cryptography

MDPI

Article

Investigating CRYSTALS-Kyber Vulnerabilities: Attack Analysis and Mitigation

Maksim Iavich * and Tamari Kuchukhidze *

Department of Computer Science, Caucasus University, Tbilisi 0102, Georgia
* Correspondence: miavich@cu.edu.ge (M.I.); tkuchukhidze@cu.edu.ge (T.K.)

Abstract: Significant advancements have been achieved in the field of quantum computing in recent years. If somebody ever creates a sufficiently strong quantum computer, many of the public-key cryptosystems in use today might be compromised. Kyber is a post-quantum encryption technique that depends on lattice problem hardness, and it was recently standardized. Despite extensive testing by the National Institute of Standards and Technology (NIST), new investigations have demonstrated the effectiveness of CRYSTALS-Kyber attacks and their applicability in non-controlled environments. We investigated CRYSTALS-Kyber's susceptibility to side-channel attacks. In the reference implementation of Kyber512, additional functions can be compromised by employing the selected ciphertext. The implementation of the selected ciphertext allows the attacks to succeed. Real-time recovery of the entire secret key is possible for all assaults.

Keywords: post-quantum cryptography; quantum cryptography; side-channel attacks; CRYSTALS-Kyber; masking; deep learning; lattice-based cryptography

Citation: Iavich, M.; Kuchukhidze, T. Investigating CRYSTALS-Kyber Vulnerabilities: Attack Analysis and Mitigation. *Cryptography* **2024**, *8*, 15. https://doi.org/10.3390/cryptography8020015

Academic Editor: Carlo Blundo

Received: 6 March 2024
Revised: 16 April 2024
Accepted: 17 April 2024
Published: 19 April 2024

1. Introduction

Eventually, quantum computing will take off and become more widely used. Post-quantum cryptography, or quantum encryption, is a cryptographic approach for classical computers that can deflect attacks from quantum computers. If computers can utilize quantum mechanics' unique properties, they will be able to do complicated computations far faster than they could with conventional computers [1]. The possibility that a quantum computer may complete some challenging jobs quickly should be evident. The fact that these computations would take several years for a typical computer to complete is noteworthy.

As quantum computing improves, there is rising concern regarding the long-term efficacy of present cryptography approaches. One such technique that is being examined is the well-known public-key cryptosystem RSA. The security of RSA is predicated on challenging mathematical issues like integer factorization. The advent of quantum computing, and in particular techniques such as Shor's algorithm, makes it possible to solve hitherto hard factorization issues. RSA's defense against cryptographic assaults is seriously threatened by this flaw [2]. Another popular cryptographic approach is elliptic curve cryptography (ECC), which is particularly useful in contemporary systems where efficiency and reduced key sizes are essential. The Elliptic Curve Discrete Logarithm Problem (ECDLP), which is likewise thought to be computationally challenging for conventional computers, is the basis for ECC. Elliptic curve cryptography, however, may be broken more quickly by quantum computers than by RSA. ECC becomes more susceptible to assaults when its effective key size is lowered by quantum techniques such as Grover's algorithm. ECC could be even more prone to attack than RSA.

Concerns about the potential obsolescence of conventional encryption techniques are being raised by the advent of quantum computing. This has led to the exploration of novel approaches to data protection, such as lattice-based encryption. These methods are intended to withstand attacks from quantum computers [3].

Aware of this difficulty, post-quantum cryptosystems that can safely and successfully withstand quantum attacks must be developed and put into use [4,5]. With the development of quantum computing, conventional asymmetric methods like RSA could not be adequate to protect private data. The way that quantum technology is developing has prompted an ongoing endeavor to design resilient post-quantum systems [6].

The National Institute of Standards and Technology (NIST) initiated the Post-Quantum Cryptography Standardization Initiative (NIST PQC) in 2016 in response to the changing threat scenario provided by quantum computers. NIST PQC's main objective is to provide strong cryptographic algorithm standards that can withstand attacks from quantum computers. The goal of the project is to secure sensitive data in the post-quantum computing age by requesting, assessing, and standardizing quantum-resistant cryptographic algorithms.

NIST chooses a group of potential algorithms that the cryptography community has submitted to start the process. These candidates were put through extensive testing, with an emphasis on how resilient they were to quantum attacks. The selected primitives are based on linear error-correcting code decoding and lattices, two mathematical issues that are thought to be difficult for quantum computers.

NIST announced in July 2022 that CRYSTALS-Kyber will become the new standard for key setup and public key encryption (PKE) [7]. This is a major development. The reason for this choice is that it has been identified as a key encapsulation mechanism (KEM) that secures IND-CCA2 in models of random oracles that are both classical and quantum. The intricacy of the module learning with errors (M-LWE) problem, which introduces unknown noise into linear equations, forms the basis of CRYSTALS-Kyber's security.

Moreover, CRYSTALS-Kyber has been expeditiously incorporated by the National Security Agency (NSA) into its collection of suggested cryptographic algorithms for national security applications [8]. The algorithm's significance in strengthening cryptographic systems against new quantum threats is highlighted by this acknowledgment.

Known for its IND-CCA2 security, it is undetectable under an adaptively selected ciphertext attack [9]. Because it involves inserting unknown noise into linear equations, the module learning with errors (M-LWE) problem is difficult, which determines its security.

CRYSTALS-Kyber and other post-quantum Public Key Encryption (PKE)/KEM algorithms have weaknesses that have been made public in protected software implementations, despite their theoretical security. Superior side-channel analysis techniques, especially those grounded in deep learning, have been successful in breaching higher-order masked implementations, first-order masked and shuffled software implementations of CRYSTALS-Kyber on a first-order masked and shuffled implementation of Saber on a hardware ARM Cortex-M4. As a result of the discovery of these vulnerabilities, better defenses against side-channel attacks have been created, and CRYSTALS-Kyber implementations after them have improved.

Evaluating the resilience of CRYSTALS-Kyber implementations against side-channel attacks is crucial in light of the vulnerabilities that have been proven. Side-channel attacks take advantage of data gathered via non-primary, physically observable channels, including the timing or power usage of the device executing the application. The security of cryptographic implementations is seriously threatened by these assaults.

Kocher et al. [10] made significant strides in the field by developing Differential Side-Channel Analysis, which made use of differences in physical data. Deep Learning-Based Side-Channel Analysis [11] was another important development that made it possible to launch attacks on a variety of cryptographic systems. Traditional defenses are unable to withstand these onslaughts. Last but not least, Wang et al.'s [12] Error Injection Method breaks difficult targets like hardware implementations of CRYSTALS-Kyber by converting non-differential assaults into differential ones.

Many countermeasures, including masking [13–15], shuffling [16–18], randomized clock [19,20], random delay insertion [21–23], constant-weight encoding [24], and code polymorphism [25,26], are used to lessen side-channel assaults. By preventing information from leaking through physically quantifiable channels like time [27,28], power consump-

tion [29,30], or electromagnetic radiation [31,32], these countermeasures seek to safeguard cryptographic systems.

In conclusion, side-channel attacks are becoming more sophisticated, which emphasizes the significance of continuously evaluating and improving the security of cryptographic implementations—especially when it comes to post-quantum cryptography algorithms like CRYSTALS-Kyber.

We examine the field of post-quantum cryptography, concentrating on the Kyber cryptographic algorithm. First, we give a brief introduction to post-quantum cryptography and stress the need to switch to quantum-resistant encryption techniques. The basics of the Kyber algorithm are covered in Section 2, along with a discussion of its applicability for post-quantum security and its guiding principles. The threat that side-channel attacks offer to cryptographic implementations is next examined in Section 3, emphasizing the necessity of strong countermeasures. Our contribution here is to give a full grasp of Kyber's theoretical foundations, allowing both researchers and practitioners to assess its efficacy in real-world implementations. In Section 4, we provide masking strategies as a potentially effective defense against side-channel assaults, especially when utilizing Kyber.

We examine and evaluate the performance and security aspects of the most recent Kyber implementations in Section 5. Here, we offer an assessment of the advantages and disadvantages of current implementations, which will be helpful to academics and developers who want to apply Kyber in practical applications. Expanding our contribution by pointing out potential areas for development and future lines of inquiry. In Section 6, possible vulnerabilities are discussed, and known attacks against CRYSTALS-Kyber are investigated. Section 7 provides an overview of many defense strategies against these assaults and closes with thoughts on post-quantum cryptography's future and upcoming difficulties. We contribute by summarizing the main conclusions of our investigation and providing tactical suggestions for the development of post-quantum cryptography. Section 8 concludes the paper.

2. Kyber and CRYSTALS-Kyber Overview

Based on the difficulty of solving the learning-with-errors (LWE) problem over module lattices, Kyber is an IND-CCA2-secure key encapsulation mechanism (KEM). Among the contenders for the NIST post-quantum cryptography project is Kyber. Three distinct parameter sets, each targeting a different security level, are listed in the proposal. Kyber-512, for example, seeks security that is roughly equal to that of AES-128, Kyber-768 is roughly equal to that of AES-192, and Kyber-1024 is roughly equal to that of AES-256.

It is recommended to employ Kyber in a hybrid mode, integrating it with well-known "pre-quantum" security procedures. Combining Diffie–Hellman with an elliptic curve is one particular example given. This method makes use of the advantages of both post-quantum and classical cryptography [33].

It is advised to utilize the Kyber-768 parameter set in particular. This parameter selection offers more than 128 bits of security against all known conventional and quantum attacks, according to a very conservative analysis that informed this decision. In the world of cryptography, 128 bits of security are regarded as extremely strong and resilient to both known and unknown threats.

NIST, the US National Institute of Standards and Technology, has chosen a post-quantum cryptography (PQC)-based candidate proposal for CRYSTALS-Kyber, a novel quantum-safe key encapsulation technique, for standardization in the summer of 2022. The acronym CRYSTALS refers to the Cryptographic Suite for Algebraic Lattices.

Kyber is a CCA-secure KEM scheme that is part of CRYSTALS-Kyber. Built atop the selected plaintext attack (CPA) secure PKE technique, Kyber, is CCAKEM.CPAPKE (Figures 1 and 2) uses an adjusted version of the Fujisaki–Okamoto (FO) transform [34].

CPAPKE.KeyGen()

seed $_A \leftarrow \mathcal{U}(\{0,1\}^{256})$

$A \leftarrow \mathcal{U}\left(R_q^{k\times k}; seed_A\right)$
$s \leftarrow \mathcal{B}_{\eta_1}\left(R_q^{k\times 1}\right)$
$e \leftarrow \mathcal{B}_{\eta_1}\left(R_q^{k\times 1}\right)$
$b = As + e_{p_1}$
$pk = (seed_A, b), sk = s$
return (pk, sk)

CPAPKE. Enc$(\boldsymbol{pk} = (\boldsymbol{seed_A}, \boldsymbol{b}), \boldsymbol{m}, \boldsymbol{r})$

$A \leftarrow \mathcal{U}\left(R_q^{k\times k}; seed_A\right)$
$s' \leftarrow \mathcal{B}_{\eta_1}\left(R_\eta^{k\times 1}; r\right)$
$e' \leftarrow \mathcal{B}_{\eta_2}\left(R_q^{k\times 1}; r\right)$
$e'' \leftarrow \mathcal{B}_{\eta_2}\left(R_q^{1\times 1}; r\right)$
$u = \left\lfloor (As' + e') \cdot 2^{d_u}/q \right\rceil$
$v = \left\lfloor (b \cdot s' + e'' + encode(m)) \cdot 2^{d_v}/q \right\rceil$
return $c = (u, v)$

CPAPKE.Dec(s, $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$**)**

$y = \left\lfloor v \cdot q/2^{d_v} \right\rceil - s\left\lfloor u \cdot q/2^{d_u} \right\rceil$
$m' = decode(y)$
return m'

Figure 1. CCAPKE algorithms.

Kyber.KeyGen()

$z \leftarrow \mathcal{U}(\{0,1\}^{256})$
$(pk, \mathrm{s}) = \mathrm{CPAPKE.KeyGen()}$
$sk = (\mathrm{s}, pk, \mathcal{H}(pk), z)$
return (pk, sk)

Kyber.Encaps $(\boldsymbol{pk})$

$m \leftarrow \mathcal{U}(\{0,1\}^{256})$
$(\hat{K}, r) = \mathcal{G}(m, \mathcal{H}(pk))$
$c = \mathrm{CPAPKE.\,Enc}(pk, m, r)$
$K = \mathrm{KDF}(\hat{K}, \mathcal{H}(c))$
return (c, K)

Kyber.Decaps $(\boldsymbol{sk} = (\mathbf{s}, \boldsymbol{pk}, \boldsymbol{\mathcal{H}(pk)}, \boldsymbol{z}), \boldsymbol{c})$

$m' = \mathrm{CPAPKE.Dec(s,}\ c)$
$(\hat{K}', r') = \mathcal{G}(m', \mathcal{H}(pk))$
$d' = \mathrm{CPAPKE.Enc}\ (pk, m', r')$
if $c = c'$ then
    return $K = \mathrm{KDF}(\hat{K}, \mathcal{H}(c))$
else
    return $K = \mathrm{KDF}\left(z, \mathcal{H}(c)\right)$
end if

Figure 2. CCAKEM algorithms.

CRYSTALS-Kyber employs vectors of ring elements in R_q^k, where k is the rank of the module used to scale the security level. For $k = 2,\ 3$, and 4, there are three different

variants of CRYSTALS—Kyber, Kyber-512, Kyber-768, and Kyber-1024. Since the target implementations support Kyber-512, that version is the main focus. Number-theoretic transform (NTT) is used by CRYSTALS-Kyber to efficiently perform multiplications in R_q.

The value represented by the letter K is the outcome of concatenating a message and public key hash with the hash of the output from the CPAPKE.Enc function using a key derivation function. To put it another way, the encryption key (K) is obtained from extra data about the message and public key, as well as from certain outputs of the encryption function (CPAPKE.Enc).

This method is described in the function definition for Kyber.Encaps. The K value is returned during the encryption process, and the returned K value after the decryption process (Kyber.Decaps) may be the same as during the encryption or a fake value, depending on the assessment of the potentially maliciously created ciphertext (c).

A change is made to the input r for CPAPKE.Enc. It is now the result of a hash of the message and public key rather than an arbitrary value. Improved security is the goal of this modification.

Error-based learning schemes are prone to decryption failures, such as the ones found in Kyber. Adversaries may take advantage of these mistakes to learn personal values. Failures in decryption are more likely to happen if an attacker creates secret vectors and error values that go beyond what is allowed in the CPAPKE.Enc procedure.

By using a modified version of the Fujisaki–Okamoto transform, the Kyber.Encaps and Kyber.Decaps procedures make sure that random secret and error values are generated legally and are verified during the decryption procedure.

The CRYSTALS-Kyber algorithm uses the Fujisaki–Okamoto (FO) transformation to provide CCA2 security. It starts by decrypting the ciphertext using CPA. Next, ciphertext c' is produced when the message is "re-encrypted" using CPA encryption. The program then determines if c' and the public ciphertext c are equal. The algorithm becomes True if $c = c'$ and False otherwise. The generation of the session key K is contingent upon the Boolean outcome. The FO transform is carried out to check for any modifications made by an adversary.

Essentially, this Kyber mechanism guards against adversaries trying to take advantage of holes in the encryption and decryption processes by addressing potential weaknesses associated with decryption failures in Learning with Errors schemes.

3. Side-Channel Attacks

Because of the difficulty of the underlying mathematics, a cryptographic system may appear to be resistant to mathematical assaults, yet it may still be susceptible to side-channel attacks. Side-channel attacks, first identified by Paul Kocher in 1996, make use of data that are disclosed while a cryptographic device is in use. This information that has been released might be in the form of electromagnetic radiation, sound waves, power use, or execution time [35]. Side-channel attacks are a serious risk, particularly for embedded systems that use cryptography. Although a lot of post-quantum cryptography (PQC) contenders are made to withstand straightforward timing assaults, additional side-channel techniques like power and electromagnetic analysis could still be able to penetrate them. Scholars are now examining and mitigating these vulnerabilities; NIST highlights the need to include side-channel resistance in PQC implementations. The goal of this continuing research is to guarantee PQC's resilience to different side-channel attacks.

Scholars have conducted a thorough investigation of how susceptible lattice-based Key Encapsulation Mechanisms (KEMs) are to various side-channel attacks. Notably, side-channel-assisted chosen-ciphertext attacks (CCAs) have been the subject of several investigations. CCAs seek to receive the secret key. These studies explore CCAs for different processes inside lattice-based KEMs [36,37]. These operations include the Fujisaki–Okamoto (FO) transform, message encoding/decoding, inverse Number Theoretic Transform (NTT), and error-correcting codes. Attacks using side channels take advantage of non-primary channels, such as timing or power usage. In order to find vulnerabilities in the electrical

signals generated during cryptographic operations, researchers employed vertical side-channel leakage detection to examine CRYSTALS-Kyber's decryption mechanism.

There are flaws in KYBER-512 that allow an attacker to know the contents of decrypted communications and fully recover the key with some simple queries [38]. They concentrated on elements of the clean and m4 schemes, such as message encoding and the inverse Number Theoretic Transform (NTT). The term "m4 scheme" describes an enhanced application of the Kyber cryptographic algorithm running on an ARM Cortex-M4 embedded CPU, which is a very effective processor. This implementation is included in the library and is called pqm4. Notably, for clean and m4, the secret key could be recovered with just four and eight searches, respectively. Additionally, they suggested methods for message recovery that included cyclic message rotation and targeted message bit flipping [39]. Although these methods necessitated (w + 1) traces when a side-channel Hamming weight classifier was present, they pointed out that applications that included masking and shuffling countermeasures might still be vulnerable. On the other hand, using shuffling and masking for attack-protected implementations required a strong presumption that the attacker could disable the countermeasures in order to produce templates.

In addition, the researchers suggested a message-based key recovery attack that would need six specific ciphertexts. It is crucial to remember that the KYBER-512 noise value was raised and the CRYSTALS-KYBER specification was modified, suggesting that more carefully prepared ciphertexts are now required [40].

4. Masking

Masking will be utilized to shield CRYSTALS-Kyber from side-channel attacks. In order to hide the underlying arithmetic behavior of the cryptographic algorithms, a countermeasure known as masking involves splitting a secret into many partially randomized shares (where fifth-order refers to the secret split five times). We will employ a technique called masking to fortify CRYSTALS-Kyber against side-channel attacks [41].

A common defense against power and electromagnetic side-channel investigation is masking. Fundamentally, masking entails dividing a hidden value into several shares at random. The algorithm processes these shares independently at each stage, recombining the results to yield the desired result. Working inside the masking domain stops sensitive variable, which depends on x information from leaking out because it is never utilized directly. A sensitive variable x is divided into $\omega + 1$ shares in an ω-order masking, $x = x_1 \circ x_2 \circ \ldots \circ x_{\omega+1}$, so that $x = x_1 \circ x_2 \circ \ldots \circ x_{\omega+1}$. Arithmetic and Boolean masking are the two options available. Depending on the masking technique, "o" might represent different operations. For example, in arithmetic masking, "o" is the arithmetic addition, whereas in Boolean masking, it is the XOR.

The computations avoid involving x directly by carrying out operations on shares independently, which theoretically prevents side-channel information about x from leaking. Every time a share is executed, it is randomly assigned. Randomization is usually accomplished by allocating random masks $x_1, x_2, \ldots, x_\omega$ to ω shares and calculating the final share as $x - (x_1 + x_2 + \ldots + x_\omega)$ for arithmetic masking or $x \oplus x_1 \oplus x_2 \oplus \ldots \oplus x_\omega$ for Boolean masking [42].

5. State of the Art Implementation of Kyber

The post-quantum cryptography algorithm Kyber, which is highly recommended by NIST, has made considerable progress in terms of hardware platform implementations. The goal of recent research has been to improve Kyber's performance and efficiency by using creative hardware designs and optimizations.

Several studies have proposed dedicated hardware accelerators and FPGA implementations [43] tailored for Kyber, aiming to accelerate polynomial operations and modular arithmetic crucial for its encryption and decryption processes.

A prominent instance is the CRYPHTOR architecture (CRYstals Polynomial HW acceleraTOR), which has specific ALUs and memory configurations tailored for Kyber and

Dilithium algorithms [44]. In comparison to software-based methods, CRYPHTOR has been effectively incorporated into 64-bit and 32-bit RISC-V-based systems-on-chip (SoCs), yielding impressive speedups of up to 26 times for Number Theoretic Transform (NTT) operations and up to 140 times for matrix–vector multiplication.

Furthermore, by utilizing the inherent parallelism and reconfigurability of FPGAs, Kyber implementations implemented in FPGAs have shown tremendous performance increases. Significant speedups for polynomial multiplication, modular arithmetic, and other basic operations necessary for Kyber's encryption and decryption procedures have been demonstrated by these implementations.

New methods and optimizations have been investigated in an attempt to improve Kyber implementations' effectiveness and speed. Novel hardware designs and techniques have been developed by researchers to speed up polynomial operations and modular arithmetic, which are essential elements of the Kyber algorithm [45].

One noteworthy breakthrough is the creation of fast hardware designs for polynomial multiplication based on the Number Theoretic Transform (NTT) in CRYSTAL-Kyber and CRYSTAL-Dilithium, utilizing Digital Signal Processing (DSP) approaches [46]. With the dedicated DSP units for modular multiplication, butterfly operations, and Point-Wise Multiplication (PWM) found in these designs, critical route delays are significantly reduced, and area and performance are improved.

Furthermore, efforts have been made to improve the efficiency of Kyber implementations through the investigation of compact instruction set extensions and improved modular multiplication approaches [47]. Kyber may be executed efficiently on devices with limited resources thanks to these strategies, which seek to maximize speed while minimizing the use of hardware resources.

Even though hardware advancements have resulted in performance advantages, security is still the first priority when implementing Kyber. The identification of side-channel attack vulnerabilities by recent research has prompted the investigation of solutions to reduce the associated dangers. Power side-channel information has been exploited, and encryption keys have been extracted from Kyber implementations using machine learning techniques [48]. Researchers have suggested recursive learning techniques and disguised implementations to counter these dangers and improve security against side-channel attacks.

In addition, new developments in cryptography have been studied to fortify Kyber's security against possible intrusions, including masked polynomial operations and improved key derivation procedures [49].

6. Attacks against CRYSTALS-Kyber

NIST has recommended CRYSTALS-Kyber as one of the public-key algorithms for standardizing post-quantum encryption. A side-channel attack has been successfully used by researchers against an algorithm implementation that was previously believed to be resistant to these kinds of attacks. The researchers employed machine learning techniques to take advantage of this side-channel attack, which entails power usage.

With the increased ease of measuring and analyzing computer hardware power usage in recent years, side-channel attacks have grown in importance as a security concern. It is well known that some processor or circuit processes can result in energy fluctuations. These fluctuations can be identified and utilized to deduce details about the system or the data being processed.

The side-channel attack in CRYSTALS-Kyber was successful in exposing details regarding the encryption key. This makes it possible to decrypt the data since the hacker can now determine the key using the information that was disclosed.

Utilizing machine learning to teach the system to take advantage of the side channel allowed for the assault. Given that machine learning is not frequently employed in security research, this is an amazing accomplishment. It serves as a reminder that machine learning

can be abused and that businesses need to be mindful of the possible security threats it may provide.

We should not be overly concerned about the security of the CRYSTALS-Kyber algorithm because this assault does not imply that it is "ruined" or "broken." It seems doubtful that this kind of side-channel assault will be employed in actual attacks. We should be aware of the possible security threats that machine learning may provide, as it may be used to exploit these kinds of attacks. The algorithm remains safe, and corporations should not worry too much about it despite the attack against CRYSTALS-Kyber.

Prior research has utilized artificial intelligence (AI) to breach first-, second-, and third-order masked Kyber implementations. However, it was very hard to break any higher-order masked implementations using conventional AI training and profiling techniques. By employing a new kind of deep learning and rotations on the intercepted message to raise the bits' leakiness and, thus, the likelihood of a successful attack, Dubrova et al. were able to overcome this challenge [50]. The attack was initially presented by Dubrova et al. on a C version of Kyber's first-order masking, whereby masked_poly_frommsg() is extended to include higher-order masking. The power consumption of this method, which is called Kyber's re-encryption phase, will be the subject of discussion.

They go after the stage of decapsulation. Following the extraction of the shared key, it is re-encapsulated in the decapsulation process and checked for tampering against the original ciphertext. The secret, or the predecessor of the shared key, is bit-by-bit stored into a polynomial for this re-encryption process. More specifically, the 256-bit secret must be transformed into a polynomial modulo $q = 3329$ with 256 coefficients, where the i-th coefficient is equal to $(q-1)/2$ in the case when the i-th bit is 1 and 0 in the other case. Although the function seems straightforward, it might be challenging to create a masked version. The problem is that shares that *xor* together to form the secret are the natural method to produce shares of the secret, just as shares that add together to form the intended polynomial are the natural way to share polynomials.

Unlike other research, the AI will use recursive learning throughout the profiling phase. In essence, training a w-order masked implementation involves duplicating the input Batch Normalization layer weights of the model M^{w-1} trained on the $(w-1)$-order masked implementation, then expanding the layer to include an additional share to produce the beginning network M^w. Recursive learning is utilized once $w > 3$, and the AI is taught using a network with a conventional random weight distribution when $w \leq 3$.

Two universal models, M_0^w and M_1^w, are obtained by making use of the cut-and-join training traces byte-wise. These recover the strongest leakage, which is the first and second bits of each message byte. Additionally, message bits "0" and "1" are employed as labels, and the AIs are taught to retrieve the message directly without removing the random masks at each iteration.

The final six bits of each byte are shifted to the locations of the initial two bits after the message is rotated three times, as described in this paper's attacks. In this method, we extract the bit values with a higher probability by utilizing the "leakier" bit locations. As a result, we are able to raise the assault success rate.

The assault stage employs a cyclic rotation approach. This is employed because of the non-uniform distribution of the leakage from masked_poly_frommsg(), which is demonstrated by the 9% discrepancy in the likelihood of a successful recovery between bits 0 and 7. This is also made feasible by the fact that module-LWEs are extensions of ring-LWEs, whose ciphertexts may be changed to rotate their messages cyclically. By rotating the final 6 bits of each byte to the initial 2 bits, the attack rotates the message negacyclically three times by 2 bits. This allows the bits to leak out more information without using an excessive amount of time, in contrast to other cycle approaches.

Manipulating the matching ciphertext allows one to rotate a message. Polynomials in the ring $\mathbb{Z}_q[X]/(X^{256}+1)$ make up a ciphertext $c = (u, v)$ in CRYSTALS-Kyber. A negacyclic rotation of the message may be achieved by multiplying both u and v by an indeterminate X, provided that c is created correctly. Decode (-y) and decode (y) can

evaluate different values, which is why this approach may result in mistakes for specific ciphertexts used in secret key recovery attempts [51,52].

The two shares' portions are looped over by the code. It generates a mask for every bit, which is 0xffff if the bit is 1 and 0 otherwise. If necessary, this mask is then utilized to increase the polynomial share by $(q + 1)/2$. It will need a little more electricity to process a 1. An AI is not needed to determine that this function will leak. It was actually noted in 2016 that this pattern was poor and that there may be a risk of concealed Kyber in 2020. As an appropriate countermeasure, processing many bits at once is one technique to lessen this.

The authors, Dubrova et al., make no claims that this is a radically novel attack. Rather, they enhance the attack's efficacy in two ways: by training the neural network and by figuring out how to make better use of numerous traces by altering the sent ciphertext.

Using an ARM Cortex-M4 CPU with an STM32F415-RGT6 device, a CW308 UFO board, and a CW308T-STM32F4 target board operating at 24 MHz, Dubrova et al. tested the suggested attack. The power consumption is measured at a high 10-bit precision of 24 MHz.

In order to train the neural networks, 150,000 power traces for the decryption of various ciphertexts for the same KEM keypair (with a known shared key) are gathered. For a real-world assault, this is already a little unusual because KEM key pairs for key agreements are ephemeral, meaning they are created and used only once. Long-term KEM key pairs do, however, have several valid applications, including ECH, HPKE, and authentication.

Training is essential since, even when executing the same code, devices from the same make and model might display remarkably varied power traces. Neural networks are trained to attack "shares," which are implementations with different degrees of security. Attacking a five-share implementation is the first step toward a six-share implementation. One-fifth of the 150,000 power traces from a six-share implementation, another one-fifth from a five-share implementation, and so on are required to implement their technique. It does not seem probable that someone would deploy a gadget that lets an attacker change the share numbers. The real assault starts with the authors stating that, in perfect circumstances, they could, with a 0.127% chance, retrieve the shared key from a single power trace of a two-share decapsulation. For single-trace assaults on more than two shares, they do not give any figures.

Side-channel attacks are far more successful when many traces of the same decapsulation are used. By rotating the ciphertext rather than leaving traces of the exact same message, the authors cleverly provide a twist. When four identical traces are rotated, the likelihood of success in comparison to a two-share implementation rises to 78%. At 0.5%, the six-share implementation is still robust nonetheless. Eighty-seven percent of the shared key may be recovered when 20 traces are allowed from the six-share implementation.

It should be noted that 2.5 K messages are chosen at random for each w-order masked implementation. Since each trace contains three 2-bit cyclic message rotations, there are a total of 10 K traces for each message. Without cyclic rotations, the average message recovery probability for a first-order masked implementation with one trace is 0.127%. Cyclic rotations increase this chance to 78.866%. The likelihood is 0.56% with a single trace on a fifth-order masked implementation employing cyclic rotations, 54.53% with three traces, and 87.085% with five traces.

In terms of hardware, it may resemble a smart card in certain ways, but it differs greatly from high-end gadgets like desktop PCs, servers, and cell phones. Even with simply integrated 1 GHz CPUs, simple power analysis side-channel assaults are far more difficult to execute, needing tens of thousands of traces with a high-end oscilloscope placed in close proximity to the processor. This type of physical access to a server offers far better attack vectors; all you need to do is connect the oscilloscope to the memory bus.

Power-side channel assaults are generally regarded as unfeasible, with the exception of extremely sensitive applications. However, throttling may occasionally cause an exceptionally strong power side-channel assault to become a distant timing attack when the planets align. To be clear, this attack is not even close to what is happening.

Furthermore, this attack is not very strong or unexpected, even for certain susceptible applications like smart cards. In practice, it does not matter if a disguised implementation divulges its secrets—it always does. The question is how difficult it is to pull off in real life. Papers like this one assist manufacturers in determining how many countermeasures to use in order to make assaults prohibitively expensive.

7. Countermeasures

Reducing the duration of the application's secret key is the best defense against the majority of existing assaults. The assault would be more difficult the fewer times the secret key is made public. The attacker may only employ the attack of message recovery if a secret key is used just once. However, this may also result in other issues. For instance, it might be required to create a large number of secret keys, or the use of secret keys will be eliminated.

If it were not feasible to repeatedly perform the decapsulation procedure, the attack that was given would not succeed. Limiting how many times the same ciphertext may be decapsulated with the same secret key can help achieve this. It might be required to allow a few repetitions in order to accept random communication errors.

Stronger defenses against power analysis assaults, such as the suggested duplication with the clock randomization approach [53], can be used as an alternative. A main and a dummy cryptographic core are the two identical cores that make up the protected implementation. Although the two cores employ two distinct secret and public key pairs for their respective tasks, they are controlled by two different randomized clocks and receive identical input data. Such a technique has the following advantages over masking: zero clock cycle overhead, immunity to glitches, universal coverage, and higher resilience to repetition assaults.

8. Conclusions

The suggested key encapsulation system, CRYSTALS-Kyber, is confronting increasing difficulties due to advanced side-channel attacks. Current studies reveal weaknesses even in cases with strong security, necessitating ongoing defensive enhancements. Masking and shuffling are two countermeasures that are essential to strengthening cryptographic systems. The need to assess algorithms for both mathematical strength and resilience to outside attacks increases as we approach the post-quantum era.

Instead of totally undermining a new wave of encryption, AI is a useful tool for handling noisy data and identifying their flaws. A power side-channel assault and a straight cryptography breach are very different from one another. Surprisingly, few traces are used for the real assault, but deep learning may make use of extremely noisy traces for training. The lack of practically achievable, straightforward, affordable, and efficient defenses to stop these power side-channel assaults is one of the things that made this discussion so fascinating.

Author Contributions: Conceptualization, M.I.; formal analysis, M.I. and T.K.; methodology, T.K. and M.I.; writing—original draft preparation, T.K.; writing—review and editing, M.I. and T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Shota Rustaveli National Science Foundation of Georgia (SRNSF) [STEM–22-1076].

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study, in the collection, analysis, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Buchmann, J.; Dahmen, E.; Szydlo, M. Hash-based Digital Signature Schemes. In *Post-Quantum Cryptography*; Bernstein, D.J., Buchmann, J., Dahmen, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
2. Chen, L.; Chen, L.; Jordan, S.; Liu, Y.K.; Moody, D.; Peralta, R.; Perlner, R.; Smith-Tone, D. *Report on Post-Quantum Cryptography*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016; Volume 12.
3. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
4. Iavich, M.; Kuchukhidze, T.; Gagnidze, A.; Iashvili, G. Advantages and Challenges of QRNG Integration into Merkle. *Sci. Pract. Cyber Secur. J.* **2020**, *4*, 93–102.
5. Gagnidze, A.; Iavich, M.; Iashvili, G. Novel version of merkle cryptosystem. *Bull. Georgian Natl. Acad. Sci.* **2017**, *11*, 28–33.
6. Iavich, M.; Kuchukhidze, T.; Bocu, R. A Post-Quantum Digital Signature Using Verkle Trees and Lattices. *Symmetry* **2023**, *15*, 2165. [CrossRef]
7. Alagic, G.; Apon, D.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.-K.; Miller, C.; Moody, D.; et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*; US Department of Commerce, NIST: Gaithersburg, MD, USA, 2022.
8. National Security Agency, U.S Department of Defense. Announcing the Commercial National Security Algorithm Suite 2.0. Available online: https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF (accessed on 2 April 2024).
9. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber algorithm specifications and supporting documentation. *NIST PQC Round* **2019**, *2*, 1–43.
10. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
11. Wu, L.; Perin, G.; Picek, S. On the Evaluation of Deep Learning-Based Side-Channel Analysis. In *Constructive Side-Channel Analysis and Secure Design, Proceedings of the COSADE 2022, Leuven, Belgium, 11–12 April 2022*; Balasch, J., O'Flynn, C., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13211. [CrossRef]
12. Wang, R.; Ngo, K.; Dubrova, E. A message recovery attack on LWE/LWR-based PKE/KEMs using amplitude-modulated EM emanations. In Proceedings of the 25th Annual International Conference on Information Security and Cryptology, Seoul, Republic of Korea, 30 November–2 December 2022. Available online: https://eprint.iacr.org/2022/852 (accessed on 4 April 2024).
13. Fritzmann, T.; Van Beirendonck, M.; Basu Roy, D.; Karl, P.; Schamberger, T.; Verbauwhede, I.; Sigl, G. Masked accelerators and instruction set extensions for post-quantum cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *2022*, 414–460. [CrossRef]
14. Gigerl, B.; Primas, R.; Mangard, S. Formal verification of arithmetic masking in hardware and software. In Proceedings of the International Conference on Applied Cryptography and Network Security, Kyoto, Japan, 19–22 June 2023; Springer Nature: Cham, Switzerland, 2023; pp. 3–32.
15. Coron, J.S.; Gérard, F.; Montoya, S.; Zeitoun, R. High-order Polynomial Comparison and Masking Lattice-based Encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2023**, *2023*, 153–192. [CrossRef]
16. 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, Virtual, 19 November 2021; pp. 51–61. [CrossRef]
17. Kairouz, P.; McMahan, B.; Song, S.; Thakkar, O.; Thakurta, A.; Xu, Z. Practical and private (deep) learning without sampling or shuffling. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR. pp. 5213–5225. [CrossRef]
18. Nguyen, T.T.; Trahay, F.; Domke, J.; Drozd, A.; Vatai, E.; Liao, J.; Wahib, M.; Gerofi, B. Why globally re-shuffle? Revisiting data shuffling in large scale deep learning. In Proceedings of the 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Lyon, France, 30 May–3 June 2022; IEEE: New York, NY, USA, 2022; pp. 1085–1096.
19. Brisfors, M.; Moraitis, M.; Dubrova, E. Side-channel attack countermeasures based on clock randomization have a fundamental flaw. *Cryptol. ePrint Arch.* 2022. Available online: https://eprint.iacr.org/2022/1416 (accessed on 4 April 2024).
20. Jayasinghe, D.; Udugama, B.; Parameswaran, S. FPGA Based Countermeasures Against Side channel Attacks on Block Ciphers. In Proceedings of the 28th Asia and South Pacific Design Automation Conference, Tokyo, Japan, 16–19 January 2023; pp. 365–371.
21. Coron, J.-S.; Kizhvatov, I. An efficient method for random delay generation in embedded software. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Lausanne, Switzerland, 6–9 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 156–170.
22. Leplus, G.; Savry, O.; Bossuet, L. Insertion of random delay with context-aware dummy instructions generator in a RISC-V processor. In Proceedings of the 2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean, VA, USA, 27–30 June 2022; IEEE: New York, NY, USA, 2022; pp. 81–84.
23. Xagawa, K.; Ito, A.; Ueno, R.; Takahashi, J.; Homma, N. Fault-injection attacks against NIST's post-quantum cryptography round 3 KEM candidates. In Proceedings of the Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, 6–10 December 2021; Proceedings, Part II 27. Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 33–61.

24. Maghrebi, H.; Servant, V.; Bringer, J. There is wisdom in harnessing the strengths of your enemy: Customized encoding to thwart side-channel attacks. In Proceedings of the Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, 20–23 March 2016; Revised Selected Papers 23. Springer: Berlin/Heidelberg, Germany, 2016; pp. 223–243.
25. Belleville, N.; Couroussé, D.; Heydemann, K.; Charles, H.P. Automated software protection for the masses against side-channel attacks. *ACM Trans. Archit. Code Optim. (TACO)* **2018**, *15*, 1–27. [CrossRef]
26. Kreuzer, K.; Nipkow, T. Verification of NP-Hardness Reduction Functions for Exact Lattice Problems. In *Automated Deduction—CADE 29—29th International Conference on Automated Deduction, Rome, Italy, 1–4 July 2023*; Pientka, B., Tinelli, C., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerand, 2023; Volume 14132. [CrossRef]
27. Wang, Z.; Meng, F.H.; Park, Y.; Eshraghian, J.K.; Lu, W.D. Side-channel attack analysis on in-memory computing architectures. *IEEE Trans. Emerg. Top. Comput.* **2023**, *12*, 109–121. [CrossRef]
28. Moraitis, M.; Ji, Y.; Brisfors, M.; Dubrova, E.; Lindskog, N. Securing CRYSTALS-Kyber in FPGA Using Duplication and Clock Randomization. *IEEE Des. Test*, 2023; *early access*. [CrossRef]
29. Jeon, H.; Xie, J.; Jeon, Y.; Jung, K.J.; Gupta, A.; Chang, W.; Chung, D. Statistical power analysis for designing bulk, single-cell, and spatial transcriptomics experiments: Review, tutorial, and perspectives. *Biomolecules* **2023**, *13*, 221. [CrossRef]
30. Zulberti, L.; Di Matteo, S.; Nannipieri, P.; Saponara, S.; Fanucci, L. A script-based cycle-true verification framework to speed-up hardware and software co-design: Performance evaluation on ecc accelerator use-case. *Electronics* **2022**, *11*, 3704. [CrossRef]
31. Köpf, B.; Dürmuth, M. A provably secure and efficient countermeasure against timing attacks. In Proceedings of the 2009 22nd IEEE Computer Security Foundations Symposium, Port Jefferson, NY, USA, 8–10 July 2009; IEEE: New York, NY, USA, 2009; pp. 324–335.
32. He, J.; Guo, X.; Tehranipoor, M.M.; Vassilev, A.; Jin, Y. EM Side Channels in Hardware Security: Attacks and Defenses. *IEEE Des. Test* **2022**, *39*, 100–111. [CrossRef]
33. Ricci, S.; Dobias, P.; Malina, L.; Hajny, J.; Jedlicka, P. Hybrid Keys in Practice: Combining Classical, Quantum and Post-Quantum Cryptography. *IEEE Access* **2024**, *12*, 23206–23219. [CrossRef]
34. Hofheinz, D.; Hövelmanns, K.; Kiltz, E. A modular analysis of the Fujisaki-Okamoto transformation. In Proceedings of the Theory of Cryptography Conference, Baltimore, MD, USA, 12–15 November 2017; Springer International Publishing: Cham, Switzerland, 2017; pp. 341–371.
35. Kocher, P.C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Proceedings of the Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996; Proceedings 16. Springer: Berlin/Heidelberg, Germany, 1996; pp. 104–113.
36. Ngo, K.; Dubrova, E.; Guo, Q.; Johansson, T. A side-channel attack on a masked IND-CCA secure saber KEM implementation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *2021*, 676–707. [CrossRef]
37. Bhasin, S.; D'Anvers, J.-P.; Heinz, D.; Pöppelmann, T.; Van Beirendonck, M. Attacking and defending masked polynomial comparison for lattice-based cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *2021*, 334–359. [CrossRef]
38. Guo, Q.; Nabokov, D.; Nilsson, A.; Johansson, T. Sca-ldpc: A code-based framework for key-recovery side-channel attacks on post-quantum encryption schemes. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, 4–8 December 2023; Springer Nature: Singapore, 2023; pp. 203–236.
39. 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 Trans. Comput.* **2021**, *71*, 2163–2176. [CrossRef]
40. Ravi, P.; Bhasin, S.; Roy, S.S.; Chattopadhyay, A. Drop by Drop you break the rock-Exploiting generic vulnerabilities in Lattice-based PKE/KEMs using EM-based Physical Attacks. *IACR Cryptol. ePrint Arch.* **2020**, *2020*, 549.
41. Beirendonck, M.V.; D'anvers, J.-P.; Karmakar, A.; Balasch, J.; Verbauwhede, I. A side-channel-resistant implementation of SABER. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **2021**, *17*, 1–26. [CrossRef]
42. Emmanuel, P.; Rivain, M. Masking against side-channel attacks: A formal security proof. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin, Heidelberg, 2013.
43. Bisheh-Niasar, M.; Azarderakhsh, R.; Mozaffari-Kermani, M. Instruction-set accelerated implementation of CRYSTALS-Kyber. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 4648–4659. [CrossRef]
44. Di Matteo, S.; Sarno, I.; Saponara, S. CRYPHTOR: A Memory-Unified NTT-Based Hardware Accelerator for Post-Quantum CRYSTALS Algorithms. *IEEE Access* **2024**, *12*, 25501–25511. [CrossRef]
45. Nguyen, T.H.; Kieu-Do-Nguyen, B.; Pham, C.K.; Hoang, T.T. High-speed NTT Accelerator for CRYSTAL-Kyber and CRYSTAL-Dilithium. *IEEE Access* **2024**, *12*, 34918–34930. [CrossRef]
46. Wang, H.; Zhou, J.; Xing, Z.; Feng, Q.; Zhang, K.; Zheng, K.; Chen, X.; Gui, T.; Li, L.; Zeng, J.; et al. Fast-convergence digital signal processing for coherent PON using digital SCM. *J. Light. Technol.* **2023**, *41*, 4635–4643. [CrossRef]
47. Li, L.; Qin, G.; Yu, Y.; Wang, W. Compact Instruction Set Extensions for Kyber. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *43*, 756–760. [CrossRef]
48. Zhao, Y.; Pan, S.; Ma, H.; Gao, Y.; Song, X.; He, J.; Jin, Y. Side channel security oriented evaluation and protection on hardware implementations of kyber. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 5025–5035. [CrossRef]
49. Kundu, S.; Karmakar, A.; Verbauwhede, I. On the Masking-Friendly Designs for Post-quantum Cryptography. In Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering, Roorkee, India, 14–17 December 2023; Springer Nature: Cham, Switzerland, 2023; pp. 162–184.

50. Dubrova, E.; Ngo, K.; Gärtner, J.; Wang, R. Breaking a fifth-order masked implementation of crystals-kyber by copy-paste. In Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop, Melbourne, VIC, Australia, 10–14 July 2023; pp. 10–20.
51. Azouaoui, M.; Kuzovkova, Y.; Schneider, T.; van Vredendaal, C. Post-Quantum Authenticated Encryption against Chosen-Ciphertext Side-Channel Attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**, *4*, 372–396. [CrossRef]
52. Backlund, L.; Ngo, K.; Gärtner, J.; Dubrova, E. Secret Key Recovery Attack on Masked and Shuffled Implementations of CRYSTALS-Kyber and Saber. In Proceedings of the International Conference on Applied Cryptography and Network Security, Kyoto, Japan, 19–22 June 2023; Springer Nature: Cham, Switzerland, 2023; pp. 159–177.
53. Nikova, S.; Rechberger, C.; Rijmen, V. Threshold implementations against side-channel attacks and glitches. In Proceedings of the International Conference on Information and Communications Security, Raleigh, NC, USA, 4–7 December 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 529–545.