}_1$, we elevate our recovery success rate of $\boldsymbol{cs}_1$ to 92%, marking a significant leap in attack efficiency.

- To tackle the issue of erroneous equations in the recovery of $\boldsymbol{cs}_1$, we introduce a constrained optimization-based residual analysis. This innovative approach rapidly solves the system of integer linear equations laden with errors, leveraging the constraints inherent in the private key $\boldsymbol{s}_1$. Notably, this technique proves effective even with only 10% accuracy in the system of equations, provided that a sufficient number of equations are present. This substantially narrows down the private key value space and expedites the ILP process.

- Our practical assault on the Dilithium2 reference implementation on an STM32F4 platform is demonstrated across three scenarios: access to only y, only $\boldsymbol{cs}_1$, and both y and $\boldsymbol{cs}_1$. The outcomes in the worst-case scenarios illustrate that targeting y enables private key recovery within 2 minutes using 8 signatures; attacking $\boldsymbol{cs}_1$

necessitates 3 signatures; and harnessing leaks from both y and $\boldsymbol{cs}_1$ facilitates private key recovery in 3 minutes with merely 2 signatures. In these tests, a 60% probability of recovering the private key with a single signature was observed.

2 Preliminaries

2.1 Dilithium

Dilithium emerges as a prominent digital signature scheme, founded on the principles of the Module Learning with Errors (MLWE) and Module Short Integer Solution (MSIS) problems. This intricate foundation affords Dilithium the flexibility to offer varied security levels, accommodating a broad spectrum of application scenarios. This adaptability ensures that Dilithium can meet the diverse performance and security expectations across different devices. Tab.1 delineates the parameter configurations for each security level.

Table 1: Dilithium parameters at different NIST security levels

NIST Security Level	2	3	5
d [dropped bits from $\boldsymbol{t}$]	13		
τ[# of non-zero coefficients in $\boldsymbol{c}$]	39	49	60
γ_1 [cofficient range of $\boldsymbol{y}$]	131,072	524,288	
γ_2 [low-order rounding range]	95,232	261,888	
$(m \times n)$ [dimensions of $\mathbf{A}$]	(4,4)	(6,5)	(8,7)
η [private key range]	2	4	2
β $[\tau \cdot \eta]$	78	196	120

Dilithium operates within the cyclotomic ring $\mathbb{R}_q^n$, where each coefficient is defined in the finite field $\mathbb{Z}_q$. The constants $q = 8380417$ and $n = 256$ are fixed across all security levels, ensuring a uniform foundation for operations. The algorithm delineates three fundamental procedures: key generation, the signing process, and signature verification. Our investigation primarily focuses on the signing process.

The signature process of Dilithium, as outlined in Alg.1, commences with the input of a secret key sk and a message M. Initially, the algorithm expands the secret key ρ to construct a structured matrix A within $R_q^{m\times n}$ using the ExpandA function, followed by generating a 384-bit string μ from the transaction identifier tr and message M through the Cryptographic Hash Function (CRH). It initializes κ and sets (z, h) to null, then creates a 384-bit string ρ either by hashing κ concatenated with μ or by selecting randomly. Subsequently, the NTT is applied to both A and the secret $\boldsymbol{s}_1$, generating $\hat{\mathbf{A}}$ and $\hat{\mathbf{s}_1}$. A masking vector$y$ is derived from ρ' and κ, within the set S_{γ_1-1}. The algorithm computes w by multiplying $\hat{\boldsymbol{A}}$ with the NTT of y and applying the Inverse NTT (INTT), then extracts high-order bits from w to form a challenge vector $\tilde{\boldsymbol{c}}$, integral to the signature's validity. The algorithm features a rejection sampling loop to ensure the generated vectors $\boldsymbol{z}$ and $\boldsymbol{r}_0$ meet specific security criteria. If the criteria are not met, the algorithm recalibrates $\boldsymbol{y}$ and iterates again, ensuring compliance with Dilithium's security standards. Upon meeting these standards, the algorithm produces a signature comprising $\boldsymbol{z}$, a hint vector $\boldsymbol{h}$ for verification, and $\tilde{\boldsymbol{c}}$. This process demonstrates the Dilithium scheme's commitment to security and efficiency, making it suitable for various post-quantum cryptographic applications.

2.2 Linear Regression-based Profiled Attacks

In cryptanalysis, Linear Regression (LR)-based profiled attacks represent an evolution over traditional template attacks by adopting a sophisticated leakage model that transcends

Algorithm 1 Dilithium Sign(sk,M)

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

the limitations of conventional methodologies. In cryptanalysis, LR-based profiled attacks represent an evolution over traditional template attacks by adopting a sophisticated leakage model that transcends the limitations of conventional methodologies. Schindler *et al.* [SLP05] introduced LR-based profiled attacks, marking a significant shift from the traditional HW leakage model. This new approach recognizes that the leakage weights of different bits can vary, a detail that linear regression captures with precision.

The power consumption model, defined through linear regression, is expressed as:

$$m(y) = \sum_{i=0}^{l_y} a_i \rho(y_i) + a_{l_y+1}$$

where y is the target data, y_i is the i-th bit from least to most significant, l_y is the bit length of y, a_i are coefficients reflecting the leakage weight of each bit, and $\rho(y_i)$ is a mapping function that ensures the influence of $y_i = 0$ is considered in the model. This mapping is particularly chosen to maintain the impact of each bit value on the model accurately.

$$\rho(y_i) = \begin{cases} 1 & \text{if } y_i = 1 \\ -1 & \text{if } y_i = 0 \end{cases}$$

In the modeling phase, actual power consumption curves, L, facilitate the determination of the coefficients a_i via a linear least squares method. The derived $m(y)$ functions as the mean template in traditional template attacks, necessitating an additional step to compute the covariance matrix Σ for template creation. During the attack phase, given an observed power consumption curve L, the probability density function is outlined as:

$$f[L|Y = y] = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(L - m(y))^T \Sigma^{-1} (L - m(y))\right)$$

where k is the dimension of L. Utilizing Bayes' theorem, this is reformulated into the desired probability $f[Y = y|L]$, illustrated as:

$$f[Y = y|L] = \frac{f(L|y)p(y)}{\sum_{y'} f(L|y')p(y')}$$

assuming equal probability for each y, simplifying the expression to:

$$f[Y = y|L] = \frac{f(L|y)}{\sum_{y'} f(L|y')}$$

This advanced LR-based profiled attacks methodology signifies a transition towards models that accurately reflect the nuanced reality of bit-level leakage differences in real devices, thereby enhancing the precision and efficacy of cryptanalytic endeavors in digital security.

2.3 CNN-based Template Attacks

The deployment of CNN in SCA is predicated on the assembly of a comprehensive training dataset, comprising power traces, plaintext-ciphertext pairings, and corresponding cryptographic keys. Each trace is meticulously labeled with sensitive intermediate values, thus categorizing the data and furnishing the CNN with supervisory signals for the duration of the training phase. This structured compilation of labeled energy traces forms the bedrock for the CNN's learning process, equipping it to discern the subtle nuances of cryptographic operations.

CNNs are distinguished by their complex architecture that includes convolutional operations. Their application in side-channel analysis has been proven effective, thanks to a layered structure comprising convolutional layers for initial feature extraction, pooling layers for reducing dimensionality, and fully connected layers for classification. Convolutional layers employ a set of filters—each with unique weights and biases—to conduct convolution operations on the input data. This process effectively captures and highlights essential patterns. Pooling layers simplify the feature set by summarizing data within specific input regions, applying max and average pooling techniques to preserve vital information efficiently. Fully connected layers integrate these refined features to produce the final output classifications. The strategic placement of batch normalization layers between select convolutional and pooling stages significantly boosts the network's efficiency and stability during training by standardizing the inputs to each layer.

The architecture of a CNN encapsulates a holistic strategy for feature extraction and classification, making it uniquely adept at detecting patterns that signal cryptographic vulnerabilities within side-channel attack data. Its mathematical formulation can be succinctly represented as [ZBHV20]:

$$g(x) = s \circ \lambda^{n_3} \circ \delta \circ (\gamma \circ \alpha \circ \gamma)^{n_1} \circ \delta^{n_2} \circ \gamma = \hat{y}.$$

Here γ, α, δ, λ, and s represent convolutional layers, activation functions, pooling layers, fully connected layers, and the activation function of the output layer, respectively. The variables n_1, n_2, and n_3 indicate the respective counts of these computational components, illustrating the CNN's structural depth and complexity.

The training phase of the CNN is pivotal, as it trains the model to accurately discern patterns in power trace data. Once trained, the CNN can effectively predict sensitive intermediate values from previously unseen traces, facilitating a cryptanalytic approach that is both highly efficient and effective.

3 Side-channel Attacks Against Dilithium

In addressing the polynomial addition process, denoted as $z = y + \boldsymbol{cs}_1$, we can utilize the availability of the known signature z to indirectly derive $\boldsymbol{cs}_1$ by first recovering y through

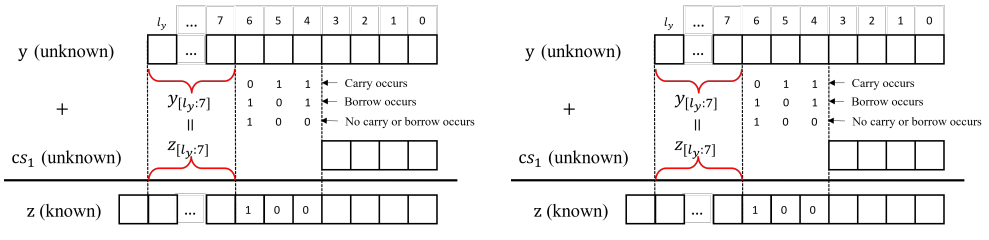
a side-channel attack. Alternatively, $\boldsymbol{cs}_1$ can be directly recovered via side-channel vectors. By combining insights from the side-channel attack aimed at y with the analysis targeting $\boldsymbol{cs}_1$, we anticipate a considerable enhancement in the efficacy of our cryptographic analysis. The methodologies for conducting side-channel attacks on both y and $\boldsymbol{cs}_1$ will be elaborated in the forthcoming sections.

3.1 Attacking y with LR-Based Side-channel Attack

For the mask polynomial y, which operates within a value range of $q = 8380417$, and given that each signature y is randomly generated, we are limited to using only one trace to accomplish the recovery of y in the practical attack. Direct application of the conventional template attack, under the identity model, is theoretically viable but practically challenging due to the extensive dataset required for accurate modeling and the correspondingly low success rate.

In practical applications, concerning polynomial addition $z = y + \boldsymbol{cs}_1$, where $\boldsymbol{cs}_1$ lies within the interval $(-\beta, \beta)$ and and approximates a Gaussian distribution, the magnitude of β is contingent on the security level, show in Tab.1. Notably, for Dilithium3—the scheme's most secure variant—β remains below 198. This realization obviates the need to enumerate all conceivable y values when conducting a template attack aimed at y's recovery. With the known signature message z, it becomes evident that viable y values, pertinent to the trace targeted in the attack, are effectively constrained to the interval $(z - ||\boldsymbol{cs}_1||_\infty, z + ||\boldsymbol{cs}_1||_\infty)$. This constraint significantly enhances the attack's efficiency.

Liu *et al.* [LZS+21, QLZ+23a] have illuminated the process of deducing high-bit information of y when access to z is available, especially under the condition that $\boldsymbol{cs}1$ is substantially smaller than y. In the Dilithium signature process, the signature z and the random number y can each assume one of 2^{18} possible values. The intermediate value $\boldsymbol{cs}_1$ typically assumes values significantly smaller than both z and y. By judiciously leveraging the arithmetic property where a larger number is added to a smaller one, it is possible to directly deduce partial information of the random number y. Fig.1 illustrates this concept. Under the assumption that $||\boldsymbol{cs}_1||_\infty < 2^4$, and when $z_{[i-1:\tau]} = 10\ldots00_2$ or $z_{[i-1:\tau]} = 01\ldots11_2$, we identify three scenarios enabling the attacker to derive a segment of y (specifically $y_{[l_y:i]}$) through its addition with $\boldsymbol{cs}1$ to yield $z_{[i-1=6:\tau=4]} = 100_2$. These scenarios encompass instances where the addition involves rounding, borrowing, or neither rounding nor borrowing. Importantly, none of these cases impact y_7 or higher bits, leading to the conclusion that $y_{[l_y:i]} = z_{[l_y:i]}$. This analysis demonstrates that in many instances, the high-bit information leakage of y can effectively be disregarded during a template attack, thereby substantially enhancing the attack's success rate.



(a) $z_{[i-1=6:\tau=4]} = 100_2$ (b) $z_{[i-1=6:\tau=4]} = 011_2$

Figure 1: Example of $y_{[l_y:i]} = z_{[l_y:i]} (i = 7, \tau = 4)$

LR-based profiled attacks offer a more efficient approach to modeling, requiring fewer training samples to recover the entire target value beyond merely the Hamming weight. This method enhances template accuracy by mapping leakage features for each bit of

the target value. During template matching, the process iterates through all possible y_i values, comparing the theoretical leakage predicted by the LR model against actual trace observations. Crucially, bits within different y_i values that remain constant do not interfere with the matching process, making this approach particularly effective for scenarios involving polynomial addition $z = y + \boldsymbol{cs}_1$ in the context of the Dilithium cryptographic algorithm.

The precise methodology for executing an attack on the y within the Dilithium framework is delineated in the following steps:

(1) Templates building: Utilize the captured traces along with their corresponding labels to carve out the leakage for each bit of the y. Subsequently, a LR model is constructed to encapsulate this data.

(2) Trace capturing: Capture traces pertinent to y from the target device.

(3) Template Matching: For every captured trace, iterate over all possible $\boldsymbol{cs}_1$ values. By integrating these with the known signature z, deduce all feasible y values. The precise y value is then ascertained through the process of template matching, comparing the theoretical and observed leakage to pinpoint the exact match.

The Gaussian-like distribution of $\boldsymbol{cs}_1$ offers a strategic advantage in enhancing the success rate of attacks by allowing for the exclusion of less probable cases. This principle is particularly applicable across various Dilithium security levels, with the value range of $\boldsymbol{cs}_1$ detailed in Tab.2.

Table 2: Probability of $||\mathbf{cs_1}||_\infty < 2^\tau$ in Dilithium.

Security Level	τ=3	τ=4	τ=5	τ=6	τ=7
Dilithium2	0.57	0.95	0.99	1	1
Dilithium3	0.36	0.64	0.93	0.99	1
Dilithium5	0.65	0.86	0.99	1	1

Taking Dilithium2 as a case study, approximately 95% of the data falls within $||\boldsymbol{cs}_1||_\infty < 2^4$, while the remaining 5% of the cases, as outlined in Tab.2, span a wider value range of $(\pm 16, \pm 78)$. These outliers represent a significantly larger value space but are much less likely to occur. Including all potential candidate values in the attack could, paradoxically, lower the success rate. By strategically disregarding these less probable outliers, both the success rate and efficiency of the attack can be substantially improved.

3.2 Attacking cs_1 with CNN-Based Side-channel Attack

Beyond targeting the random number y in side-channel attacks, directly attempting to recover the $\boldsymbol{cs}_1$ value presents a compelling alternative. Owing to its smaller value space, $\boldsymbol{cs}_1$ inherently offers more favorable conditions for successful exploitation. The attack scenario is similar to y in that only one $\boldsymbol{cs}_1$ factor can be recovered using a single trace, underscoring the pivotal role of template quality in determining the attack's success rate.

Lattice-based cryptographic schemes, exemplified by Dilithium, inherently feature a high-dimensional landscape characterized by numerous repetitive and independent operations during polynomial processing. Taking the Dilithium2 as a case in point, the generation of a valid signature necessitates, on average, four instances of rejection sampling. This translates to approximately $256 \times 4 \times 5$ relevant operations for $\boldsymbol{cs}_1$. Considering the contributions of $\boldsymbol{cs}_2$, a single legitimate signature, in a deterministic implementation, can yield an average of 10,240 samples. Even in stochastic implementations, a substantial 2048 samples can be amassed. This abundance of data aligns well with the prerequisites of deep learning, which necessitates a substantial dataset for pre-training to effectively construct a distinguisher.

In the process of model selection, it is crucial to account for the fact that numerous samples from each signature are generated at distinct moments. This necessitates the use

of alignment operations to isolate leakage features, during which offsets are an unavoidable consequence. Deep learning methodologies, particularly those utilizing CNN, have proven to be exceptionally adept at handling data characterized by such offsets. This effectiveness is largely attributable to the architecture of CNNs, which incorporates convolutional layers and pooling layers. These layers are specifically designed to extract and distill features from data, even when it exhibits variability in alignment. Consequently, CNN-based side-channel analysis methods stand out for their capacity to perform robust feature extraction across datasets with inherent offsets.

In this study, we implement a CNN model following the approach outlined by Zaid *et al.* in [ZBHV20], which is designed with a three-layer convolutional structure. This model adeptly handles complex side-channel datasets, notably ASCAD, incorporating countermeasures like Random Delay and first-order masking, achieving a Guessing Entropy (GE) of 1 with minimal traces—244 for $N^{[0]} = 50$ and 270 for $N^{[0]} = 100$.

Drawing inspiration from Zaid *et al.*'s methodology, our implementation utilizes a comparable three-layer CNN architecture. This structure is composed of convolutional layers interspersed with pooling and batch normalization operations. The design facilitates the gradual recognition of features, ranging from simple to intricate, and culminates in a dense layer equipped with a softmax activation function for precise classification of the processed inputs. The core attributes of our CNN architecture are delineated in Tab.3 below.

Table 3: Our CNN Architecture

Layers	*kernel_size*	*pooling_stride*	*number_filter*
1^{st} Convolutional Block	64	2	12
2^{nd} Convolutional Block	128	4	24
3^{rd} Convolutional Block	512	4	48

Leveraging a convolutional approach, our model excels in identifying and learning intricate patterns present in side-channel traces. Its architectural design facilitates a progressive extraction and condensation of features, ensuring an efficient representation of pertinent information. This capability not only enhances robust classification performance but also preserves computational efficiency.

3.3 Enhancing Success Rates through Integrated Results of y and cs_1

From the perspective of side-channel analysis, recovering the full value directly on the ARM platform using a single power trace presents significant challenges. However, the task of ascertaining the Hamming weight of variables like y and $\boldsymbol{cs}_1$ proves to be considerably more viable. Notably, with exclusive access to the Hamming weight information of y and $\boldsymbol{cs}_1$, and the utilization of the signature z, it is feasible to infer $\boldsymbol{cs}_1$ under specific conditions. By adeptly combining the insights gained from attacks aimed at these pivotal intermediate values, one can significantly boost the success rate of such analytical endeavors.

In practical analysis, for targets y and $\boldsymbol{cs}_1$, we normalize the corresponding sets of probability vectors derived from the side-channel analysis. In this context, the probability values aligned with the correct Hamming weight of the candidate values are elevated, while the probabilities for all other candidates are negligible. This allows for the application of a dot product calculation to yield the final result.

Contrastingly, Dilithium, unlike traditional cryptographic algorithms such as ECC and RSA, encompasses a more intricate computational process. This complexity introduces numerous sensitive values throughout the computation, which can be exploited via side-channel analysis, thereby elevating the security risk. By integrating attack results across various sensitive values, it's feasible to compromise the security of the algorithm at minimal cost, potentially even facilitating the recovery of the private key.

4 Resolving Equations with Errors in cs_1 for Dilithium

Upon retrieving $\boldsymbol{cs}_1$ via side-channel or fault attacks, a conventional step involves formulating integer linear equation sets with the known challenge ciphertext c to solve for $\boldsymbol{s}_1$. It is a fundamental challenge that, regardless of the method employed, achieving 100% accuracy rate in the determination of $\boldsymbol{cs}_1$ is nearly impossible. Traditionally, the Big-M method has been utilized to transform these challenges into integer ILP problems. However, this approach becomes prohibitively time-consuming with an increase in the number of errors. In response, this chapter introduces the Constrained Optimization-Based Residual Analysis, a novel methodology crafted to expedite the resolution of integer linear equation sets laden with a significant number of errors.

4.1 Constrained Optimization-Based Residual Analysis

Let's briefly introduce the calculation process of $\boldsymbol{cs}_1$. Given the challenge c, a 256-dimensional vector predominantly composed of zeros and including elements -1, 0, and 1, the polynomial multiplication $\boldsymbol{cs}_1$ can be depicted via matrix multiplication. In one signature, the transformation to the coefficient matrix $\boldsymbol{C}$ from the challenge vector $\boldsymbol{c} = (c_0, c_1, \ldots, c_{255})$ is achieved through finite field cyclotomic transformations. The specific calculations are as follow:

$$\begin{bmatrix} c_0 & -c_{n-1} & -c_{n-2} & \cdots & -c_2 & -c_1 \\ c_1 & c_0 & -c_{n-1} & \cdots & c_3 & -c_2 \\ c_2 & c_1 & c_0 & \cdots & -c_4 & -c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-2} & c_{n-3} & c_{n-4} & \vdots & c_0 & -c_{n-1} \\ c_{n-1} & c_{n-2} & c_{n-3} & \cdots & c_1 & c_0 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{bmatrix} = \boldsymbol{Cs}$$

Given that specific values of the challenge ciphertext c eliminate modular operations throughout the computation process, we can employ methods in the real domain to attempt solving this problem. In the context of solving integer linear equation sets with a significant number of erroneous equations, direct resolution by any means cannot eliminate the interference caused by these erroneous equations. The core idea of our method is to filter out and remove these erroneous equations, retaining the correct ones to achieve private key recovery.

Assuming an attacker achieves a 30% success rate in obtaining $\boldsymbol{cs}_1$, theoretically, ten signatures would yield 2560 equations pertaining to a set of $\boldsymbol{s}_1$ coefficients, out of which 768 are correct. Should a mechanism be devised that enables the selection of 256 correct equations from these, the attacker could then proceed to recover the authentic private key by resolving the most straightforward equation set. It is important to note that the coefficients of the Dilithium private key, depending on the security level, are limited to either 5 or 9 distinct integer values. This limitation not only aids the attacker in conducting a preliminary evaluation of the attack's outcome but also serves as a critical constraint in the process of solving the equations, a consideration that is often overlooked.

The pivotal process of filtering erroneous equations is bifurcated into two distinct phases: initially, the constrained equation system is solved to derive an interim solution, $\boldsymbol{s}_1^{'}$. This challenge can be reformulated as a large-scale bound-constrained minimization problem. Branch *et al.* [BCL99] introduced a subspace adaptation of the Coleman-Li trust region and interior method for this purpose. The implementation of this method can utilize either sparse Cholesky factorization or conjugate gradient computation for efficiency. Subsequently, $\boldsymbol{s}_1^{'}$ is substituted back into the equation system to calculate the residuals; equations yielding larger residuals are deemed more likely to be erroneous. Building on this foundational concept, we introduce Constrained Optimization-Based Residual Analysis, a

methodology designed to swiftly isolate correct equations within a vast array of error-laden equations, thereby facilitating the accurate computation of the private key.

Algorithm 2 Constrained Optimization-Based Residual Analysis

Input: $\boldsymbol{C}, \boldsymbol{sca_r}, bounds, e_{th}, d_{num}, r_{num}, s_{num}$
Output: $\boldsymbol{s}_1$

$available\_ind \leftarrow$ InitializeIndices($\boldsymbol{C}$)
$err\_weights \leftarrow$ **zeros**($|\boldsymbol{C}|$)
while $|available\_ind| > r_{num}$ **do**
  $s\_ind \leftarrow$ RandomSample($available\_ind, s_{num}$)
  $\boldsymbol{s}_1 \leftarrow$ SolveLSQ($\boldsymbol{C}[s\_ind], \boldsymbol{sca_r}[s\_ind], bounds$)
  $residuals \leftarrow$ CalculateResiduals($\boldsymbol{C}[s\_ind], \boldsymbol{sca_r}[s\_ind], \boldsymbol{s}_1$)
  $err\_weights \leftarrow$ UpdateWeights($residuals, err\_weights, d_{num}$)
  $available\_ind \leftarrow$ UpdateIndices($err\_weights, e_{th}$)
end while
if $\boldsymbol{C}[available\_ind]\boldsymbol{s}_1 == \boldsymbol{sca_r}[available\_ind]$ **then**
  return $\boldsymbol{s}_1$
end if

Alg.2 delineates the pseudo-code for Constrained Optimization-Based Residual Analysis. The inputs include the coefficient matrix $\boldsymbol{C}$ from the equation set generated by the vector c, the side-channel attack result sca_r, solution $bounds$, error threshold e_{th}, increment of error weights per iteration d_{num}, the retention count of the smallest equations r_{num}, and the selection count of equations per iteration s_{num}. Initially, the algorithm assigns zero as the error weight for each equation and initializes the equation set index.

The essence of the algorithm unfolds through an iterative procedure. In each iteration, a subset of equations is randomly selected. Utilizing these equations' coefficients and the side-channel attack outcomes, a constrained optimization solution method is employed to derive an approximate solution. Subsequently, residuals for this solution are computed, and error weights for each equation are updated according to the residuals' magnitude. These weights indicate the likelihood of error presence in the equations; those with higher weights are more prone to exclusion in future iterations. The iterative process persists until the count of retained equations meets a predefined minimum threshold. A solution is deemed reliable when the filtered equation set exhibits zero residuals with the current solution, indicating correct recovery from the error-containing equation set.

Key parameters such as the error threshold e_{th} and the number of equations d_{num} for iterative error weight incrementation are crucial for balancing attack efficiency and stability. The iteration's random equation selection count s_{num} introduces necessary randomness for processing solutions that cannot be correctly recovered otherwise. With an adequate number of correct equations within the set, our Constrained Optimization-Based Residual Analysis algorithm is typically able to recover the complete private key swiftly, leveraging constraints to enhance solution accuracy and recovery speed.

Typically, resolving a system of equations with 256 unknowns necessitates an equivalent number of equations to uniquely determine the solution. However, the distinct distribution of the coefficient matrix combined with constraints on $\boldsymbol{s}_1$ coefficients uniquely positions us to recover the complete private key by accurately identifying a subset of correct equations. This observation enables the application of ILP to a reduced set of 200 equations for Dilithium2. Leveraging this insight, an attacker could feasibly achieve full private key recovery utilizing merely a single signature.

4.2 Big-M with Constrained Optimization-Based Residual Analysis

The problem we face is actually to find a solution $\boldsymbol{s}_1^*$ such that the number of correct equations in the system of equations is maximized, and Marzougui *et al.* [MUTS22] have suggested that this problem can be converted to an integer ILP problem using the Big-M method.

In practical scenarios, it is observed that when the equation system comprises a relatively small number of correct equations, the constrained optimization-based residual analysis algorithm may not recover the complete private key. However, the coefficients retrieved often include partially correct values, with many incorrect coefficients deviating by only ± 1. Should the attacker manage to identify which coefficients are accurate—possibly through methods like majority voting—the key space could be notably reduced. Subsequent application of the Big-M method might then enable full private key recovery.

Notably, Alg.2 employs a strategy of randomly selecting indices in each iteration, leading to varied outcomes across executions. When a single run of Alg.2 falls short of recovering the complete private key, reiterating the algorithm's core computational module may yield alternative solutions. For instance, in the case of Dilithium2, by statistically analyzing occurrences within each coefficient for ± 2, ± 1, and 0—effectively totaling five potential outcomes—and establishing threshold criteria, the solution space CRF_j (Coefficient Recovery Field) for each coefficient s_j can be delineated. When a larger pool of original equations is available, typically, only 2-3 values emerge within each CRF_j. Notable, with single signature, equating to merely 256 equations, a more cautious approach is adopted by preserving the outcomes of each scenario as potential candidate values.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{|I|} x_i \\
\text{subject to} \quad & x_i - C_i s^* \leq K \cdot (1 - x_i), && \forall i \in \{1, \ldots, |I|\} \quad (1) \\
& x_i - C_i s^* \geq -K \cdot (1 - x_i), && \forall i \in \{1, \ldots, |I|\} \quad (2) \\
& x_i \in \{0, 1\}, && \forall i \in \{1, \ldots, |I|\} \quad (3) \\
& s_j^* \in CRF_j, && \forall j \in \{1, \ldots, n\} \quad (4)
\end{aligned}
$$

Figure 2: Optimized ILP formulation used for recovering noisy equation system of $\boldsymbol{cs}_1$.

Ultimately, the challenge we address—identifying a solution s that satisfies the maximal number of equations for the given system—is reformulated into an integer linear optimization problem via the Big-M method, as illustrated in Fig.2. In comparison to previous studies [MUTS22, BVC+23], we refine constraint (4), originally defined as $s_j^* \in (-\eta, \ldots, 0, \ldots, \eta)$, which necessitated evaluating each coefficient against 5 potential values in Dilithium2,5 and 9 in Dilithium3. Our approach effectively narrows the value space, thereby enhancing the algorithm's operational efficiency.

4.3 Alternative Attack Strategies

Solving Equations using BP. SASCA aims to reduce the guessing entropy by combining side channel leakage at multiple points within the algorithm execution. It has achieved several accomplishments in attacks for traditional cryptographic algorithms [VGS14, GGSB20, LWL+22]. SASCA mostly adopts the BP algorithm, which replaces the global marginalization with local marginalization and the message passing. This process continues until convergence is achieved, resulting in the marginal probability of the target value.

The BP algorithm has demonstrated remarkable effectiveness in attacks for post-quantum cryptography represented by Kyber [PPM17, PP19, HHP+21].

When an attacker obtains the results of side-channel attacks on $\boldsymbol{y}$ or $\boldsymbol{cs}_1$, Bronchain *et al.* [BAE+23] proposed that the BP algorithm can be applied to solve integer linear equations with respect to $\boldsymbol{cs}_1$. The BP algorithm approximates the solution satisfying the constraints by updating and passing the local marginal probability in the factor graph, and the incorporation of the side-channel information further enhances the accuracy of its solution inference. The authors leveraged the characteristic of $\boldsymbol{c}$ containing only τ non-zero coefficients, restricted to 1 or -1, to reduce the scale of the factor graph, thereby improving efficiency. Meanwhile, $\boldsymbol{s}$ takes values from a uniform distribution within the range $\{-\eta, \ldots, \eta\}$.

The BP algorithm needs to rely on more equations when the success rate of the side channel attack is low. As the dimension increases, the size of the factor graph expands, and the propagation process is more prone to computational overflow problems. We have temporarily failed to complete the solution when the number of equations is excessive. Additionally, the BP algorithm necessitates incorporating the probability distribution of candidate values at leak points within the factor graph. For attack methods that do not yield probability distributions as outputs, such as fault attacks and cache attacks, the applicability of the BP algorithm needs to be further explored.

Reduction to a LWE Problem. One natural strategy for recovering the entire private key with known partial coefficients is reducing it to a special LWE problem. When the positions of recovered coefficients are known, it can be regarded as a form of leaky LWE problem and is resolved utilizing the leaky LWE estimator proposed by Dachman-Soled *et al.* [DDGR20]. In this case, known coefficients are integrated into the lattice basis as perfect hints, following which the remaining coefficients are retrieved using lattice reduction techniques such as BKZ. According to the results in [MN23], merely 45% of coefficients are sufficient to break Dilithium2 within 7 days. However, since we cannot determine which coefficients are recovered after solving the erroneous equations $\boldsymbol{cs}_1 = \boldsymbol{b}$, this method fails in this context.

When the positions are unknown, the problem of recovering the entire private key with known partial coefficients can be reduced to a ternary LWE problem. Let $\tilde{\boldsymbol{s}}_1$ be the estimator solved by the erroneous equations $\boldsymbol{cs}_1 = \boldsymbol{b}$ and substituting it into the public key $\boldsymbol{t} = \boldsymbol{As}_1 + \boldsymbol{s}_2$, we can obtain a new LWE problem $\boldsymbol{t}' = \boldsymbol{As}'_1 + \boldsymbol{s}_2$ where $\boldsymbol{t}' = \boldsymbol{t}' - \boldsymbol{A}\tilde{\boldsymbol{s}}_1$, $\boldsymbol{s}'_1 = \boldsymbol{s}'_1 - \tilde{\boldsymbol{s}}_1$. If most equations in $\boldsymbol{cs}_1 = \boldsymbol{b}$ are correct, the new problem is likely to be a sparse, ternary LWE problem sharing the same dimension as the original one. According to results in [May21, GM23], the heuristic time/memory complexities is $O(2^{0.345N})$. For Dilithium2, where $N = 1024$, this method appears impractical. We will explore the possibility of combining this method with ILP techniques in the future work.

5 Experiments and Results

5.1 Set Up

Our experimental evaluation takes place on a ChipWhisperer UFO target platform, which allows for the installation of various microprocessor modules. We select the STM32F405RGTx microprocessor module for our experiments. This setup employs the Dilithium reference implementation provided by NIST [ACD+22]. For signal acquisition, we use a BLP-1.9+ low-pass filter, a PA303 preamplifier, and a WR610Zi oscilloscope, all operating at a consistent sampling rate of 100 MSa/s.

Here we assume that the public is not compressed or it is reconstructed from a small number of signatures. If it is compressed as done in Dilithium, the new ternary LWE problem is $\boldsymbol{t}' = \boldsymbol{As}'_1 + \boldsymbol{e}$ where $\boldsymbol{e} = \boldsymbol{s}_2 - \boldsymbol{t}_0$, $\boldsymbol{t}_0$ denotes the low order bits of $\boldsymbol{t}$.

We execute the attack procedures on a desktop computer equipped with an Intel i5-13600KF processor and 32GB of DDR5 RAM, providing the necessary computational resources for our analysis. For the machine learning experiments, we utilize 4 TITAN Xp GPUs. Unless otherwise specified, our experimental results represent the averages of 10 repeated experiments.

5.2 LR-SCA for Recovering y

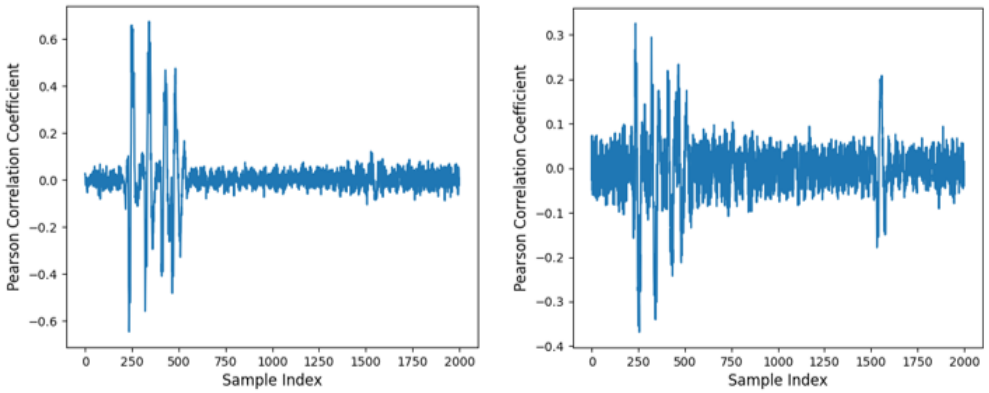
Qiao *et al.* [QLZ+23a] conducted comprehensive experiments across all operations involving y to identify potential leakages. Their research pinpointed the "polyz_unpack(a, buf)" function as exhibiting the most substantial leakage related to y, thereby designating this function as the focal point of our attack. The function, as detailed in Fig.3, is responsible for extracting an 18-bit value from a predefined random array, denoted as r, and converting it into y via the operation GAMMA1-r. This conversion is executed over 64 cycles, yielding four instances of y per cycle.

```
void polyz_unpack(poly *r, const uint8_t *a) {
    unsigned int i;

#if GAMMA1 == (1 << 17)
    for(i = 0; i < N/4; ++i) {
        r->coeffs[4*i+0] = (uint32_t)a[9*i+0] << 8;
        r->coeffs[4*i+0] |= (uint32_t)a[9*i+1];
        r->coeffs[4*i+0] &= 0x3FFFF;

        r->coeffs[4*i+1] = a[9*i+1] >> 2;
        r->coeffs[4*i+1] |= (uint32_t)a[9*i+2] << 6;
        r->coeffs[4*i+1] |= (uint32_t)a[9*i+3] << 14;
        r->coeffs[4*i+1] &= 0x3FFFF;

        r->coeffs[4*i+2] = a[9*i+3] >> 4;
        r->coeffs[4*i+2] |= (uint32_t)a[9*i+4] << 4;
        r->coeffs[4*i+2] |= (uint32_t)a[9*i+5] << 12;
        r->coeffs[4*i+2] &= 0x3FFFF;

        r->coeffs[4*i+3] = a[9*i+5] >> 6;
        r->coeffs[4*i+3] |= (uint32_t)a[9*i+6] << 2;
        r->coeffs[4*i+3] |= (uint32_t)a[9*i+7] << 10;
        r->coeffs[4*i+3] &= 0x3FFFF;

        r->coeffs[4*i+0] = GAMMA1 - r->coeffs[4*i+0];
        r->coeffs[4*i+1] = GAMMA1 - r->coeffs[4*i+1];
        r->coeffs[4*i+2] = GAMMA1 - r->coeffs[4*i+2];
        r->coeffs[4*i+3] = GAMMA1 - r->coeffs[4*i+3];
    }
#endif}
```

Figure 3: Polyz_unpack(a, buf) reference implementation.

Our objective is to recover y through side-channel analysis, with a particular emphasis on the leakage of its lower bits. However, our correlation analysis of 1,000 traces, depicted in Fig.4, reveals that focusing on r, especially its lower 8 bits, results in more pronounced leakage. This finding is attributed to the direct computational correlation between r and y, as well as the additional operations on r. Therefore, attacking the operation corresponding to r will have better results significantly higher than attacking y directly.

For the Dilithium2 scheme, we utilized 50,000 traces, applying a low-pass filter through FFT to enhance signal quality. We then selected continuous feature points exhibiting significant leakage for modeling via a LR model, ensuring that y is modeled independently



(a) PCC of r (b) PCC of y

Figure 4: PCC of r and y with 1,000 traces

for each coefficient position to mitigate potential biases from trace misalignment. Following the model's completion, the attack is executed on traces corresponding to the new signature under analysis.

In our experimental evaluation, we systematically explore the impact of varying the number of Points of Interest (POIs) on the attack's efficacy, considering two scenarios based on distribution of $\boldsymbol{cs}_1$: either $||\mathbf{cs_1}||_\infty < 2^4$ or $||\mathbf{cs_1}||_\infty < 2^5$. Tab.4 The results presented in Tab.4 are for 256 coefficients profiled and attacked, elucidate a complex interplay between the number of POIs, profiling time, and the attack's success rate (SR). Notably, with 20 POIs, the profiling time amounts to 134.2 seconds, achieving a 14.2% SR under the 4-bit assumption within 7.2 seconds, and an 8.8% SR under the 5-bit assumption in 15.89 seconds. Increasing the POIs to 300 elevates the profiling time to 952.3 seconds, yet significantly boosts the SR to 39.6% for the 4-bit assumption at 240.6 seconds, and to 33.4% for the 5-bit assumption at 461.3 seconds.

Table 4: Results for Recovering 256 Coefficients of y

#POI	**Profiling Time (s)**	**Time**[1] **(s)**	**SR**[1] **(%)**	**Time**[2] **(s)**	**SR**[2] **(%)**
20	134.2	7.2	14.2	15.89	8.8
50	170.0	21.4	16.4	36.7	10.8
100	321.4	56.4	24.1	121.6	17.3
300	952.3	240.6	39.6	461.3	33.4

[1] assumption for $||\mathbf{cs_1}||_\infty < 2^4$.
[2] assumption for $||\mathbf{cs_1}||_\infty < 2^5$.

The observed variations in the success rates of our attacks can be significantly attributed to the value range and distribution characteristics of $\boldsymbol{cs}_1$ within the Dilithium2 framework. The distribution of $\boldsymbol{cs}_1$ closely approximates a Gaussian curve, with around 95% of the data falling within a 4-bit range. This Gaussian-like distribution indicates that a more constrained value space, as presumed under the 4-bit assumption, inadvertently bolsters the success of the attack by simplifying the complexity involved. Conversely, extending the bit size assumption to 5 bits broadens the value space. While this expansion might seem to offer enhanced granularity, it paradoxically diminishes the attack's effectiveness. This counterintuitive phenomenon highlights the critical role of $\boldsymbol{cs}_1$'s Gaussian-like distribution in influencing the success rates of the attack.

Our approach showcases the potential to fully recover the value of y with success rates nearing 40% using single traces, marking a substantial leap in efficiency over prior techniques. This efficiency gain is largely due to the decreased reliance on multiple

traces for templates building and less enumeration space during the templates matching. Notably, the attack process eliminates the necessity for selecting or filtering y values, further augmenting its efficiency. Additionally, the independence of y's coefficient positions permits the execution of parallel attacks on these coefficients. Such a strategy not only accelerates the attack process but also emphasizes the versatility and practical applicability of our method for conducting side-channel attacks.

5.3 DL-SCA for Recovering cs_1

Chen *et al.* [CKA+21] have pinpointed significant leakage within the Montgomery reduction operation, a critical component of the Dilithium algorithm's reference implementation submitted to NIST. This operation forms an essential part of the entire INTT process, indispensable after polynomial multiplication in the NTT domain for executing subsequent polynomial additions. The specific implementation of the Montgomery reduction, crucial in the INTT phase, is depicted in Fig.5. Notably, the reduction process, especially the shifting and storage operations at its culmination, is probably the main operation that causes the $\boldsymbol{cs}_1$ leakage to become significant. Our targeted experimental attacks on these operations revealed a maximum SNR of 2.5 for the leakage related to $\boldsymbol{cs}_1$.

```
int32_t montgomery_reduce(int64_t a) {
    int32_t t;

    t = (int32_t)a*QINV;
    t = (a - (int64_t)t*Q) >> 32;

    return t;}
```

Figure 5: Montgomery_reduce reference implementation.

Dilithium's signing process has deterministic and randomised schemes. The deterministic scheme permits the recovery of all discarded challenge $\boldsymbol{c}$ values when the private key is known, unlike the randomized scheme. For the purposes of our analysis, we confined our attention to the leakages emanating from the final round of legitimate signatures. For Dilithium2, our collection focused on capturing a complete round of $\boldsymbol{cs}_1$ leakage. Although it is theoretically possible to gather a larger number of samples by including $\boldsymbol{cs}_2$ leakage, we chose to stay within the scope of our current discussion, which does not encompass $\boldsymbol{cs}_2$.

For Dilithium2, each signature provided 1,024 actionable training samples. By statically aligning and segmenting the data into blocks of 400 samples, we trained the CNN model as outlined in Sec.3.2. The training incorporated two strategies to mitigate the distribution skewness of $\boldsymbol{cs}_1$, considering data within $||\mathbf{cs_1}||_\infty < 2^4$ and $||\mathbf{cs_1}||_\infty < 2^5$ bounds, as well as the entire data range, involving 250,000 samples over 40 minutes.

Fig.6 presents a comparative analysis of guess entropy and success rates across models trained with varying sample sizes. Upon training with a dataset corresponding to 100 signatures (102,400 traces), the discriminators' performance metrics stabilized. Intriguingly, models predicated on the 5-bit assumption for $\boldsymbol{cs}_1$ exhibited a subtle yet notable performance uplift, with success rates advancing from 70% to 74%. This contrast in success rates, juxtaposed with a more significant discrepancy in guessed entropy for the 4-bit assumption (attributable to some actual data falling outside the labeled range), highlights a paradigm where the 5-bit assumption supersedes the 4-bit approach in effectiveness. This divergence from the findings related to y, where a 4-bit assumption was more favorable, may be ascribed to $\boldsymbol{cs}_1$'s inherently smaller value space and absence of influence from higher-order bits, as seen with y. It also accentuates the CNN model's adeptness at adapting to and learning a more intricate leakage model, as facilitated by the nuanced characteristics of

$\boldsymbol{cs}_1$'s leakage.

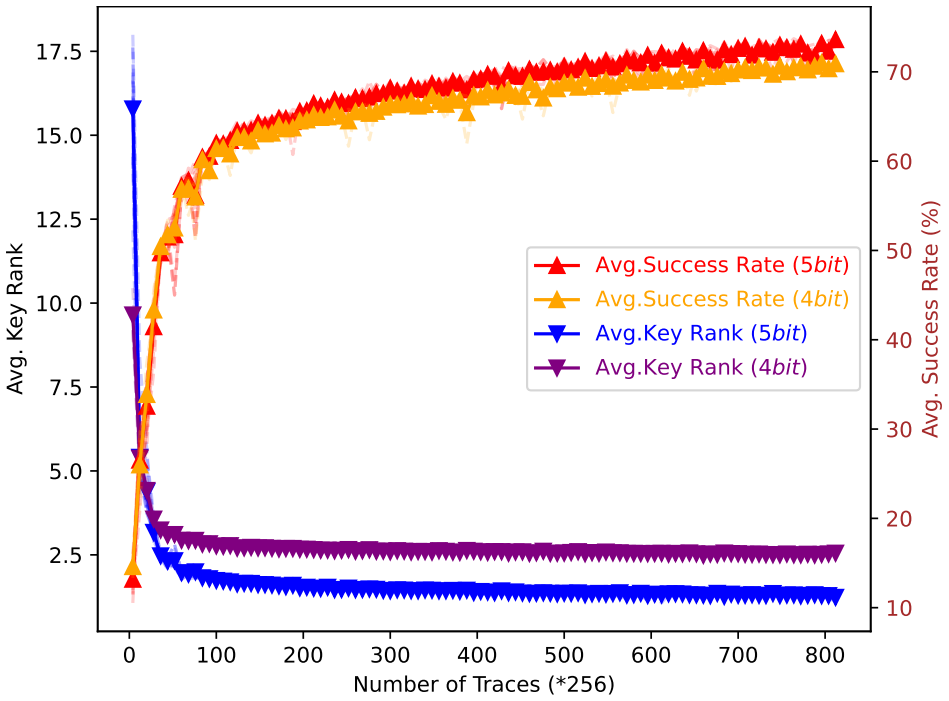


Figure 6: Guess entropy and success rates of $\boldsymbol{cs}_1$

In addition, we benchmarked our approach against traditional template attacks (TAs) for comparative analysis. In the TAs, feature points were selected based on the top 20, 50, and 100 correlation coefficients, besides attempting to utilize all available points. The success rates, as summarized in Tab.5, reveal a correlation where an increased count of feature points typically enhances the attack's success. Specifically, utilizing all feature points, TAs registered success rates of 54.6% and 57.7% under the assumptions of $||\mathbf{cs_1}||_\infty < 2^4$ and $||\mathbf{cs_1}||_\infty < 2^5$, respectively.

Table 5: Success Rates of CNN and TA

Approach	**SR of** $||\mathbf{cs_1}||_\infty < 2^4$	**SR of** $||\mathbf{cs_1}||_\infty < 2^5$	**SR of** $||\mathbf{cs_1}||_\infty < \beta$
CNN	70.5	74.3	70.6
TA (PoI 20)	25.9	23.1	22.9
TA (PoI 50)	28.7	22.6	22.6
TA (PoI 100)	40.5	40.4	31.2
TA (PoI ALL)	54.6	57.7	57.5

Despite the observed incremental improvements in TAs with a greater selection of feature points, CNN-based attacks consistently outperformed TAs in both success rates and computational efficiency. This underscores the dual benefits of CNNs in enhancing the effectiveness and efficiency of side-channel attacks, particularly when leveraging all feature points, thus establishing CNNs as the preferred methodology for dissecting and exploiting $\boldsymbol{cs}_1$ leakage in the Montgomery reduction process.

Furthermore, we applied the method outlined in 3.3 to devise a more effective attack by amalgamating the results for y with those for $\boldsymbol{cs}_1$. Tab.6 details the success rates obtained, highlighting the effect of varying the number of PoIs for y. Assuming $||\mathbf{cs_1}||_\infty < 2^4$, we attain an optimal success rate of 86%. This success rate further increases to an average

of 92.8% when the model is based on the assumption that $||\mathbf{cs_1}||_\infty < 2^5$. These findings convincingly affirm the superior efficacy of the integrated attack strategy, significantly bolstering its effectiveness.

Table 6: Success Rates for Combined y and cs_1 Results

Range of cs_1	**cs_1 SR (%)**	**#PoI of y**	**y SR (%)**	**Merged SR (%)**
$\|\|\mathbf{cs_1}\|\|_\infty < 2^4$	70.5	20	14.2	80.9
		50	16.4	82.3
		100	24.1	83.1
		300	39.6	86.7
$\|\|\mathbf{cs_1}\|\|_\infty < 2^5$	74.3	20	8.8	84.9
		50	10.8	86.5
		100	17.3	88.7
		300	33.4	92.8

5.4 Constrained Optimization-Based Residual Analysis Result

In our empirical study, we discovered that if the set of 256 equations contains more than 8 inaccuracies, it becomes impracticable to retrieve the private key using the Big-M method within an hour on our computational setup. We employed simulation data to evaluate the efficacy of constrained optimization-based residual analysis at varying success rates, aiming to establish benchmarks for future practical attacks.

Table 7: Performance of Constrained Optimization-Based Residual Analysis

Correct Eq. Ratio	**Eq. Count**	$\mathbf{d_{num}}$	$\mathbf{e_{th}}$	SR (%)	**Time (s)**
10%	65×256	50	0	100	189.5
30%	9×256	50	10	100	53
50%	4×256	50	5	100	10.0
70%	3×256	40	0	100	1.63
90%	2×256	30	0	100	0.8
95%	1×256	2	2	10	4.1

Table 7 delineates the requisite number of equations and specific parameter configurations for private key recovery across different success rates. Notably, the required equation counts are in multiples of 256, correlating directly with the number of signatures. The findings indicate that achieving a certain threshold of equations facilitates successful private key recovery, with higher success rates reducing the necessary equation count. The parameter settings adopted are heuristic; they are not optimized to their limit and can be dynamically adjusted in actual deployments to enhance efficiency. As the proportion of incorrect equations within the system increases, elevating d_{num} becomes essential to expedite the elimination of inaccuracies. In instances where the minimum equation count r_{num} exceeds 230 and all equations are accurate, private key recovery is achievable. Nonetheless, parameter adjustments may be warranted based on real-world conditions to quicken the attack.

Particularly in scenarios where only 10% of the equations are correct, successful attacks have been executed with fewer than 60×256 equations (corresponding to 60 signatures), albeit at the cost of significant time overhead due to increased e_{th}. Should the attacker possess an ample equation set, it is advisable to augment the equation count to expedite resolution. Setting d_{num} excessively high is discouraged as it could inadvertently filter out a substantial portion of correct equations in the preliminary phase.

Our methodology has demonstrated the capacity for rapid solution recovery even when the 256 equations contain errors. In trials where the correct equation ratio approximated

95%, and the experiment was repeated 100 times, we observed a 75% probability of direct private key $\boldsymbol{s}_1$ recovery within a brief period. This approach proves more efficient than conventional ILP solutions.

Furthermore, should attackers find themselves with an insufficient equation count, our method still offers a significant reduction in the private key's entropy and diminishes the time complexity associated with integer ILP. This advantage facilitates the realization of the actual attack, underscoring our method's practicality and effectiveness in cryptographic analysis.

5.5 Practical SCAs of Dilithium

The practical assault against Dilithium2 yielded insightful results. Given that attackers in real-world scenarios may only access leakage from y and $\boldsymbol{cs}_1$ independently, we delineate the comprehensive attack outcomes across three distinct cases: targeting y, $\boldsymbol{cs}_1$, and simultaneously obtaining leakage from both y and $\boldsymbol{cs}_1$.

Table 8: Signatures Required for Private Key Recovery

Range of $\boldsymbol{cs}_1$	**Side-Channel Strategy**	**#Signatures**
$\|\|\mathbf{cs_1}\|\|_\infty < 2^4$	Attack on y (#poi=300)	8(6)
	Attack on cs_1	3(3)
	Hybrid	2(2)
$\|\|\mathbf{cs_1}\|\|_\infty < 2^5$	Attack on y (#poi=300)	9(7)
	Attack on cs_1	3(2)
	Hybrid	2(1)

() corresponds to the optimal case that occurs in the attack.

Table 8 catalogs the requisite number of signatures for private key recovery under various schemes, highlighting both the maximum and minimum number of signatures needed across all successful trials in a set of 10 experiments. Intriguingly, the attack efficacy exhibits minimal variance across different assumptions regarding the bit range of $\boldsymbol{cs}_1$. Isolating the leakage of y necessitates approximately 10 signatures to unearth $\boldsymbol{s}_1$. Conversely, accessing solely the leakage of $\boldsymbol{cs}_1$ facilitates attack completion with merely 3 signatures, courtesy of the high success rates afforded by the CNN model, and in cases assuming $||\mathbf{cs_1}||_\infty < 2^5$, predominantly only 2 signatures are required. Exploiting leaks from both y and $\boldsymbol{cs}_1$ consistently mandates 2 signatures for $\boldsymbol{s}_1$ recovery. Remarkably, in our empirical investigations, we observed that about 15% of instances could leverage leaks produced by a single signature, typically demanding the recovery of 240 out of 256 coefficients via the side channel attack.

These outcomes underscore the robustness of our approach, demonstrating that even when attacking solely on y at a modest success rate, the attack can be executed with fewer than 10 signatures—an advancement over preceding efforts by Marzougui *et al.* [MUTS22] and Berzati *et al.* [BVC+23].

Furthermore, our refined attack strategy, integrating a hybrid approach with the voting technique and integer ILP as detailed in Sec.4.2, underwent 1,000 repetitions under the Constrained Optimization-Based Residual Analysis algorithm to refine the potential values for $\boldsymbol{s}_1$, typically concluding within 20 minutes. This preparatory phase was succeeded by leveraging an optimized Big-M method for final solution derivation. Observations revealed that when the side-channel attack successfully identifies more than 236 coefficients, the integer ILP phase generally concludes within 30 minutes. Should the recovered coefficients fall below this threshold, solution times often extend beyond one hour, vastly surpassing the original scheme's tolerance for up to 8 errors within a similar timeframe, hence marking a considerable enhancement. In this experimental framework, approximately 60% of datasets facilitated the recovery of the private key within an hour using a singular signature.

6 Conclusion and future work

In this study, we conducted a comprehensive side-channel analysis of Dilithium2, focusing on the polynomial addition operation $z = y + \boldsymbol{cs}_1$. Utilizing LR-based profiled attacks, we achieved a 40% success rate in recovering the complete value of y, and with the aid of a CNN model, we succeeded in recovering the value of $\boldsymbol{cs}_1$ with a 75% success rate. By integrating these findings, we enhanced the success rate for recovering $\boldsymbol{cs}_1$ through side-channel analysis to 92%. Furthermore, we introduced a constrained optimization-based residual analysis, enabling the swift recovery of the private key $\boldsymbol{s}_1$ from extensive sets of $\boldsymbol{cs}_1$ equations, even those containing errors. The results from actual attacks on Dilithium2 indicate that our approach can efficiently recover the private key $\boldsymbol{cs}_1$ with minimal leakage from generated signatures—in the optimal scenario, requiring only a single signature, with comparatively low time overhead.

Given that $\boldsymbol{cs}_2$ also undergoes the Montgomery reduction operation, our method theoretically extends to the recovery of $\boldsymbol{s}_2$, albeit necessitating approximately 2-3 signatures due to the lack of y to bolster the attack's efficacy.

Despite the substantial success in recovering most of $\boldsymbol{cs}_1$, the challenge remains that even with 230 coefficients recovered and after reducing the constraints with our constrained optimization-based residual analysis, the computational expense of solving through ILP remains significantly high. We posit that amalgamating our side-channel findings with the BP algorithm could facilitate a more consistent realization of the 1-signature attack. In future endeavors, we aim to explore more efficient mathematical methods to achieve Dilithium attacks under single signatures with improved stability and efficiency. Furthermore, we plan to investigate the effectiveness of our approach against protected implementations of Dilithium, potentially offering insights into enhancing the security measures against side-channel attacks.

References

[ACD+22] Gorjan Alagic, David Cooper, Quynh Dang, Thinh Dang, John M. Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl A. Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Daniel Apon. Status report on the third round of the nist post-quantum cryptography standardization process, 2022-07-05 04:07:00 2022.

[BAE+23] Olivier Bronchain, Melissa Azouaoui, Mohamed ElGhamrawy, Joost Renes, and Tobias Schneider. Exploiting small-norm polynomial multiplication with physical attacks: Application to crystals-dilithium. *IACR Cryptol. ePrint Arch.*, page 1545, 2023.

[BCL99] Mary Ann Branch, Thomas F. Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM J. Sci. Comput.*, 21(1):1–23, 1999.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BJL+14] Aurélie Bauer, Éliane Jaulmes, Victor Lomné, Emmanuel Prouff, and Thomas Roche. Side-channel attack against RSA key generation algorithms. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded*

Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings, volume 8731 of *Lecture Notes in Computer Science*, pages 223–241. Springer, 2014.

[BVC+23] Alexandre Berzati, Andersson Calle Viera, Maya Chartouny, Steven Madec, Damien Vergnaud, and David Vigilant. Exploiting intermediate value leakage in dilithium: A template-based approach. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(4):188–210, 2023.

[CKA+21] Zhaohui Chen, Emre Karabulut, Aydin Aysu, Yuan Ma, and Jiwu 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*, pages 583–590. IEEE, 2021.

[DDGR20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[FDK20] Apostolos P. Fournaris, Charis Dimopoulos, and Odysseas G. Koufopavlou. Profiling dilithium digital signature traces for correlation differential side channel attacks. In Alex Orailoglu, Matthias Jung, and Marc Reichenbach, editors, *Embedded Computer Systems: Architectures, Modeling, and Simulation - 20th International Conference, SAMOS 2020, Samos, Greece, July 5-9, 2020, Proceedings*, volume 12471 of *Lecture Notes in Computer Science*, pages 281–294. Springer, 2020.

[GGSB20] Qian Guo, Vincent Grosso, François-Xavier Standaert, and Olivier Bronchain. Modeling soft analytical side-channel attacks from a coding theory viewpoint. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):209–238, 2020.

[GM23] Timo Glaser and Alexander May. How to enumerate LWE keys as narrow as in kyber/dilithium. In Jing Deng, Vladimir Kolesnikov, and Alexander A. Schwarzmann, editors, *Cryptology and Network Security - 22nd International Conference, CANS 2023, Augusta, GA, USA, October 31 - November 2, 2023, Proceedings*, volume 14342 of *Lecture Notes in Computer Science*, pages 75–100. Springer, 2023.

[HHP+21] Mike Hamburg, Julius Hermelink, Robert Primas, Simona Samardjiska, Thomas Schamberger, Silvan Streit, Emanuele Strieder, and Christine van Vredendaal. Chosen ciphertext k-trace attacks on masked CCA2 secure kyber. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):88–113, 2021.

[HLK+21] Jaeseung Han, Taeho Lee, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Jihoon Cho, Dong-Guk Han, and Bo-Yeon Sim. Single-trace attack on NIST round 3 candidate dilithium using machine learning-based profiling. *IEEE Access*, 9:166283–166292, 2021.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[LWL+22] Sinian Luo, Weibin Wu, Yanbin Li, Ruyun Zhang, and Zhe Liu. An efficient soft analytical side-channel attack on ascon. In Lei Wang, Michael Segal, Jenhui Chen, and Tie Qiu, editors, *Wireless Algorithms, Systems, and Applications - 17th International Conference, WASA 2022, Dalian, China, November 24-26, 2022, Proceedings, Part I*, volume 13471 of *Lecture Notes in Computer Science*, pages 389–400. Springer, 2022.

[LZS+21] Yuejun Liu, Yongbin Zhou, Shuo Sun, Tianyu Wang, Rui Zhang, and Jingdian Ming. On the security of lattice-based fiat-shamir signatures in the presence of randomness leakage. *IEEE Trans. Inf. Forensics Secur.*, 16:1868–1879, 2021.

[May21] Alexander May. How to meet ternary LWE keys. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 701–731. Springer, 2021.

[MN23] Alexander May and Julian Nowakowski. Too many hints - when LLL breaks LWE. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part IV*, volume 14441 of *Lecture Notes in Computer Science*, pages 106–137. Springer, 2023.

[MUTS22] Soundes Marzougui, Vincent Ulitzsch, Mehdi Tibouchi, and Jean-Pierre Seifert. Profiling side-channel attacks on dilithium: A small bit-fiddling leak breaks it all. *IACR Cryptol. ePrint Arch.*, page 106, 2022.

[Ols04] Loren D. Olson. Side-channel attacks in ECC: A general technique for varying the parametrization of the elliptic curve. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 220–229. Springer, 2004.

[PP19] Peter Pessl and Robert Primas. More practical single-trace attacks on the number theoretic transform. In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*, volume 11774 of *Lecture Notes in Computer Science*, pages 130–149. Springer, 2019.

[PPM17] Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In Wieland Fischer and Naofumi

Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 513–533. Springer, 2017.

[QLZ+23a] Zehua Qiao, Yuejun Liu, Yongbin Zhou, Jingdian Ming, Chengbin Jin, and Huizhong Li. Practical public template attack attacks on crystals-dilithium with randomness leakages. *IEEE Trans. Inf. Forensics Secur.*, 18:1–14, 2023.

[QLZ+23b] Zehua Qiao, Yuejun Liu, Yongbin Zhou, Mingyao Shao, and Shuo Sun. When NTT meets SIS: efficient side-channel attacks on dilithium and kyber. *IACR Cryptol. ePrint Arch.*, page 1866, 2023.

[QS01] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): measures and counter-measures for smart cards. In Isabelle Attali and Thomas P. Jensen, editors, *Smart Card Programming and Security, International Conference on Research in Smart Cards, E-smart 2001, Cannes, France, September 19-21, 2001, Proceedings*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.

[Sho94] Peter 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*, pages 124–134. IEEE Computer Society, 1994.

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.

[VGS14] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.

[WNGD23] Ruize Wang, Kalle Ngo, Joel Gärtner, and Elena Dubrova. Single-trace side-channel attacks on crystals-dilithium: Myth or reality? *IACR Cryptol. ePrint Arch.*, page 1931, 2023.

[ZBHV20] Gabriel Zaid, Lilian Bossuet, Amaury Habrard, and Alexandre Venelli. Methodology for efficient CNN architectures in profiling attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):1–36, 2020.