Blink: Breaking Lattice-Based Schemes Implemented in Parallel with Chosen-Ciphertext Attack

Jian Wang[1,2], Weiqiong Cao[1(✉)], Hua Chen[1], and Haoyuan Li[3]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
{wangjian2019, caoweiqiong, chenhua}@iscas.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Zhongguancun Laboratory, Beijing, China
lihy@zgclab.edu.cn

Abstract. As the message recovery-based attack poses a serious threat to lattice-based schemes, we conducted a study on the side-channel security of parallel implementations of lattice-based key encapsulation mechanisms. Initially, we developed a power model to describe the power leakage during message encoding. Utilizing this power model, we propose a multi-ciphertext message recovery attack, which can retrieve the required messages for a chosen ciphertext attack through a suitable message recovery oracle. Building upon the successful message recovery, we further develop a key recovery method based on a ciphertext-choosing strategy that maximizes key recovery accuracy, as well as a lattice reduction attack capable of solving the whole private key from the target LWE instance. To assess the effectiveness of the attack, we conducted experiments using Kyber768 implemented on a Xilinx FPGA board. The experimental results demonstrate that our attack could successfully recover the private key with 9600 power traces and a computational complexity of 100 bikz, which is a significant advantage over existing attacks. Notably, our attack remains effective despite countermeasures such as masking and shuffling being implemented. This study reveals that parallel implementations remain vulnerable to side-channel attacks, and highlights the necessity of additional analysis and countermeasures for lattice-based schemes implemented in parallel.

Keywords: PQC · Kyber · Parallel implementation · FPGA · CCA · Lattice reduction.

1 Introduction

In preparation for the impending threat of quantum computers to classical public-key cryptographic algorithms (e.g., RSA, ECC), the National Institute of Standards and Technology (NIST) initiated the standardization process of post-quantum cryptographic (PQC) algorithms in 2016. In July 2022, after three rounds of evaluation, NIST adopted the lattice-based scheme CRYSTALS-Kyber

(abbr. Kyber) as the first post-quantum key encapsulation mechanism (KEM) to be standardized [20]. Throughout the standardization process, lattice-based schemes have played a significant role.

For a real-world deployed scheme, implementation security is of utmost importance. Therefore, analyzing potential side-channel vulnerabilities and respective countermeasures is a crucial task. NIST has also emphasized the importance of resistance to side-channel attacks in the standardization process [21]. In this regard, it is noteworthy that many lattice-based KEMs have lots of similarities on higher abstraction levels. For this reason, attacks that exploit such high-level properties have broad applicability to a wide range of schemes.

1.1 Related Work

There are two primary categories of side-channel attacks for lattice-based KEMs. One exploits the leakage of polynomial multiplication to recover the private key, including single-trace attacks based on belief propagation [16] [27] or classical correlation power attack [33] [22]. The other is the side channel-assisted chosen ciphertext attack, which primarily employs power analysis to retrieve intermediate values like message bits, comparison results, etc. Subsequently, the attackers can conduct chosen ciphertext attacks to accomplish private key recovery. Such attacks fall into three categories depending on the intermediate values recovered, including message recovery-based attacks [38], plaintext checking-based attacks [31], and decryption failure-based attacks [15].

Among the various types of attacks discussed above, message recovery-based attacks have received considerable attention. Message recovery attacks were first proposed for retrieving session keys established between the communicating parties [3,34]. Xu et al. [38] initially suggested that the power leakage from decapsulating a limited number of chosen ciphertexts may enable an attacker to recover the private key. They focused on the message encoding in Kyber and leveraged a Simple Power Analysis (SPA) to recover the message bit by bit. Subsequently, Wang et al. [36] demonstrated the influence of compilation optimizations on the leakage model of message encoding and utilized deep learning-based power analysis to successfully break a masked Kyber implementation. Similarly, there have been successive proposals for attacks on other lattice-based schemes and protected implementations [23–25].

However, current attacks of this nature depend purely on power leakage information aligned with the Hamming weight model and serial processing of message bits/bytes in the software implementation. Consequently, they exclusively operate on embedded software platforms. As a result, these attacks are ineffective against hardware implementations that utilize parallel processing capabilities. In 2023, Ji et al. [17] demonstrated a message recovery attack on a Kyber's hardware implementation. However, their attack is restricted to message recovery through enumeration up to 2^{64}, rendering private key recovery almost impossible. In addition, the correlation electromagnetic attack proposed by Rodriguez et al. [33] necessitates the knowledge of the initial values of the target

registers. This raises the question of whether CCA-secure lattice-KEMs implemented on hardware platforms, such as FPGA, are vulnerable to more practical side-channel attacks.

1.2 Contributions

In this study, we provide a positive answer to the question. We investigated the security of lattice-based KEMs when they are implemented in parallel, especially on the FPGA platforms. Basically, our work consists of two logically connected parts.

The first and fundamental part is a theoretically sound key recovery attack against lattice-based KEMs. Specially,

- We study the side-channel leakage of the message encoding and develop a generalized power leakage model for parallel implementations of the lattice-based KEMs. We then present a new multi-ciphertext message recovery attack based on the power model. By utilizing structured ciphertexts, this attack enables partial recovery of message bits, which can be beneficial for a chosen ciphertext attack. With an appropriate binary oracle, the attacker can retrieve a complete message by analyzing a few power traces. The attack can be applied to a broad range of schemes since it targets a high-level property. The attacker can select different oracles based on their capabilities and the intended attack targets. Furthermore, we observed that the attack remains effective against the protected implementations with masking and shuffling countermeasures.
- As the oracle in a side-channel attack is typically imperfect, the accuracy of key recovery decreases multiplicatively as the accuracy of message recovery decreases. To address this issue, we propose two methods for achieving a complete key recovery. The first involves using an optimal binary recovery tree to reduce the number of dependent message bits of a secret coefficient, this maximizes the key recovery accuracy while maintaining the same message recovery accuracy. Innovatively integrating the a posteriori probabilities of message bits into a lattice reduction attack is the second step. By transforming the a posteriori distribution from message bits to secret coefficients, this approach creates an efficient lattice reduction attack to recover the private key using side information acquired from our message recovery attack.
- Kyber serves as a case study to demonstrate and validate our key-recovery attack. The results of our simulation experiment confirm the correctness of our method and also show that our ciphertext-choosing strategy outperforms two preceding methods under the same attacker capabilities.

The second part is supportive, aiming to provide an illustrative case of executing a key recovery attack in practice. Specifically, we instantiate a shallow neural network as our side-channel oracle and perform a practical power analysis attack against a four-way parallel implementation of Kyber's message encoder on a Xilinx Artrix-7 FPGA board, which is highly comparable to other LPR-based

schemes. The attack results show that our method is feasible in a practical attack setting. Compared to previous attacks on hardware implementation of lattice-based schemes, our attack requires fewer power traces and achieves a higher success rate. This attack serves as an initial demonstration of the practicality of the proposed fundamental method in the first part.

Outline In Sec. 2, we provide the notations used in this paper. In addition, we recall the LPR scheme and provide the necessary knowledge related to our attack. In Sec. 3, we detail our attack on a tiny LPR-based KEM. Sec. 4 then applies the attack on Kyber768 and provides an instance of our attack. In Sec. 5, we present practical evidence that supports our claims by attacking a message encoder that is implemented on an FPGA board. We examine the impact of our attack on masking and shuffling countermeasures and discuss potential future work in Sec. 6.

2 Premilinaries

2.1 Notation

To simplify notation, we denote by $\{0,1\}^k$ the set of bit arrays of length k and by $\{0,1\}^*$ the set of bit arrays of arbitrary length. Furthermore, the ring of integers module q is denoted as $\mathbb{Z}_q$, and we denote by $\mathcal{R}_q$ the ring $\mathbb{Z}_q[X]/(X^n+1)$. Regular font letters, such as v, denote elements in R_q, with $v[i]$ representing the i-th coefficient in v. Bold lower-case letters indicate vectors with entries in $\mathcal{R}_q$. By default, all vectors are column vectors, and we write $\mathbf{v}[i]$ to denote its i-th entry for a vector $\mathbf{v}$, and $\mathbf{v}[i][j]$ to denote the j-th coefficient in its i-th entry when the entry is a polynomial (with indexing starting at zero). For a set of vector M, we refer to $M[i,\cdot]$ as the set of i-th entries in every vector. When assigning a vector to a polynomial, we indicate that the entries in the vector are assigned to the coefficients in the polynomial one by one. When we refer to a message m, we use $m[i]$ to denote the i-th bit in this message. For an element $x \in \mathbb{Q}$ we denote by $\lfloor x \rceil$ to the closest integer with ties being rounded up. For a probability distribution S, we write $s \leftarrow S$ to denote the s is chosen according to the distribution S. The symbol $\cdot$ represents the multiplication between polynomials and the symbol $\odot$ represents (accumulate) vector multiplication between vectors (matrices and vectors). Additionally, the hamming weight $\mathrm{hw}(a)$ corresponds to the number of 1 bits in a, and the hamming distance $\mathrm{hd}(a,b) = \mathrm{hw}(a \oplus b)$.

2.2 The LPR scheme

Lyubashevsky, Peikert, and Regev [18] extended the original Learning with Errors (LWE) problem [32] to an algebraic variant known as Ring-LWE (RLWE) in 2010. They also defined a public-key encryption scheme, frequently referred to as the LPR scheme. Currently, the LPR scheme has emerged as the primary method for constructing lattice-based KEMs. The leading lattice-based KEMs, such as NewHope [2], Kyber [5] and Saber [11], are all founded on the LPR scheme.

The plain LPR scheme is secure against the chosen-plaintext attack (CPA) and comprises three procedures, namely, key generation, encryption (CPA-Enc), and decryption (CPA-Dec). The chosen-ciphertext attack (CCA) based on message recovery exploits the restricted range of secret coefficient candidates, enabling the attacker to narrow down potential candidates by utilizing the corresponding message bit of a chosen ciphertext. After several reductions, only one candidate remains, which is the correct coefficient. To protect against the chosen ciphertext attack, many LPR-based schemes utilize the Fujisaki-Okamoto transform [14] to construct a CCA-secure KEM. A CCA-secure KEM entails key generation, encapsulation, and decapsulation (CCA-Dec). The decapsulation process involves a decryption and a re-encryption (CPA-ReEnc) procedure as illustrated in Alg. 1. In addition, a hash function G: $\{0,1\}^* \rightarrow \{0,1\}^{2\times n}$ and a key derivation function KDF: $\{0,1\}^* \rightarrow \{0,1\}^n$ is utilized. During the decapsulation process, the initial step is to perform decryption. The ciphertext is decrypted to an n-bit message using the private key, whose coefficients belong to $[-\eta, \eta]$ and follow a discrete near-Gaussian distribution. The message is subsequently re-encrypted to prevent outside exposure. The re-encryption will reject this decapsulation and return a random bit sequence once the ciphertext is modified, making classical chosen-ciphertext attacks infeasible. In other words, if an attacker is able to access the message, such as through a side channel attack, a chosen-ciphertext attack remains possible.

2.3 Message Encoding

In lattice-based KEMs, to incorporate message bits into arithmetic operations, the messages need to be encoded as polynomials, with each message bit mapped to a coefficient in the polynomial.

The encoding of a single message bit can be simply represented as Eq. 1:

$$mp[i] = \begin{cases} 0, & \text{if } m[i] = 0 \\ \lfloor \frac{q}{2} \rceil, & \text{if } m[i] = 1 \end{cases} \tag{1}$$

Here, $m[i]$ represents the i-th message bit, and $mp[i]$ represents the encoded polynomial coefficient of the corresponding bit.

The message encoding operation generally takes place within the encryption process of key encapsulation and the re-encryption process during key decapsulation. Recovering the encoded messages helps an attacker reconstruct the session key between the communicating parties [3]. In addition, an attacker can retrieve the long-term private key directly from the recovered messages during re-encryption by utilizing chosen-ciphertext attacks.

The message decoding is the reverse process of message encoding and can be depicted as Eq. 2:

$$m[i] = \begin{cases} 0, & \text{if } mp[i] \in (\frac{3q}{4}, q) \cup [0, \frac{q}{4}] \\ 1, & \text{if } mp[i] \in (\frac{q}{4}, \frac{3q}{4}] \end{cases} \tag{2}$$

Algorithm 1 CCA-Dec

Require: $sk = (s, pk, h, z)$, $c = (u, v)$
Ensure: $ss \in \{0,1\}^n$
1: $m' = \text{CPA-Dec}(s, c)$
2: $(K, r') = \text{G}(m'||h)$
3: $c' = \text{CPA-ReEnc}(pk, m', r')$
4: **if** $c == c'$ **then**
5: $\quad ss = \text{KDF}(K||H(c))$
6: **else**
7: $\quad ss = \text{KDF}(z||H(c))$
8: **end if**

9: **procedure** CPA-DEC
Require: s, $c = (u, v)$
Ensure: $m \in \{0,1\}^n$
10: $\quad mp = v - u \cdot s$
11: $\quad m = \text{Decode}(mp)$
12: **end procedure**

13: **procedure** CPA-REENC
Require: $pk = (a, b)$, $m \in \{0,1\}^n$, $r \in \{0,1\}^*$
Ensure: $c = (u, v)$
14: $\quad r, e_1, e_2 \in \mathcal{R}_q \leftarrow \mathcal{X}^n(r)$
15: $\quad u = ar + e_1$
16: $\quad v = br + e_2 + \text{Encode}(m)$
17: **end procedure**

2.4 Integration into LWE

In 2020, Dachman-Soled et al. presented a framework for the integration of side information into LWE. This generalizes the primal attack on LWE [10]. Initially, the LWE instance is embedded into the Distorted Bounded Distance Decoding Problem (DBDD). Thereafter, side information is considered as hints and integrated into the DBDD instance, which makes the instance easier to solve. After all available hints have been applied, the DBDD instance is embedded into an unique Shortest Vector Problem (uSVP) instance which can be solved using lattice reduction techniques such as the Blockwise Korkine-Zolotarev (BKZ) algorithm [8]. Additionally, a tool is provided to assess the complexity of solving the secret key of the left lattice instance, with "bikz" as the unit. The "bikz-to-bit" conversion of security estimation has a dependency on specific parameters. In Kyber, the conversion can be demonstrated as a security increase of 0.292 bit every 1 bikz under classical computation [1]. Furthermore, Dachman-Soled et al. also provide an example of combining their method with a template attack on FrodoKEM [7]. They utilize perfect hints and approximate hints corresponding to coefficients with rather high guessing confidence and the left part. In their results, the exact security of FrodoKEM with the CCS2 parameter set decreases from 448 bikz to 29 bikz. In addition, coefficients with relatively high guess con-

fidence can also be assumed to be perfect hints, which are referred to as extra guesses that can further reduce the difficulty of solving the instance, while the success rate of solving the instance is correspondingly lower. Please refer to the original paper for more details if needed.

3 A General Attack Method

In this section, we present a CCA-assisted side-channel attack against parallel lattice-based KEM implementations, enabling an attacker to retrieve the target device's complete private key using a limited number of power traces. As parallel implementations are chiefly found in hardware platforms, specifically FPGA and ASIC, we concentrate on hardware implementations in the following sections. Nevertheless, our attack holds equally valid for software parallel implementations.

3.1 Power Model

In hardware implementation of cryptographic primitives, power consumption is associated with the number of flipped bits in the registers. As illustrated in Sec. 2, the message encoding process involves mapping each message bit to a constant in $\mathbb{Z}_q$. Specifically, a bit of 1 is mapped to $\lfloor \frac{q}{2} \rceil$, while a bit of 0 is mapped to 0. Consequently, the number of flipped bits in the registers that store encoded coefficients is proportional to the number of flipped bits in the encoded message bits. In essence, the power leakage that results from bit flips in the encoded coefficients may facilitate attackers in retrieving the corresponding message bits.

By examining multiple hardware implementations of lattice-based KEMs [37, 39, 40], we have identified a common pattern in the message encoder components. This pattern meets the following criteria: 1) In each clock cycle, P bits of message, referred to as a **nibble**, are encoded simultaneously; 2) The encoded coefficients are stored in P registers, with each register's bit width being $\lceil \log q \rceil$; 3) Prior to the message encoding operation, the P registers have been utilized for other operations, rendering the retrieval of their initial values impossible for any potential attacker.

It is worth noting that, for compatibility with other arithmetic operations, the value assigned to P in this context is usually selected to be a small power of 2, such as 2, 4, 8, and the like.

Based on the aforementioned observations, we can consolidate relevant information to develop a power model. Specifically, when encoding two consecutive nibbles, denoted as n_0 and n_1, within a **block**, the measured power consumption depends on the number of bits that differ between n_0 and n_1. If we focus on the i-th bit in either of the two nibbles, we can estimate the hypothesized power consumption using Eq. 3,

$$h = \alpha * \mathrm{hd}(n_0[i], n_1[i]) + N_0 + N_1 \tag{3}$$

where α denotes the scaling factor, which is positively correlated with $\lceil \log q \rceil$, and N_0 accounts for the noise originating from the acquisition and execution environments. Furthermore, N_1 represents the power consumption associated with the flipping of the remaining bits within the same block, and it can be computed using Eq. 4.

$$N_1 = \sum_{j \neq i}^{P} \alpha * \mathrm{hd}(n_0[j], n_1[j]) \tag{4}$$

In Fig. 1, we present an illustrative example using a message with $n = 16$ and $P = 4$. When focusing on the $m[8]$ bit, we obtain the hypothesized value $h = \alpha * 1 + N_0 + \alpha * 2$. It is worth noting that the noise term N_1 is significantly larger than the useful leakage, making it difficult for an attacker to directly recover $m[i]$ through a side-channel attack.

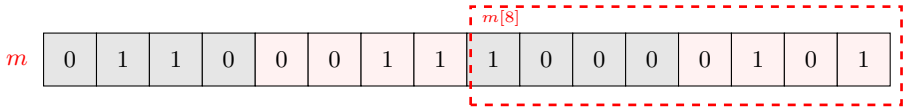


Fig. 1: An example message with $n = 16$ and $P = 4$

In this context, carrying out a message recovery attack utilizing the power consumption information of the target device poses two challenges: 1) In a single nibble, the power consumption associated with encoding each individual bit overlaps, making it nearly impossible to extract information about the specific bit value through a side-channel attack; 2) The power leakage is associated with message bit flipping, and the precise value cannot be restored without knowledge of the adjacent registers' reference state.

To address these challenges, this paper introduces a novel attack method for key recovery in hardware implementations of lattice-based KEMs, which is based on the proposed power model. This attack method focuses on retrieving the message obtained from decapsulating specific ciphertexts in the hardware implementation, which further leads to the retrieval of the private key.

The success of the attack relies on exploiting chosen ciphertexts to eliminate noise term N_1 in the power model. Additionally, when targeting one nibble in a block, the attack fixes the value in the other nibble, which offers a reference value for the hamming distance calculation. By carefully manipulating these chosen ciphertexts, the attacker can effectively extract sensitive information from the power consumption measurements, enabling the recovery of the private key.

3.2 Key Recovery Attack

In this subsection, we will present the entire key recovery attack process proposed in this paper. Using a common LPR scheme as an example, an overview

of the attack is available in Alg. 2. Furthermore, the attack on a single private key polynomial can be divided into three stages:

Algorithm 2 Key Recovery Attack

Require: Target device $\mathcal{D}$
Ensure: Private key sk
Craft M pairs (k_u, k_v)
for $i = 0$ to M **do**
  Initial $u = \mathbf{0}$
  $u[0] = k_u$
  for $j = 0$ to $2P$ **do**
    $v = \mathbf{0}$
    for $k = 0$ to $n/2P$ **do**
      $v[(k * 2P) + j] = k_v$
    end for
    $\mathcal{CT} = (u, v)$
    CCA-DEC($\mathcal{CT}, \mathcal{D}$)
    $\mathcal{T}$ = PowerMeasure($\mathcal{D}$)
    for $k = 0$ to $n/2P$ **do**
      $m^{(i)}[(k * 2P) + j] = \mathcal{O}(\mathcal{T})$
    end for
  end for
end for
$\mathbf{MSG} = (\mathbf{m}_0, \mathbf{m}_1, ..., \mathbf{m}_M)$
for $i = 0$ to n **do**
  $sk[i]$ = KeyRec($\mathbf{MSG}[i, \cdot]$)
end for

- **Stage 1**: Construct malicious ciphertexts. Choose appropriate ciphertext pairs that enable message recovery attacks under our power model and key recovery attacks through a chosen-ciphertext attack. It is necessary to construct $2P \times M$ sets of ciphertexts according to the attack criteria.
- **Stage 2**: Recover messages corresponding to the chosen ciphertexts. The attacker can transmit each $\mathcal{CT}$ constructed in the previous stage to the target device individually, while passively observing the power information generated during the decapsulation process of the ciphertext. With the help of a suitable oracle $\mathcal{O}$, the attacker can recover the respective message bits at each position. Ultimately, $M \times 2P$ incomplete messages are restored, and then these messages are combined into M complete n-bit messages.
- **Stage 3**: Recover the complete private key based on the obtained messages and their corresponding ciphertexts.

For our attack in MLWE-based schemes, adjusting the structure of $\mathbf{u}$ is sufficient even with multiple polynomials in a private key, as explained in Sec. 4. Further details regarding the attack will be presented in subsequent subsections.

3.3 Multi-Ciphertext Message Recovery Attack

Following the aforementioned attack method, the primary obstacle is retrieving the messages within our power model. As a solution, we suggest a multi-ciphertext message recovery attack (MCMRA) in this subsection. To elaborate, our attack divides a message into $\frac{n}{2P}$ blocks at first. Next, our attack targets the utilization of the chosen ciphertext method to establish a relation between a single bit in a message nibble and its corresponding private key coefficient. All other bits in the same nibble will be fixed to 0. Additionally, we require the adjacent nibble in the same block to solely comprise 0 bits. We refer to this pair of nibbles as a target block. We provide an illustration using an example of a tiny LPR scheme with $n = 16$ and $P = 4$ in Fig. 2. Each target block consists of a gray nibble and a pink nibble. Executing each attack enables the recovery of only certain message bits, such as recovering $m[6]$ and $m[14]$ in one attack, and $m[7]$ and $m[15]$ in the next attack. By executing P message recovery attacks, we are able to retrieve all the message bits in the pink nibble. Similarly, additional P attacks allow us to recover the message bits in the gray nibble. In other words, it is necessary to carry out a total of $2P$ message recovery attacks to recover a message of n bits.

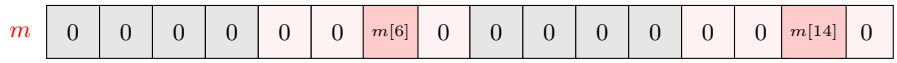


Fig. 2: Structure of message in MCMRA

Such an approach has two implications: Firstly, every target block always contains a nibble filled with 0 bits, which offers a reference value for the other nibble. By counting the number of flipped bits in each attack, the attacker can calculate the hamming distance as needed. The number of flipped bits can be derived from power leakage in accordance with our power model. Furthermore, when targeting a nibble, the attacker can solely concentrate on one particular bit, while disregarding all other bits set to 0. This eliminates the noise term N_1 in Eq. 4.

Based on the above attack, if the attacker has access to an oracle that can determine whether a bit flip has occurred based on power measurements, they could potentially recover that specific message bit in a nibble. Using multiple traces of decapsulating chosen ciphertext as demonstrated in Alg. 2, the whole message can be retrieved. To perform this method, we will construct an oracle to recover a specific bit, the use of the oracle is depicted in Alg. 3. Specifically, for each trace, the attacker divides it based on the relationship between message bits and power samples. The attacker then selects samples of the target bits as $\frac{n}{2P}$ groups of points of interest(PoIs). Each group of PoIs will be used as input of the oracle to recover a target bit. An attacker can choose an appropriate oracle for their specific situation, such as a threshold of power samples (Simple Power Attack), a mean vector and a covariance matrix (Template Attack), or a neural

network model (Deep Learning-based Attack). In the next subsection, we will explain how to acquire such structured messages using chosen ciphertexts.

Algorithm 3 Message Recovery with the Oracle $\mathcal{O}$

Require: a power trace $\mathcal{T}$
Ensure: $m = \{0,1\}^*$
1: $PoIs \leftarrow \mathcal{T}$
2: **for** $i = 0$ to $\frac{n}{2P}$ **do**
3: $\quad m[i] = \mathcal{O}(PoIs[i])$
4: **end for**

3.4 The Principle of Choosing Ciphertext

We have stated the necessary conditions and will now detail how to achieve them in our attack. As demonstrated in Alg. 1, in the decryption phase of LPR schemes, the equation $mp = v - u \cdot s$ is performed. We define $u = \{k_u, 0, 0, 0, ...\}$, a polynomial having only a constant term k_u, and all other coefficients being 0. Therefore, the i-th coefficient of the resulting polynomial mp is calculated as Eq. 5. To achieve the message structure displayed in Fig. 2, the ciphertext structure illustrated in Fig. 3 is required. As a result, the corresponding structure of mp is shown as Fig. 4. Here, X represents $(k_v - k_u * s[i] \mod q)$ and the decoded bit will reveal some information about $s[i]$; Y represents $(0 - k_u * s[i] \mod q)$ and the decoded bit of Y will always be fixed to 0. Subsequently, we will explicate how to determine the values of k_u and k_v to enable a successful key recovery attack.

$$mp[i] = v[i] - k_u * s[i] \mod q \tag{5}$$

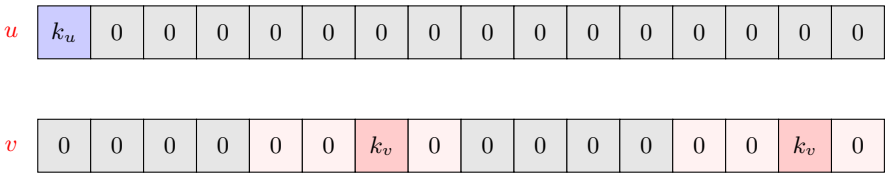


Fig. 3: Structure of (u, v) in MCMRA

Principle of choosing k_u: The decoding process is demonstrated in Eq. 2. In order to guarantee that Decode$(0 - k_u * s[i])$ remains fixed at 0, it is necessary to satisfy that $-\frac{q}{4} < -k_u * s[i] \leq \frac{q}{4} \mod q$. Now that $s[i]$ is within the range of

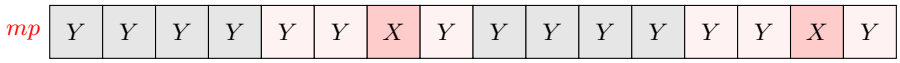


Fig. 4: Structure of mp in MCMRA

$[-\eta, \eta]$, k_u must meet the requirements set in Eq. 6.

$$k_u \in \left(-\frac{q}{4 * \eta}, +\frac{q}{4 * \eta}\right] \mod q \tag{6}$$

Principle of choosing k_v: We have constrained k_u to eliminate the noise term N_1 within our power model. Next, we must choose a suitable k_v to identify the associated private key coefficient based on the retrieved message bits. The chosen value of required k_v varies depending on different parameter sets. An example will be provided in Sec. 4.

3.5 Key Recovery with Imperfect Oracle

When it comes to the side-channel attacks, the oracles are often *not-so-perfect*, unlike the theoretical attacks. As a result, the answers provided by the oracles are not always correct and may have a level of inaccuracy. The error rate from message recovery to key recovery increases multiplicatively because a single secret coefficient relies on message bits at the corresponding position in multiple messages. Consequently, it becomes necessary for a practical attack to address the challenge of achieving key recovery with an imperfect oracle. In this paper, we provide a two-step solution.

The first step is to reduce the impact of message recovery accuracy on key recovery accuracy, achieved by minimizing the number of message bits that rely on a secret coefficient. In 2021, Qin et al. [28] proposed to utilize a binary recovery tree (BRT) to minimize the number of required queries for key mismatch attacks. This technique can also be applied to reduce the number of required power traces [29] for attacks based on plaintext checking. In short, the BRT method uses fewer message bits to determine the coefficients that occur more frequently. Our attack falls into the category of attacks based on message recovery, where the BRT might not significantly reduce the number of required power traces. Nevertheless, we will show that using the BRT technique can significantly improve the accuracy of private key recovery in the presence of imperfect message recovery oracles. This enhancement comes at the expense of a slight increase in the number of required power traces. In Sec. 4, we will provide a specific comparison with other ciphertext-choosing strategies in previous message recovery-based attacks.

The second step involves combining probabilistic results with a lattice reduction attack using the method proposed by Dachman-Soled et al. [10]. As explained in Sec. 2, the a posteriori distribution of secret coefficients obtained from the template attack can be used as perfect hints and approximate hints. By integrating these hints, the target LWE instance becomes easier to solve. It should be noted that direct access to the a posteriori distribution of any secret

coefficient is not feasible in a message recovery-based attack. However, if the attacker employs a template attack or a deep learning-based attack as the message recovery oracle, the a posteriori distribution of the message bits is computed prior to the final prediction, as shown in the distribution table of m in Fig. 5. Subsequently, the distribution of the message bits can be utilized and transformed into the distribution for the corresponding secret coefficients according to the key recovery rule. We apply this method for the first time to a successful attack in our paper, and a detailed description of the transformation will be depicted in Sec. 4. After obtaining the distribution of each secret coefficient, we sort them based on the probability values associated with the candidate having the highest probability for each coefficient. We then set a threshold $1 - \frac{1}{2n}$, following the reference code[4] offered by the original paper [10]. Coefficients with a candidate probability higher than the threshold are considered perfect hints. The mean and variance of the remaining coefficient distributions are calculated and considered approximate hints. After integrating all available hints, we are left with a more solvable uSVP instance. We can estimate its computational complexity and solve it directly using sufficient computational resources.

3.6 The Whole Attack Workflow

At the end of this section, we provide a detailed overview of our attack workflow, which is depicted in Fig. 5. The attack consists of two main phases.

In the first phase, we select random messages and calculate their corresponding ciphertexts using the public key. These ciphertexts are then transmitted to the target device, and power traces are captured during the decapsulation process. With the traces and the corresponding messages, we construct the message recovery oracle for the key recovery attack in Alg. 2.

In the second phase, the MCMRA is employed to retrieve the messages from the chosen ciphertexts. If a perfect oracle is accessible, we can directly recover the private key with the retrieved messages. Otherwise, the a posteriori distribution of message bits will be transformed into the distribution of the corresponding secret coefficients. This is followed by the sorting and integrating procedure. Finally, after integrating all available hints, the remaining lattice instance is solved to obtain the target private key.

4 Case of Study: Kyber768

Kyber is a highly optimized instantiation of an LPR-based CCA-secure KEM [6], it has three security levels (Kyber512/768/1024) that correspond to three parameter sets. For Kyber768, the parameters involved in the attack are $q = 3329$, $n = 256$, $k = 3$, and $\eta = 2$. Unlike general LPR schemes, Kyber employs a modular polynomial multiplication. During the decryption process,

[4] https://github.com/lducas/leaky-LWE-Estimator

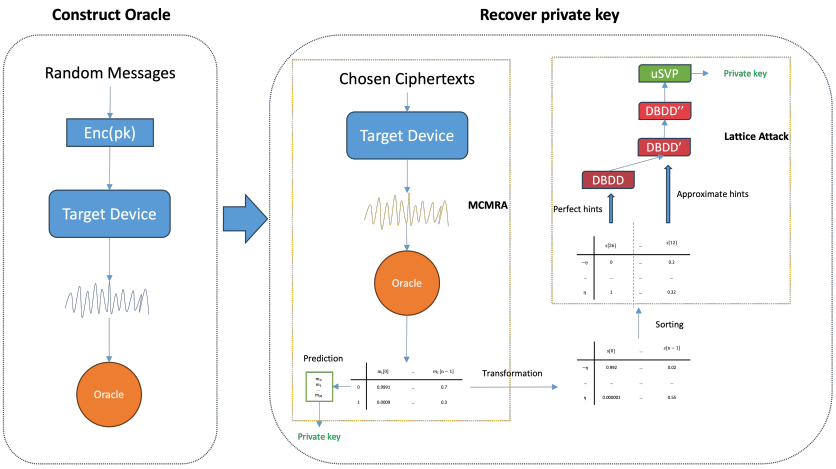


Fig. 5: The whole attack workflow

$mp = v - \mathbf{u} \odot \mathbf{s}$. Here, $\mathbf{u}$ and $\mathbf{s}$ are polynomial vectors containing k polynomials. The cumulative multiplication used is denoted as Eq. 7.

$$\mathbf{u} \odot \mathbf{s} = \sum \mathbf{u}[\mathbf{i}] \cdot \mathbf{s}[\mathbf{i}] \tag{7}$$

Additionally, to reduce the ciphertext size, Kyber applies lossy compression to the coefficients in $\mathbf{u}$ and v, compressing each coefficient to d_u or d_v bits, respectively. A coefficient value is considered *compressible* if its original value can still be correctly obtained after compression and subsequent decompression. The compression and decompression procedure can be found in Eq. 8 and Eq. 9 . The parameter $d_u = 10$ and $d_v = 4$ in Kyber768.

$$\text{Compress}_q(x, d) = \lfloor (2^d/q) * x/q \rceil \mod 2^d \tag{8}$$

$$\text{Decompress}_q(x, d) = \lfloor (q/2^d) * x \rceil \tag{9}$$

In this section, Kyber768 is selected as an instance to demonstrate our attack method, which can also be applied to Kyber with various parameter sets or other LPR-based schemes. We avoid instantiating any specific oracle and instead focus on detailing the process of the chosen ciphertext attack part. Using the BRT method, we choose $k_u = 208$ and determine the value of k_v as depicted in Fig. 6. It is worth noting that the non-leaf nodes correspond to the values of k_v, while the leaf nodes represent the candidates of a secret coefficient [5]. The values of k_u and k_v are all *compressible*, meaning that the attack will not be prevented by ciphertext compression. If the recovered bit is 0 from decapsulating the first ciphertext where $k_v = 832$, then the candidate set $\{-2, -1\}$ is excluded. Next,

[5] The values of k_u and k_v are picked with reference to [29].

the message bit from the ciphertext with $k_v = 1040$ is taken into consideration. Finally, a private coefficient will be determined by message bits at corresponding positions in up to 3 messages. Unlike the plain LPR scheme, the first part of the ciphertext in Kyber is a polynomial vector. According to Eq. 7, when targeting $s[i]$, only $\mathbf{u}[i]$ is set to $\{k_u, 0, 0, ...\}$, whereas the other polynomials contain all 0 coefficients. Consequently, it is necessary to individually recover the k secret polynomials.

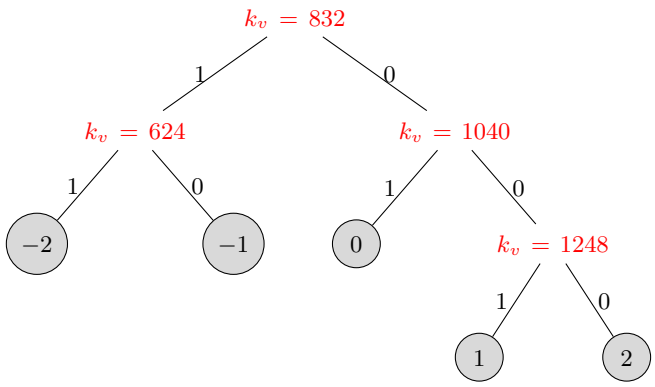


Fig. 6: An optimal BRT for Kyber768

The candidate values of the secret coefficients follow a discrete near-Gaussian distribution, with the highest frequency of occurrence at 0 and decreasing towards both sides. Therefore, the choice of k_v aligns with the principle that candidates with higher frequency have fewer associated message bits. Although this choice does not minimize the number of used traces [30], it enhances the accuracy of key recovery given the same accuracy of message recovery.

To validate the effectiveness of our attack and compare it with previous message recovery-based attacks in [38] and [30], We conducted a simulation experiment whose results are depicted in Fig. 7. It is evident that with a message recovery accuracy of 100%, the private key will be recovered with an accuracy of 100% as well. In addition, our strategy achieves the highest key recovery accuracy when considering the same message recovery accuracy. Furthermore, as the message recovery accuracy decreases, our method exhibits the slowest decline in key recovery accuracy, highlighting its distinct advantage when dealing with an imperfect oracle.

When conducting a lattice reduction attack to recover the complete private key, the a posteriori probability distribution of message bits is initially transformed into the distribution of the corresponding secret coefficients. According to Fig. 6, the probabilities of candidates for a secret coefficient are associated with multiple message bits, which means that we can compute the probability

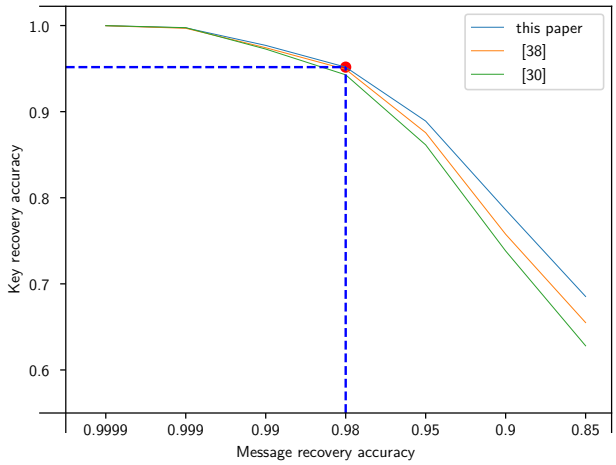


Fig. 7: Key recovery accuracy as the message recovery accuracy changes

of obtaining a coefficient of -2 as Eq. 10. The probabilities for the remaining candidates are also calculated using the same method.

$$P(s[i] = -2) = P(m_0[i] = 1) * P(m_1[i] = 1) \tag{10}$$

Subsequently, the probabilities are normalized by Eq. 11, to ensure that their sum equals to 1. Next, we integrated the distribution of secret coefficients into the DBDD model. Following the procedures outlined in the original paper, we began by integrating perfect hints. Once the perfect hints were exhausted, we proceeded to incorporate approximate hints. Finally, after all available hints are used up, the tool will provide an evaluation of the number of resources required to solve the left uSVP instance, which reflects the level of complexity involved in recovering the entire private key.

$$P_{norm}(s[i] = -2) = \frac{P(s[i] = -2)}{\sum_j P(s[i] = j)} \tag{11}$$

5 Experiments

This section begins with a correlation analysis to verify the correctness of our power model. Under this power model, we will analyze the feasibility of oracles based on a simple power attack and a deep learning-based attack in a practical experimental setup. The cost and accuracy of a message and key recovery attack are then evaluated. Finally, we compare our attack results with those of two previous studies.

5.1 Experimental Setup

We have selected the NAE-CW305 [12] as the target platform for our experiments. This standalone board features a Xilinx Artix-7 FPGA chip. To measure power consumption, a shunt resistor is positioned between the power supply and the FPGA chip. Additionally, we utilize a Chipwhisperer-Lite board to capture power measurements. During each experiment, data and instructions are transmitted from a personal computer, and the target board performs message encoding. The resulting power trace is recorded and transmitted to the computer via the Chipwhisperer-Lite board.

Based on Kyber's reference implementation [4], we developed a four-way parallel implementation of message encoding in Verilog hardware description language as our experimental target. To facilitate more convenient results observation, several null cycles were inserted following each encoding operation. Furthermore, we configured the target board to operate at a frequency of 0.92 MHz and set the sample rate of the Chipwhiperer-Lite to 29.5 MS/s. The complete measurement setup is depicted in Fig. 8.

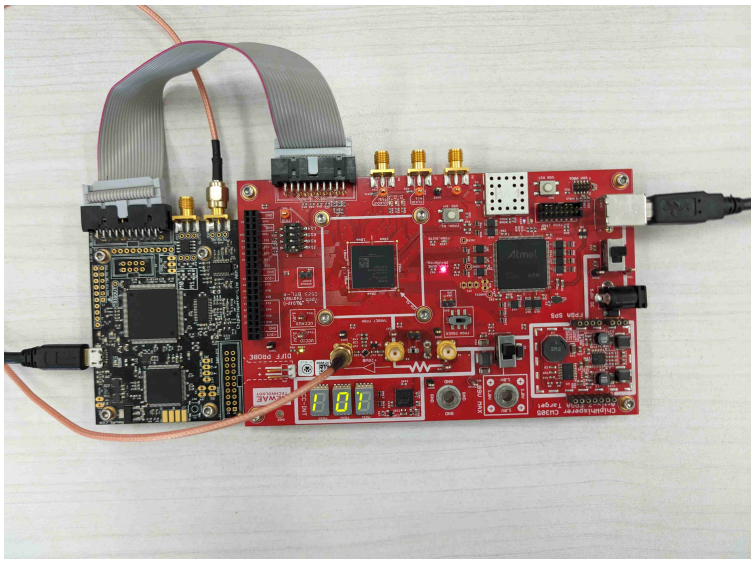


Fig. 8: Measurement setup

5.2 Validation of Power Model

To confirm the presence of the leakage in the target device is identical to our power model, we initiated the encoding of random messages and collected power traces. Based on our power model, the hypothesized value is calculated as the

Hamming distance between the left nibble and the right nibble within the same block. In our experimental setup, each message yields 32 hypothesized values. We transmitted 1000 randomly generated messages to the CW305 board and collected 1000 corresponding power traces. These traces can be represented as a matrix, denoted as $\mathbf{T}$, where each row contains the samples of one trace. The hypothesized values of the messages are represented as a matrix $\mathbf{H}$. Similarly, each row in $\mathbf{H}$ contains the 32 hypothesized values that correspond to each trace.

Next, we computed the correlation between power samples and hypothesized values using the Pearson Correlation Coefficient method, which is illustrated in Eq. 12. The Pearson Correlation Coefficient, denoted by ρ, quantifies the linear relationship between the two variables X and Y. In our experiment, for each column of $\mathbf{H}$, we calculated the correlation matrix $\mathbf{C}[i,j] = \rho(\mathbf{H}[\cdot,i], \mathbf{T}[\cdot,j])$, where $\mathbf{T}[\cdot,j]$ represents the j-th column of $\mathbf{T}$, and $\mathbf{H}[\cdot,i]$ denotes the i-th column of $\mathbf{H}$.

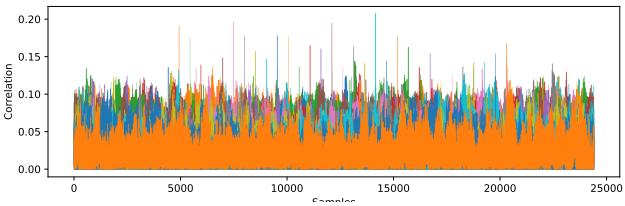
We plot every row vector of $\mathbf{C}$ as illustrated in Fig. 9a. As expected, the correlation analysis reveals a noteworthy leakage. The 32 peaks accurately reflect our model's ability to detect the number of flipped bits between the two nibbles in a block, providing strong support for our power model.

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{12}$$

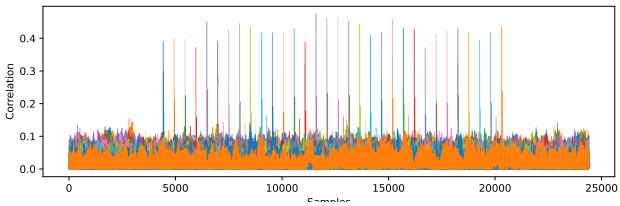
However, despite the presence of correlation peaks, the peak values are relatively insignificant due to the noise of the acquisition and execution environment, i.e., the noise term N_0 in our model. To address this issue, we attempted to preprocess the traces with averaging, and the resulting correlation analysis outcomes are presented in Fig. 9b and Fig. 9c. After performing a basic preprocessing, the power samples exhibit a compelling correlation with our hypothesis. As we increase the number of traces utilized for averaging, the correlation becomes even stronger. This allows us to effectively recover the corresponding Hamming distance using these samples. Furthermore, this indicates that our attack is still highly feasible even in a low signal-to-noise environment.

5.3 Message Recovery Oracle

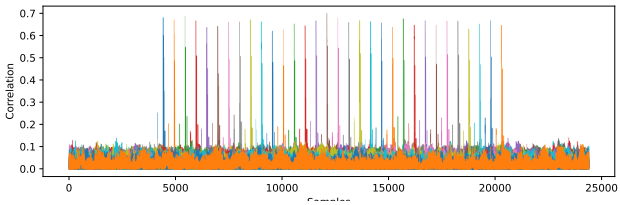
In this subsection, we will instantiate an oracle for the message recovery procedure. Initially, we assumed that relying on a single sample or a limited number of samples is sufficient to serve as a SPA oracle. Thus, we conducted an experiment to assess its feasibility. We choose to analyze the power samples corresponding to the highest and second highest peaks of correlation. These samples are denoted as SP_{h1} and SP_{h2}, respectively. The one-point distribution of SP_{h1} and the two-point distribution of SP_{h1} and SP_{h2} are shown in Fig. 10. {The upper figure consists of a histogram, where the height of each bar indicates the occurrence frequency of the values of samples in SP_{h1}. The red part represents the samples corresponding to bits of 1, while the blue part represents the samples corresponding to bits of 0. The lower figure is a scatter plot illustrating a two-point distribution, where the horizontal coordinate of each point is the value



(a) No averaging



(b) Averaging with 10 traces



(c) Averaging with 100 traces

Fig. 9: Correlation between hypothesized data and power samples

of the sample in SP_{h1} and the vertical coordinate is the value of the sample in SP_{h1}.}. The figure demonstrates a Gaussian distribution of sample values, which aligns with the general noise assumption. However, there is a significant overlap between the samples of bits of 0 and 1, meaning it is nearly unachievable to attain high accuracy in message bit recovery using SPA.

Due to the difficulty of distinguishing between bits of 0 and 1 using SPA, we opted to employ a deep learning-based side channel attack as our message recovery oracle. This type of attack is known to be one of the most powerful side-channel attacks [19]. In our attack, we utilized correlation analysis results to identify the point of highest correlation and then selected 64 points on each side of that position as inputs for our neural network model. To train the binary classification model, we implemented a multilayer perception (MLP) model using the Keras framework [9]. The structure of the MLP is detailed in Tab. 1. Specifically, the input of the model consists of 64 power samples corresponding to the encoding of one message bit. These samples are then passed through two hidden layers, each containing 64 neurons. The output layer produces a 2-dimensional vector that represents the probabilities of the message bit being 0

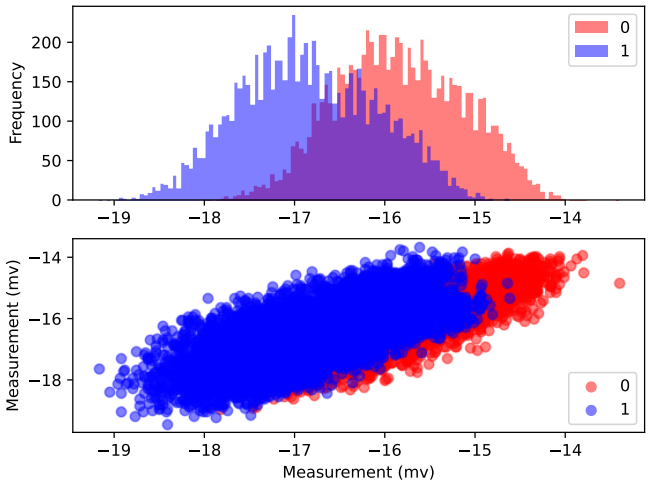


Fig. 10: One/two-point distribution

or 1, calculated through a softmax function. Batch normalization is performed before each hidden layer to speed up the training process.

During the training phase, our attack method involves constructing random messages composed of 32 target blocks. Each block contains a maximum of one '1' bit, with the remaining bits set to '0'. Using the cut-and-join method, we acquire 32×1000 sample groups for training, with labels belonging to the set $\{0, 1\}$. The validation set ratio is set to 0.2 and the batch size is set to 64.

For the training process, we employed RMSProp as the optimizer, and incorporated dropout layers and learning rate decay techniques to mitigate overfitting. The initial learning rate was set to 1E-4. To determine the appropriate number of epochs to conclude the training process, a callback function is utilized. This function ends the training if the training loss fails to decrease in the previous three epochs.

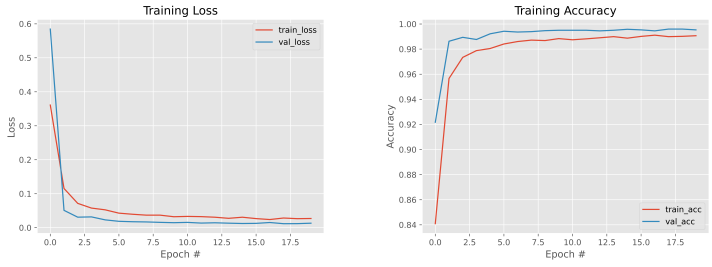
The accuracy and loss on the training and validation sets were tracked during the training process, as shown in Figure Fig. 11. It can be observed that after no more than 10 epochs of training, the accuracy of the validation set consistently stabilized at over 99.6%, while the loss remained at a low level. These results indicate that we have successfully constructed a highly effective side channel oracle for message recovery.

5.4 Key Recovery Attack

After completing the model training, we transmitted the constructed ciphertexts in Sec. 4 to the device and captured corresponding traces. We then utilized our multi-ciphertext message recovery attack to retrieve the required

Table 1: MLP structure in message recovery attack

Layer Type	(Input, Output) Shape
Input	(64,)
Batch Normalization 1	
Dense1	(64, 64)
Batch Normalization 2	
Dense2	(64, 64)
Batch Normalization 3	
Output	(64, 2)



(a) Loss (b) Accuracy

Fig. 11: Training process

four complete messages with the model. This process was repeated 100 times, yielding a message recovery accuracy of 98.67% and a key recovery accuracy of 95%. These results are consistent with the theoretical value presented in Fig. 7.

The 95% key recovery accuracy poses a serious threat to the implementation security. In addition, to further enhance the recovery accuracy, various approaches are suggested, such as employing a majority vote of multiple results or utilizing the ensemble softmax approach proposed in [35]. However, these methods require multiple times of power traces, and even then, successful attacks are not guaranteed. Therefore, we choose to transform the a posteriori distribution of message bits to the distribution of the secret coefficients using multiplication of the probabilities of the determined bits as stated in Sec. 4. After integrating all available hints, we evaluate the left instance and record the related parameters.

After conducting 50 experiments, we obtained a result of 130 bikz by integrating perfect and approximate hints. Furthermore, the computational complexity decreases to 97 bikz with 80 additional extra guesses, respectively, and the attack success probability is decreased to 0.8. The results of a single attack are shown in Fig. 12. It should be noted that the short vector hint is employed,

which is not obtained by side-channel information and falls outside the scope of this paper. Therefore, we omit the details here.

In short, with the side channel information obtained beforehand, the secret can be recovered with a lattice reduction attack of less than 130 (100) bikz. More generally, the computational complexity can be reduced to 38 (30) bits, which is feasible for computation.

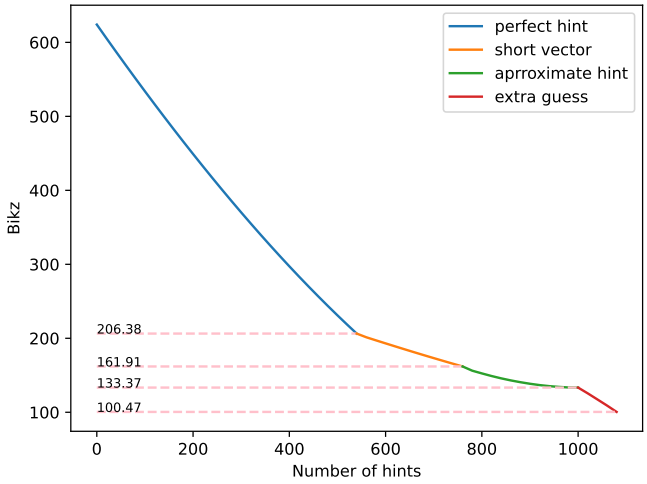


Fig. 12: The lattice attack complexity with integrating the hints

5.5 Comparison

Based on our experimental results, our research has demonstrated significant advantages over existing attacks [17,33] in terms of cost and final outcomes. As shown in Tab. 2, in comparison to the message recovery attack explained in [17], our attack ensures a message recovery accuracy that is nearly 100%, allowing for the possibility of performing chosen-ciphertext attack to recover the private key. Compared to the correlation electromagnetic analysis implemented in [33], our attack phase requires significantly fewer power traces to recover the private key. Additionally, thanks to our ciphertext-choosing strategy, our attack does not rely on any prior knowledge of the initial reference value of the registers. Furthermore, identical to the two attacks, our attack does not require a profiled device, even when a profiled attack is applied as the oracle. This is because the profiling process does not require any information about the secret key, allowing us to complete the process on the target device itself.

Furthermore, in the next section, we will demonstrate the effectiveness of our attack against multiple countermeasures, which cannot be claimed by the previous two attack methods.

Table 2: Comparison between this paper and previous attacks

	Message Recovery		Key Recovery	
	number of traces	search space	number of traces	complexity
[17]	$256 * 5$	2^{64}	/	/
[33]	/	/	$166,620$	$\approx 2^{37}$
this paper	$8(\times 100)$	/	$32 \times k(\times 100)$	$\approx 2^{38}$

6 Conlusion

In this paper, we evaluate the side-channel security of lattice-based KEMs. Based on our proposed power model, we have shown that the multi-ciphertext message recovery attack can overcome the obstacle posed by parallel implementation and retrieve entire messages from a limited number of power traces. To recover the private key, a suitable principle for choosing ciphertext and a lattice reduction technique are utilized, which is useful even in the presence of an imperfect oracle. Our attack was experimentally validated on a Kyber hardware implementation. In our experiments, the accuracy of message recovery is higher than 99%, and recovering the complete private key using a lattice reduction attack requires only a computational complexity of 38 bits. The experimental results indicate that our proposed power model is accurate and our attack can recover the private key at a reasonable cost.

6.1 Discussion of Possible Countermeasures

Shuffling and masking are the common countermeasures to mitigate side-channel analysis vulnerabilities [26]. We will discuss that if our attack remains effective despite the presence of these countermeasures.

Shuffling is a well-known countermeasure against side-channel analysis. In general, a modernized version of the Fisher-Yates algorithm [13] is used to generate a random permutation of a finite sequence. In this scenario, the specific message structure we established is disturbed, preventing the attacker from recovering $\frac{n}{2P}$ bits through one trace. However, the attacker can still employ a similar method to assign 255 bits a fixed value of 0 with only one bit associated with the private coefficient in the matching position. Consequently, the attacker can retrieve one bit per trace. By using $2P$ times the power traces, the attacker can still recover the private key despite the shuffling countermeasure presented.

Masking is another well-known countermeasure against side-channel analysis. First-order masking divides each message bit $m[i]$ into two shares, $m_0[i]$

and $m_1[i]$, such that $m[i] = m_0[i] \oplus m_1[i]$. In our attack setting, a 0 bit will be divided into either $0 \oplus 0$ or $1 \oplus 1$, whereas an 1 bit will be divided into either $1 \oplus 0$ or $0 \oplus 1$. This means that if we determine the flip of an even number of bits by the power traces, it means that the target bit is 0. Otherwise, the target bit is 1. The experiments detailed in Sec. 5 have demonstrated that the attacker is able to determine how many bits have been flipped, which means there is no additional requirement for our attack to successfully recover the private key of a first-order mask implementation.

A viable countermeasure would be to use masking implementations that handle different shares in separate cycles. In one cycle, the encoder would encode the first shares of the P message bits, and in another cycle, it would encode the left part. Thereby, there is no useful hamming distance leakage for the attack. However, such masking implementations would lower the throughput and require additional registers to store the temporary results until all the shares have been encoded.

6.2 Future Work

The proposed attack demonstrates that current hardware implementations of lattice-based schemes are susceptible to side-channel attacks, and new countermeasures are necessary. Therefore, future work should focus on developing secure hardware implementations and conducting additional analyses on other components.

References

1. Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wunderer, T.: Estimate All the {LWE, NTRU} Schemes! In: Catalano, D., De Prisco, R. (eds.) Security and Cryptography for Networks, vol. 11035, pp. 351–367. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_19
2. Alkim, E., Avanzi, R., Bos, J., Ducas, L., Albrecht, M.R., Orsini, E., Osheter, V., Paterson, K.G., Peer, G., Smart, N.P.: NewHope: Algorithm Specifications and Supporting Documentation p. 48
3. Amiet, D., Curiger, A., Leuenberger, L., Zbinden, P.: Defeating NewHope with a Single Trace. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography. pp. 189–205. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-44223-1_11
4. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: Kyber-round3 p. 43
5. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehle, D.: CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367 (Apr 2018). https://doi.org/10.1109/EuroSP.2018.00032
6. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS – Kyber: A CCA-secure module-lattice-based KEM. Tech. Rep. 634 (2017)

7. Bos, J.W., Friedberger, S., Martinoli, M., Oswald, E., Stam, M.: Assessing the Feasibility of Single Trace Power Analysis of Frodo. Tech. Rep. 687 (2018)
8. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better Lattice Security Estimates. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. pp. 1–20. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_1
9. Chollet, F., et al.: Keras. https://keras.io (2015)
10. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with Side Information: Attacks and Concrete Security Estimation. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. vol. 12171, pp. 329–358. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_12
11. D'Anvers, J.P., Karmakar, A., Roy, S.S., Vercauteren, F.: Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. Tech. Rep. 230 (2018)
12. Dewar, A., Thibault, J.P., O'Flynn, C.: NAEAN0010: Power Analysis on FPGA Implementation of AES Using CW305 & ChipWhisperer R
13. Fisher, R.A., Yates, F.: Statistical Tables for Biological, Agricultural and Medical Research, Edited by R.A. Fisher and F. Yates. 6th Ed. Edinburgh: Oliver and Boyd (1963)
14. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. Journal of Cryptology **26**(1), 80–101 (Jan 2013). https://doi.org/10.1007/s00145-011-9114-1
15. Guo, Q., Johansson, T., Nilsson, A.: A Key-Recovery Timing Attack on Post-quantum Primitives Using the Fujisaki-Okamoto Transformation and Its Application on FrodoKEM. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 359–386. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_13
16. Hamburg, M., Hermelink, J., Primas, R., Samardjiska, S., Schamberger, T., Streit, S., Strieder, E., van Vredendaal, C.: Chosen Ciphertext k-Trace Attacks on Masked CCA2 Secure Kyber. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 88–113 (Aug 2021). https://doi.org/10.46586/tches.v2021.i4.88-113
17. Ji, Y., Wang, R., Ngo, K., Dubrova, E., Backlund, L.: A Side-Channel Attack on a Hardware Implementation of CRYSTALS-Kyber. Tech. Rep. 1452 (2022)
18. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. Journal of the ACM **60**(6), 1–35 (Nov 2013). https://doi.org/10.1145/2535925
19. Maghrebi, H., Portigliatti, T., Prouff, E.: Breaking Cryptographic Implementations Using Deep Learning Techniques. In: Carlet, C., Hasan, M.A., Saraswat, V. (eds.) Security, Privacy, and Applied Cryptography Engineering. pp. 3–26. Lecture Notes in Computer Science, Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-49445-6_1
20. Moody, D.: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. Tech. Rep. NIST IR 8413, National Institute of Standards and Technology, Gaithersburg, MD (2022). https://doi.org/10.6028/NIST.IR.8413
21. Moody, D., Alagic, G., Apon, D.C., Cooper, D.A., Dang, Q.H., Kelsey, J.M., Liu, Y.K., Miller, C.A., Peralta, R.C., Perlner, R.A., Robinson, A.Y., Smith-Tone, D.C.,

Alperin-Sheriff, J.: Status report on the second round of the NIST post-quantum cryptography standardization process. Tech. Rep. NIST IR 8309, National Institute of Standards and Technology, Gaithersburg, MD (Jul 2020). https://doi.org/10.6028/NIST.IR.8309
22. Mujdei, C., Wouters, L., Karmakar, A., Beckers, A., Mera, J.M.B., Verbauwhede, I.: Side-Channel Analysis of Lattice-Based Post-Quantum Cryptography: Exploiting Polynomial Multiplication. ACM Transactions on Embedded Computing Systems (Nov 2022). https://doi.org/10.1145/3569420
23. Ngo, K., Dubrova, E., Guo, Q., Johansson, T.: A Side-Channel Attack on a Masked IND-CCA Secure Saber KEM Implementation. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 676–707 (Aug 2021). https://doi.org/10.46586/tches.v2021.i4.676-707
24. Ngo, K., Dubrova, E., Johansson, T.: Breaking Masked and Shuffled CCA Secure Saber KEM by Power Analysis. In: Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security. pp. 51–61. Association for Computing Machinery, New York, NY, USA (Nov 2021)
25. Ngo, K., Wang, R., Dubrova, E., Paulsrud, N.: Side-Channel Attacks on Lattice-Based KEMs Are Not Prevented by Higher-Order Masking. Cryptology ePrint Archive (2022)
26. Oder, T., Schneider, T., Pöppelmann, T., Güneysu, T.: Practical CCA2-Secure and Masked Ring-LWE Implementation. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 142–174 (Feb 2018). https://doi.org/10.13154/tches.v2018.i1.142-174
27. Primas, R., Pessl, P., Mangard, S.: Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2017. pp. 513–533. Lecture Notes in Computer Science, Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_25
28. Qin, Y., Cheng, C., Zhang, X., Pan, Y., Hu, L., Ding, J.: A Systematic Approach and Analysis of Key Mismatch Attacks on Lattice-Based NIST Candidate KEMs. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021. pp. 92–121. Lecture Notes in Computer Science, Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-92068-5_4
29. Rajendran, G., Ravi, P., D'Anvers, J.P., Bhasin, S., Chattopadhyay, A.: Pushing the Limits of Generic Side-Channel Attacks on LWE-based KEMs - Parallel PC Oracle Attacks on Kyber KEM and Beyond. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 418–446 (Mar 2023). https://doi.org/10.46586/tches.v2023.i2.418-446
30. Ravi, P., Bhasin, S., Roy, S.S., Chattopadhyay, A.: On Exploiting Message Leakage in (Few) NIST PQC Candidates for Practical Message Recovery Attacks. IEEE Transactions on Information Forensics and Security **17**, 684–699 (2022). https://doi.org/10.1109/TIFS.2021.3139268
31. Ravi, P., Roy, S.S., Chattopadhyay, A., Bhasin, S.: 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). https://doi.org/10.13154/tches.v2020.i3.307-335
32. Regev, O.: New lattice-based cryptographic constructions. Journal of the ACM **51**(6), 899–942 (Nov 2004). https://doi.org/10.1145/1039488.1039490
33. Rodriguez, R.C., Bruguier, F., Valea, E., Benoit, P.: Correlation Electromagnetic Analysis on an FPGA Implementation of CRYSTALS-Kyber. Tech. Rep. 1361 (2022)

34. Sim, B.Y., Kwon, J., Lee, J., Kim, I.J., Lee, T.H., Han, J., Yoon, H., Cho, J., Han, D.G.: Single-Trace Attacks on Message Encoding in Lattice-Based KEMs. IEEE Access **8**, 183175–183191 (2020). https://doi.org/10.1109/ACCESS.2020.3029521
35. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of Re-encryption: A Generic Power/EM Analysis on Post-Quantum KEMs. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 296–322 (2022). https://doi.org/10.46586/tches.v2022.i1.296-322
36. Wang, J., Cao, W., Chen, H., Li, H.: Practical Side-Channel Attack on Message Encoding in Masked Kyber. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 882–889. IEEE, Wuhan, China (Dec 2022). https://doi.org/10.1109/TrustCom56396.2022.00122
37. Xing, Y., Li, S.: A Compact Hardware Implementation of CCA-Secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 328–356 (Feb 2021). https://doi.org/10.46586/tches.v2021.i2.328-356
38. Xu, Z., Pemberton, O., Roy, S.S., Oswald, D., Yao, W., Zheng, Z.: Magnifying Side-Channel Leakage of Lattice-Based Cryptosystems With Chosen Ciphertexts: The Case Study of Kyber. IEEE Transactions on Computers **71**(9), 2163–2176 (Sep 2022). https://doi.org/10.1109/TC.2021.3122997
39. Zhang, N., Yang, B., Chen, C., Yin, S., Wei, S., Liu, L.: Highly Efficient Architecture of NewHope-NIST on FPGA using Low-Complexity NTT/INTT. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 49–72 (Mar 2020). https://doi.org/10.13154/tches.v2020.i2.49-72
40. Zijlstra, T., Bigou, K., Tisserand, A.: FPGA Implementation and Comparison of Protections Against SCAs for RLWE. In: Hao, F., Ruj, S., Sen Gupta, S. (eds.) Progress in Cryptology – INDOCRYPT 2019. pp. 535–555. Lecture Notes in Computer Science, Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-35423-7_27