

Formal Methods-Assisted Chosen Ciphertext Attacks on PQC CRYSTALS-Kyber Using Electromagnetic Emanations

Yashaswini Makaram, Davis Ranney, Adam A. Ding, David Kaeli, Yunsi Fei
 Northeastern University
 Boston, USA
 (imakaram.y, ranney.d, a.ding, d.kaeli, y.fei)@northeastern.edu

Abstract—NIST has released a set of post-quantum cryptography (PQC) standards that address the threat posed by the emergence of quantum computing. The standard includes a modular lattice-based key exchange mechanism (ML-KEM) based on the CRYSTALS-Kyber algorithm. Recent work has shown that Kyber is susceptible to electromagnetic (EM) and power side-channel attacks. A full understanding of the side-channel vulnerabilities in Kyber is of paramount importance for next-generation communication and computing infrastructures.

In this study, we target a previously unexplored section of the Kyber algorithm and implement a chosen ciphertext side-channel attack. We focus our attack on the Barrett reduction operation in the decapsulation algorithm. Compared to previous attacks on Barrett reduction, which targeted variables after the Inverse-Number Theoretic Transform (INTT), we focus on Barrett reduction on NTT variables, allowing for more general chosen ciphertexts that can evade input sanity checking. We design a scheme that requires only a set of 12 ciphertexts and side-channel EM traces of the corresponding decapsulation processes, which can reveal distinct leakages under different key values. The secret key is retrieved by pattern matching of the EM leakages. We develop an algorithm that utilizes an SMT solver to automatically select a set of ciphertexts. We implement Kyber on an ARM Cortex M4-based microcontroller and launch this new EM side-channel attack. Our results show that the attack achieves a success rate of over 95% in recovering the secret key value.

Index Terms—Hardware Security, Power/EM Side-Channel, Post Quantum Cryptography, ML-KEM, CRYSTAL-Kyber, Chosen Ciphertext Attack

I. INTRODUCTION

As quantum computing moves forward towards commercial adoption, it poses a serious threat to current cryptographic algorithms, demanding alternative post-quantum cryptography (PQC) algorithms. Key encapsulation mechanisms (KEMs) establish shared keys between two parties, facilitating follow-on secure communications, which are integral to the security of computer networks (e.g., in the transport layer of security protocols), Internet-of-Things (IoTs), and industrial control systems [1]. Many nations are migrating their cryptographic libraries to utilize PQC, upgrading cryptographic protocols on numerous devices [2]. Ensuring secure standards of communication is of paramount importance to both personal privacy and national security.

The National Institute of Standards and Technology (NIST) has released a set of Federal Information Processing Standards (FIPS) for PQC after several years of evaluation. FIPS 203 defines the module-lattice-based key encapsulation mechanism (ML-KEM), based on the CRYSTALS-Kyber algorithm, featuring Learning-With-Errors (LWE) and lattice-based modular operations to ensure that it can withstand attacks by quantum computing [3].

Recent work evaluating side-channels of ML-KEM has reported vulnerabilities in salient operations of lattice-based cryptographic schemes, including message encoding, decoding, packing, multiplication, and reduction [4]. This work finds a new leakage point in the reduction operation of the Kyber decapsulation phase. The main advantage of our side-channel attack scheme over prior work is that it will bypass the rejection of specific inputs (ciphertexts) with chosen ciphertexts that are more similar to those produced by the algorithm.

A. Main Contribution

In this study, we analyze Kyber’s susceptibility to electromagnetic (EM) side-channel attacks and make the following contributions.

- We target a previously untouched section of the decapsulation algorithm, identifying a new EM-based side-channel attack to retrieve the secret key.
- We devise an efficient side-channel attack that only requires 12 chosen ciphertexts and EM side-channel traces, and design an algorithm to define the necessary conditions for the chosen ciphertexts.
- We formulate the search for appropriate ciphertexts as a satisfiability problem and employ an SMT solver to automatically find them.
- We evaluate our attack on an ARM Cortex-M4 microcontroller in accordance with NIST recommendations. We demonstrate that the attack is effective in recovering the secret key with over 95% accuracy.

II. BACKGROUND

A. Kyber Decapsulation

Kyber is composed of three procedures: key pair (public/private key) generation, encapsulation, and decapsulation.

For encapsulation, the sender encrypts a message with the receiver’s public key and sends the ciphertext to the intended recipient. The encapsulation procedure then generates a shared symmetric encryption key based on the message. For decapsulation, the receiver uses the private key to decrypt the ciphertext, obtaining the message, and then re-encrypts the message into ciphertext. The re-encrypted ciphertext is verified with the original input. Only if the verification passes, the receiver generates the common shared key based on the message. If any third party can recover or steal the private key from the decapsulation process, they can impersonate the authenticated sender or receiver and interject or intercept the communications. Kyber has multiple security modes. In this paper we focus on Kyber-512.

Both the secret key (denoted as \mathbf{sk}) and the public key (\mathbf{pk}) are vectors of polynomials. The public key has two components, (\mathbf{A}, t) : $t = \mathbf{A} \cdot \mathbf{sk} + e$, where e is a random error. Ciphertext output produced by the encryption is also a vector of polynomials: $(\mathbf{u}, v) : \mathbf{u} = \mathbf{A}^T r + e_1, v = t^T r + e_2 + m$. Note here m is the message, and e_1, e_2 , and r are all vectors of random numbers. Decryption in the decapsulation procedure is done with the private key to recover the message: $m = v - \mathbf{sk}^T \mathbf{u}$, where the two vectors, $\mathbf{sk}^T = (sk_0, sk_1)^T$ and $\mathbf{u} = (u_0, u_1)$, each consists of two polynomials, and the vector multiplication is done by two polynomial multiplications: $\mathbf{sk}^T \mathbf{u} = (sk_0 * u_0 + sk_1 * u_1)$.

Polynomial multiplication is the convolution of the coefficients of the two polynomials. To speed up the convolution, the Number Theoretic Transform (NTT) is applied to the polynomials and the multiplication can be accomplished by element-wise multiplications (for Kyber-512 mode, 256 elements), each producing a coefficient, $mp_{j,i} = NTT(sk_j)_i * NTT(u_j)_i$, $j \in [0, 1], i \in [0, 256)$. Each coefficient of the addition result, $r_i = mp_{0,i} + mp_{1,i}$, will then go through a Barrett reduction.

B. Barrett Reduction

As a modular cryptographic algorithm, Kyber uses the Barrett and the Montgomery schemes for computing the modular reduction of values in intermediate stages. Algorithm 1 describes the steps in the Barrett reduction, where an intermediate variable, y , conditionally takes on three values ($Q, -Q$, zero) depending on the input value x , shown in Line 3. In Kyber-512, Q is 3329. When represented in a 16-bit signed two’s complement integer, $-Q$ has a Hamming weight (HW) of 13. The HW difference between the three y values can be detected through EM emanations.

This project exploits the vulnerability that arises from the Barrett reduction in Kyber decapsulation and analyzes its side-channel EM leakage to recover the private key. We carefully select the input ciphertext (u_0 and u_1) for Kyber decapsulation so that this intermediate variable y only takes on values $-Q$ (high) or zero (low). With those ciphertext inputs, the side-channel leakage at a specific time point that corresponds to the operation in Line 3 would display a large difference between a run that produces a high y value ($-Q$) and another run with a low y value (zero). We define $HW(y)$, dependent on the

Algorithm 1 Barrett Reduction

Input: x (a coefficient of polynomial output from Montgomery reduction, the value is in range $[-3328, 3328]$)

Output: $x \bmod Q$ (Q is a constant, 3329)

- 1: $l \leftarrow \left\lfloor \frac{2^{26} + Q/2}{Q} \right\rfloor$
 - 2: $p \leftarrow \left\lfloor \frac{l \cdot x}{2^{26}} \right\rfloor$
 - 3: $y \leftarrow p \cdot Q$
 - 4: **return** $x - y$
-

ciphertext input and secret key \mathbf{sk} , as the side-channel **select function**, \mathbf{f} .

C. Number Theoretic Transform

The Number Theoretic Transform (NTT) is a method used to significantly increase the speed of polynomial multiplications [5], [6]. NTT works by transforming a polynomial in a manner similar to a Fourier Transform. The version of Kyber implemented in this project utilizes the Cooley-Tukey method of NTT which calculates the NTT in stages, with the last step computed by a convolution of coefficient pairs.

D. Related Work

Prior work has investigated various side-channel vulnerabilities of Kyber, including EM/power consumptions [7]–[14], timing [15], and fault injections [4]. EM/power side-channels on Kyber decapsulation show significant vulnerability in the Fujisaki-Okamoto (FO) transform [9], as well as the NTT algorithm [10]–[12], hashing algorithms [9], and Barrett reduction [13], [14].

One effective technique in EM/power side-channel attacks on Kyber is choosing a selection function for which the output has a large difference in HW for each secret key input. The prior attacks deliberately choose select functions that can highlight these large HW differences. They also target the locations where the variables involved are transformed by INTT [13], [14]. Sim et al. [13] target the Barrett reduction operation performed at the end of decapsulation. While their attack showed potential, their methodology requires highly manipulated and unnatural chosen ciphertexts, which would be easily caught by abnormal input detection. The chosen ciphertext only sets one coefficient, i.e., the first coefficient of u_0 , non-zero; while all other coefficients of the three polynomials (u_0, u_1, v) are set 0 (the total number is $256 * 3 - 1$), as shown in Table I. Xu et al. [14] used a similar method to attack the Montgomery reduction operation in the INTT function. Their attack also requires the chosen ciphertexts to be highly specialized, allowing for only one non-zero coefficient.

Our work draws from previous studies to find a new instance of Barrett reduction, one which has not been targeted for EM/power side-channel analysis, at the end of Montgomery multiplication. Here the chosen ciphertexts (CCs) used are more flexible and closer to a normally generated ciphertext, which would evade the abnormal input detection. In our attack, v can be any value, and $NTT(u_0)$ or $NTT(u_1)$ have alternating non-zero coefficients, occupying a much larger

input space. Furthermore, we have developed an algorithm to reduce the search complexity for finding the set of ciphertexts. In Table I we compare our work with the two prior attacks.

TABLE I
COMPARISON OF OUR WORK TO PREVIOUS STUDY

Work	[13]	[14]	This Work
Side-channel modality	Power	Power	EM
Operand	INTT	INTT	NTT
Ciphertext coefficients (non-zero:zero)	1:767	1:767	384 : 384
Number of chosen ciphertexts	3	4	12

III. ATTACK APPROACH

A. Attack Overview

We adopt a chosen ciphertext attack model, where the attacker can send an arbitrary ciphertext to a receiver device for it to decapsulate using ML-KEM, and also monitor the EM emanation trace of the decapsulation process. The attacker may run this process multiple times with different ciphertext inputs to obtain a set of EM traces. Once the point-of-interest (POI) on the EM trace is identified, which corresponds to the select function operation, a single side-channel value is extracted from each EM trace for the follow-on side-channel analysis. A side-channel EM analysis is performed over the dataset to recover the secret key of Kyber.

B. Characteristics of the Chosen Ciphertexts

As the secret key \mathbf{sk} is a vector of two polynomials, we target retrieving the polynomials one by one. Take sk_0 as an example, the ciphertext should be set in a way to interact with sk_0 specifically, i.e., v can be any value, u_0 and u_1 are controlled, where $u_1 = 0$ with all of the 256 coefficients being 0. The vector multiplication $(\mathbf{sk}^T \mathbf{u})$ therefore reduces to a single polynomial multiplication $sk_0 * u_0$.

With $NTT(sk_0)$ and $NTT(u_0)$, the polynomial multiplication becomes a pairwise convolution of values, where every time two coefficients of the resulting polynomial are produced involving two secret key coefficients:

$$r_{2i} = NTT(sk_0)_{2i} \cdot NTT(u_0)_{2i} + NTT(sk_0)_{2i+1} \cdot NTT(u_0)_{2i+1} \cdot \zeta \quad (1)$$

$$r_{2i+1} = NTT(sk_0)_{2i} \cdot NTT(u_0)_{2i+1} + NTT(sk_0)_{2i+1} \cdot NTT(u_0)_{2i} \quad (2)$$

To further reduce the computational complexity, we generate the ciphertext such that all the odd coefficients, $NTT(u_0)_{2i+1}$, are set to 0, thereby ensuring the each secret coefficient only interacts with one non-zero ciphertext coefficient in the computation shown in Eq. (1) (2). The ciphertext can be retrieved by creating a polynomial with this pattern and then inverting the NTT. The specific requirements for the non-zero values are discussed in the next section. In a single EM trace with such conforming ciphertext input, there will be 256 leakage points, each corresponding to a single output coefficient which depends on one secret key coefficient. By

using the same set of EM traces, but changing the leakage point for analysis, the 256 coefficients of sk_0 can be retrieved independently. This process can be repeated to retrieve sk_1 by setting the ciphertext inputs accordingly.

Our way of choosing appropriate ciphertexts is much less rigid and limited than the approach required in the prior work [13], [14], as shown in Table I. While our work allows for 384 of the chosen ciphertext coefficients to be set to non-zero values, previous works required that the input contains only one non-zero value.

C. Chosen Ciphertext Side-channel Attack

In the prior work on a side-channel power attack focused on Barrett reduction [13], the operation targeted occurs after INTT and therefore the secret key coefficients are in the range of $[-3,3]$. With a total of only 7 possible values for one coefficient, they need to find a minimum set of 3 ciphertexts, so that the predicted sequence of power leakages for the ciphertexts under a specific secret key value will be a 3-bit encoding. For example, for secret key coefficient -3, the predicted power sequence would be $[0, 0, 0]$; while for coefficient 3, the sequence would be $[1, 1, 0]$. In general, the minimum number of chosen ciphertexts needed to differentiate any n possible secret key coefficient values is $\log_2(n)$.

In our attack, the Barrett reduction operation we target occurs before the INTT, and the secret key is in NTT form, with each coefficient taking a value in the range of $[0, 3329)$ for Kyber-512. Therefore, at least 12 chosen ciphertexts are needed to distinguish the secret key coefficient. Table II shows how an attacker cant use a set of 12 chosen ciphertexts ($cc1, \dots, cc12$) to differentiate between 3329 different secret-key values, where each row (for a specific value of the secret key coefficient) represents an encoding for the predicted EM sequence, with one side-channel leakage value taken from one EM trace. However, the search space for the set of 12 ciphertext values is exponential (3329^{12} , in the order of 10^{42}), finding a suitable set of ciphertexts by exhaustive search is infeasible. We devise an effective algorithm to set the conditions for ciphertext inputs and automatically find them for our side-channel attacks.

Note that finding the set of chosen ciphertexts is a one-time effort that takes place in the offline phase of the attack. It does not require side-channel measurements and only relies on the select function we choose. With the set of chosen ciphertext, a corresponding side-channel pattern table, shown in Table II, is generated. During the on-line phase of victim device interaction, we only need to run the Kyber decapsulation with the set of 12 chosen ciphertexts and collect their corresponding EM traces. One EM value is taken from one trace, and a vector of 12 EM values is obtained and binarized with a threshold applied (higher EM value becomes 1 and lower EM value becomes 0). A pattern matching process is applied to the vector using the pattern table. The secret key coefficient will be found from the matching row.

TABLE II
 POTENTIAL EM LEAKAGE VALUES FOR DIFFERENT SECRET KEY COEFFICIENTS (SK1-SK3329) ACROSS 12 CHOSEN CIPHERTEXTS (CCs).

Chosen Ciphertext	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12
SK1	0	0	0	0	0	0	0	0	0	0	0	0
SK2	1	1	1	1	1	1	1	1	1	1	0	0
SK3	0	1	1	1	1	1	1	1	1	1	0	0
⋮	⋮											
SK3329	0	0	0	0	⋯	0	0	0	0	1	1	1

IV. SMT SOLVER-BASED ALGORITHM

Algorithm 2 Finding the Ciphertext for One Column

Input: A : the group size for the column (A is initially two times of the group size of the previous column. A of the first column is 2)

Output: CC : the ciphertext found for the column; I : the column pattern values; Group size B .

- 1: Choose the select function f , over the ciphertext coefficient (cc) and the secret key coefficient (sk), that gives one-bit EM prediction
 - 2: Set initial pattern for the column: I , with repeated groups of alternating 0s and 1s, and the last group partial ($3329 \bmod A$)
 - 3: $S_{cc} \leftarrow \{f(cc \cdot sk) \mid sk = 0, 1, 2, \dots, 3328\}$
 - 4: Set tolerance $\epsilon = 0$
 - 5: **while** $\epsilon < 12$ **do**
 - 6: $B \leftarrow A$
 - 7: **while** $B > A/2$ **do**
 - 8: $cc = \text{Z3}(S_{cc}, I, \epsilon)$ {//run the SAT solver to find cc that makes S_{cc} mostly match I except for ϵ bits (essentially I has number of ϵ bit flips: $0 \rightarrow 1$ and $1 \rightarrow 0$ }
 - 9: **if** cc **then**
 - 10: Validate by checking for all the columns so far, in every group of B rows, the vectors are distinct
 - 11: **Return** cc
 - 12: **else**
 - 13: $B \leftarrow B - 2$
 - 14: Re-populate I with the new group size B
 - 15: **end if**
 - 16: **end while**
 - 17: $\epsilon \leftarrow \epsilon + 2$
 - 18: **end while**
-

As a brute-force approach is computationally infeasible for finding the set of chosen ciphertexts, we design an algorithm to iteratively create a pattern table in a divide-and-conquer fashion and then automatically find suitable ciphertexts with a SAT solver. The key idea is to populate the table column by column with specific patterns, starting from the leftmost one.

For each column, the pattern setting processes begins with a group size. The pattern is alternating repeated 0's (half of the group) and 1's. The group size for the first column is 2. From the second column on, the group size is double that of the previous column. Algorithm 2 summarizes our process

for finding the ciphertext for each column and determining the pattern for each column. The column is initialized with a pattern (I) with group size A , where the first half of each group is all 0's, followed by the rest all 1's. All the possible secret values ($[0, 3329)$) are plugged into the select function (f , over cc and sk) for producing a vector of bits S_{cc} (Line 3). We then run the SMT solver Z3 [16] to find the ciphertext (cc) that yields S_{cc} matching I . However, if the SMT solver does not return a valid ciphertext, we need to relax the column matching in two ways. First, we reduce the group size by 2, and repopulate the column pattern I with the new groups. This is done in the inner `while` loop (Lines 7 to 16) until the group size decreases to the previous column group size. If there is still no valid ciphertext, the second relaxation is to adjust the matching tolerance. Instead of an exact match between the two vectors, we allow up to ϵ bits of mismatch as shown in the outer `while` loop (Lines 5 - 18). This is essentially adjusting the column pattern in some groups, where instead of a half group of 0's followed by a half group of 1's, the bit values are redistributed in the group, though still balanced.

For the Z3 SMT solver (applied in Line 8), the ciphertext (cc) is represented as a Boolean variable. For Kyber512, each coefficient of the ciphertext polynomials is 12-bit. The column pattern I is used as constraints. The SMT solver searches for a suitable ciphertext that makes the sequence of predicted EM leakages match the column pattern I , with some mismatches ϵ allowed. The resulting ciphertext will be validated (Line 10).

The complexity of our algorithm for finding all the ciphertexts is $O(n \cdot \log_2^2(n))$, where $n = 3329$, much more efficient and attainable than the exhaustive search of $O(n^{\log_2(n)})$. Our empirical results show that we need to adjust the column group size with relaxations starting from column 5.

V. EXPERIMENTS AND RESULTS

We implement the proposed attack in a real-world setting and evaluate its effectiveness. Next, we describe our experimental setup and present our results.

A. Experimental Setup

Hardware: To test our attack, experiments were carried out on an STM32F417 Piñata board. This board includes an ARM Cortex-m4 processor, which runs at 168 MHz [17], and uses 1 MB FLASH memory and 196 KB SRAM.

Software: The code base used for the experiments is from the reference code found on the official NIST website for PQC. We also used the codebase produced in the PQCclean library.

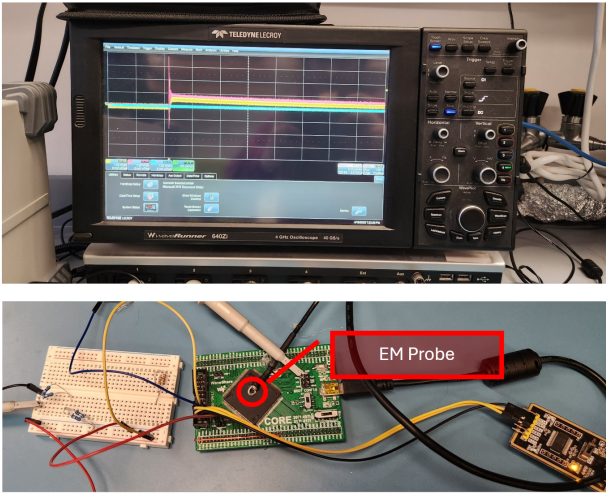


Fig. 1. Top: Oscilloscope used for trace collection. Bottom: The test board analyzed, connected to probes for EM collection and triggering.

Data Collection: Our evaluation platform is shown in Fig. 1. We use a Teledyne LeCroy WaveRunner 640Zi oscilloscope for EM trace collection. The oscilloscope collects voltage data from an EM probe placed on the microcontroller.

B. Chosen Ciphertext Side-channel Attack Implementation

Side-channel Leakage: We first identify the POI in the EM traces. We use Welch’s T-test [18] to compare two groups of EM traces. The T-test plot, shown in Fig. 2, shows the point-to-point level of difference between the two sets of traces. These sets were predicted by the algorithm to produce ‘high’ or ‘low’ EM leakage. A T-value higher than 4.5 is deemed statistically differentiable. We find that the waveform index 615 corresponds to the select function, which yields a high T-value (15.9) and a low P-value ($9.58e^{-30}$). We extract the EM value at this time point for the follow-on side-channel attack.

Profiling: Side-channel EM leakages must be categorized as either ‘high’ or ‘low’, based on a threshold. To this end, we collect EM traces of the decapsulation process with all possible ciphertexts (enumerating the coefficient values) and a known secret key. These ciphertexts are categorized by their algorithmically predicted HW from the select function. Note the input space for the select function (possible values for ciphertext coefficients) is over $NTT(\mathbf{u}_0)_i \in [0, 3329]$ to ensure the full range of inputs was covered. Originally, 1000 traces were collected for each ciphertext to improve the side-channel signal-to-noise ratio (SNR) through averaging.

Filtering: While averaging is an effective method for side-channel noise reduction, it requires collecting EM traces for repeated runs of the decapsulation process under a ciphertext, which would require significant execution overhead. Also, the environmental noise (due to heat, vibration, or nearby devices) will come into play if the data collection takes too long. We devise a pre-processing method to reduce the number of repetitions. The parameters of the filter and associated SNR improvements are discussed in the following section.

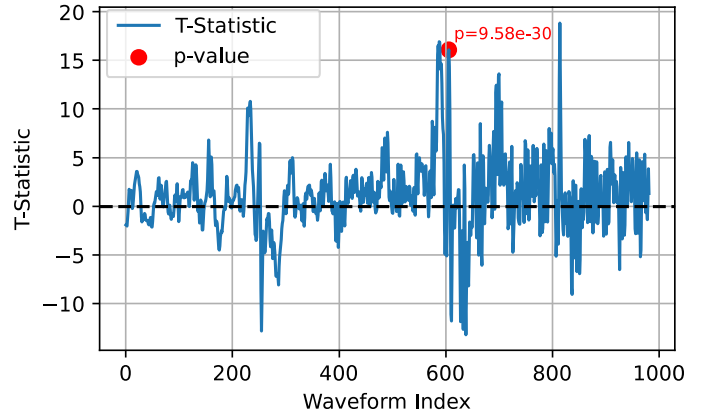


Fig. 2. T-Test of EM traces showing strong leakage at index 615

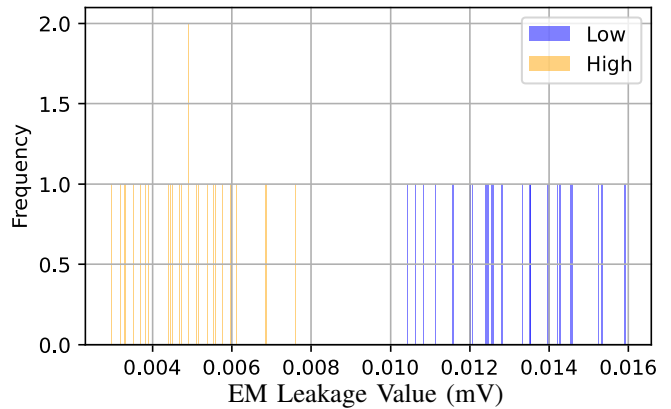


Fig. 3. Distribution of leakage values for the high HW group and low HW group, averaging values over 1000 traces.

C. Experimental Results

1) *SAT Solver-based Pattern Generation Algorithm:* As Algorithm 2 illustrates, we employ two relaxation methods to set the pattern table column by column and automatically find the suitable ciphertext for each column. Table III presents the results in terms of the group size (B) and the mismatch tolerance (ϵ). The group size B for Column 5 deviates from 2^n , while the group sizes of the subsequent columns (except for the last one) are all 2 times the preceding one. The maximum mismatch (ϵ) is 8, for Column 9.

2) *Side-channel EM Attack with Chosen Ciphertexts:* For the side-channel EM attack relying on a vector of 12 EM side-channel values, they must be categorized into one or zero correctly. During the profiling stage, a distribution of EM leakages is collected for all the ciphertext coefficient inputs, and an optimal threshold is found to classify high and low EM values. Given that clustering has an accuracy α , the accuracy of chosen ciphertext side-channel attack would be α^{12} . Fig. 3 shows a 100% clustering accuracy with a threshold value of 0.009, when each ciphertext decapsulation is repeated 1000 times for averaging the EM leakage.

To improve the attack efficiency (reducing the number of

TABLE III
GROUP SIZE (B) AND TOLERANCE (ϵ) USED DURING ALGORITHMIC SEARCH FOR CHOSEN CIPHERTEXTS (CCs)

Chosen Ciphertext	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12
B	2	4	8	16	26	52	104	208	416	832	1664	3454
ϵ	0	0	0	0	0	0	0	0	8	6	2	0

*3454 is larger than the search space, so the second half of the group was cutoff at 3329.

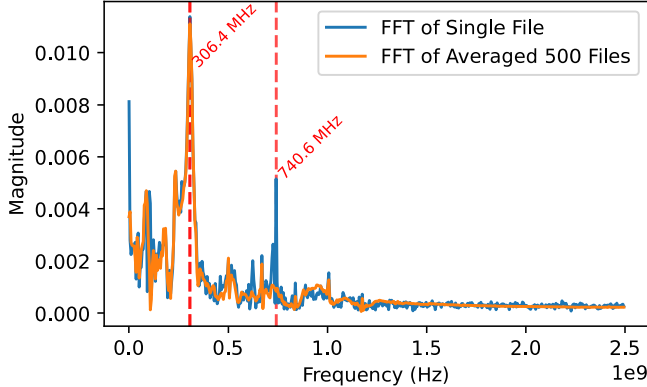


Fig. 4. Spectrum of traces produced with FFT, for a single trace and an average of 500 traces

repetitions), we filter the side-channel leakage to increase the SNR. We first conduct a spectrum analysis of the side-channel EM trace with FFT. Fig. 4 shows a comparison between the frequency spectrum of a single trace and the average of 500 traces. Frequencies where the magnitude is reduced noticeably by averaging are determined to contain noise. We notice that the highest component is at a frequency of 306 MHz (near the 2nd-order harmonic of the working frequency) and remains unaffected by averaging. In contrast, components at frequencies beyond 1 GHz are diminishing. While strong noise is present at 740.6 MHz, valuable signals can still be seen between 740 MHz and 1.2 GHz. We apply a low-pass filter to the EM traces using a cutoff frequency varying in the range of 500 MHz and 1.5 GHz. Fig. 5 shows how the error rate changes with the cutoff frequency. Using the best cutoff frequency of 1.2 GHz and only 10 repetition traces for averaging, the clustering accuracy is $\alpha = 99.6\%$ as shown in Fig. 6, and the attack success rate would be $\alpha^{12} = 95\%$. For the few incorrect values, we can identify which coefficients are likely to be incorrect based on outliers that are close to the threshold, and find the correct values for these few instances through exhaustive search.

VI. CONCLUSIONS

This work proposes a novel chosen-ciphertext side-channel attack on the PQC Kyber ML-KEM (FIPS-203), exploiting the EM leakages of Barrett reduction during decryption. The targeted location enables the use of more naturally chosen ciphertexts compared to prior work. An SMT solver-based pattern generation and matching algorithm is developed to automatically find the set of ciphertexts and the predicted

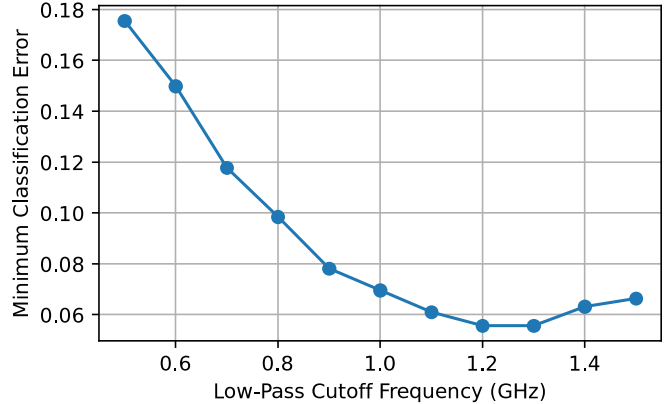


Fig. 5. Error in clustering versus the low-pass filter cutoff frequency

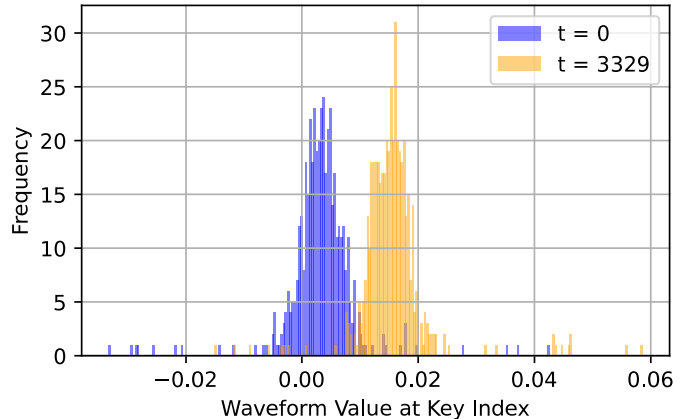


Fig. 6. Clustering of EM leakages after the use of an optimal low-pass filter

leakage patterns. Experimental results show that the attack requires as few as 10 traces per chosen ciphertext and a total of 12 ciphertexts to achieve a 95% key recovery accuracy. The results demonstrate that Barrett reduction introduces exploitable leakages, highlighting the need for improved side-channel resistance for wider adoption of post-quantum cryptographic implementations. Future work will evaluate countermeasures, such as masked arithmetic, hardware protections, and alternative reduction techniques, to mitigate leakage.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation under grant CNS-1916762 and industry funds of the IUCRC Center for Hardware and Embedded Systems Security and Trust (CHEST).

REFERENCES

- [1] T. Liu, G. Ramachandran, and R. Jurdak, “Post-Quantum Cryptography for Internet of Things: A Survey on Performance and Optimization,” Jan. 2024, arXiv:2401.17538 [cs]. [Online]. Available: <http://arxiv.org/abs/2401.17538>
- [2] “Post Quantum Government Initiatives by Country and Region.” [Online]. Available: <https://www.gsma.com/newsroom/post-quantum-government-initiatives-by-country-and-region/>
- [3] P. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, Nov. 1994, pp. 124–134. [Online]. Available: <https://ieeexplore.ieee.org/document/365700/>
- [4] P. Ravi, A. Chattopadhyay, J. P. D’Anvers, and A. Baksi, “Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Dilithium): Survey and New Results,” *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 35:1–35:54, Mar. 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/3603170>
- [5] “Conceptual Review on Number Theoretic Transform and Comprehensive Review on Its Implementations,” Nov. 2020.
- [6] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “Algorithm Specifications And Supporting Documentation.”
- [7] A. Karlov and N. L. d. Guertechn, “Power analysis attack on Kyber,” 2021, publication info: Preprint. MINOR revision. [Online]. Available: <https://eprint.iacr.org/2021/1311>
- [8] Y. Yang, L. Wu, X. Zhang, and M. Chinbat, “Power Analysis on Hardware Implementation of CRYSTALS-Kyber,” in *2024 IEEE 18th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, Nov. 2024, pp. 1–5, iSSN: 2163-5056. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10839699>
- [9] P. Ravi, S. S. Roy, A. Chattopadhyay, and S. Bhasin, “Generic Side-channel attacks on CCA-secure lattice-based PKE and KEMs,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 307–335, Jun. 2020. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/8592>
- [10] 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*, W. Fischer and N. Homma, Eds. Cham: Springer International Publishing, 2017, pp. 513–533.
- [11] P. Pessl and R. Primas, “More Practical Single-Trace Attacks on the Number Theoretic Transform,” in *Progress in Cryptology – LATIN-CRYPT 2019: 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2–4, 2019, Proceedings*. Berlin, Heidelberg: Springer-Verlag, Oct. 2019, pp. 130–149.
- [12] B.-Y. Sim, J. Kwon, J. Lee, I.-J. Kim, T.-H. Lee, J. Han, H. Yoon, J. Cho, and D.-G. Han, “Single-Trace Attacks on Message Encoding in Lattice-Based KEMs,” *IEEE Access*, vol. 8, pp. 183 175–183 191, 2020. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9217595>
- [13] B.-Y. Sim, A. Park, and D.-G. Han, “Chosen-Ciphertext Clustering Attack on CRYSTALS-KYBER Using the Side-Channel Leakage of Barrett Reduction,” *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 382–21 397, Nov. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9786796>
- [14] Z. Xu, O. Pemberton, S. S. Roy, D. Oswald, W. Yao, and Z. Zheng, “Magnifying Side-Channel Leakage of Lattice-Based Cryptosystems With Chosen Ciphertexts: The Case Study of Kyber,” *IEEE Transactions on Computers*, vol. 71, no. 9, pp. 2163–2176, Sep. 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9591340>
- [15] J.-P. D’Anvers, M. Tiepelt, F. Vercauteren, and I. Verbauwhede, “Timing attacks on error correcting codes in post-quantum schemes,” in *Proceedings of ACM Workshop on Theory of Implementation Security Workshop*, 2019, pp. 2–9.
- [16] L. M. de Moura and N. S. Bjørner, “Proofs and refutations, and z3.” in *LPAR Workshops*, vol. 418. Doha, Qatar, 2008, pp. 123–132.
- [17] *Piñata Board Manual*, Riscure B.V., 2017, rev. 2.2.
- [18] B. L. WELCH, “The generalization of ‘student’s’ problem when several different population variances are involved,” *Biometrika*, vol. 34, no. 1-2, pp. 28–35, 01 1947. [Online]. Available: <https://doi.org/10.1093/biomet/34.1-2.28>