Higher Order Leakage Assessment and Neural Network-based Attack on CRYSTALS-Kyber

Buvana Ganesh, Mosabbah Mushir Ahmed and Alieeldin Mady
Qualcomm Inc., Cork, Ireland

Keywords: Side Channel Attacks, CRYSTALS-Kyber, Leakage Assessment, Deep Learning, Higher Order Masking.

Abstract: To enable the secure deployment of CRYSTALS-Kyber as the National Institute of Standards and Technology (NIST) post-quantum cryptography (PQC) standard for key encapsulation mechanisms (KEM), several attacks have emerged for both the algorithm and its implementations. In this work, a thorough higher order test vector leakage assessment has been performed on open source implementations of CRYSTALS-Kyber. With the traces obtained using the ChipWhisperer framework, the leakage is determined and a template Side Channel Attacks (SCA) is performed with deep learning to successfully uncover the secret key from the first-order masked implementation of CRYSTALS-Kyber. Overall, this work performs a comprehensive leakage assessment and neural network-based SCAs on the masked implementation of CRYSTALS-Kyber.

1 INTRODUCTION

Key Encapsulation Mechanisms are essential for generating a shared secret key between parties, to establish secure peer-to-peer transactions. As the finalized candidate for KEMs in the PQC standardization (PQC, 2023), CRYSTALS-Kyber (Bos et al., 2018) has strong mathematical security, while being compact in size. There are open-source implementations (Heinz et al., 2022; Kannwischer et al., 2020) available for CRYSTALS-Kyber that are embedded device compatible. These implementations are built for specific requisites concerning both security and performance, with a notable emphasis on resilience against side-channel attacks (SCA).

SCAs (Kocher et al., 1999) exploit physically measured information, such as power consumption, electromagnetic radiation, sound emissions, and execution time, during cryptographic operations on hardware. In CRYSTALS-Kyber, a successful SCA (Ravi et al., 2022) implies the shared key recovery using power and electromagnetic analysis, as the secret key can be derived from the shared key, which is the message and subsequently extract the long-term secret key.

Before performing a full-scale SCA, a leakage assessment is conducted to detect potential side channel leakage from the device running the algorithm, as it requires fewer traces and complexity compared to an attack. The assessment points out weaknesses in the algorithm which can be used as target for the attack, if it also involves critical security parameters like the secret key. While there are works demonstrating leakage assessment for the unmasked implementation, no significant study has been conducted for the masked implementation to immediately determine the leakage in mkm4 (Heinz et al., 2022). To rectify this, our work demonstrates a leakage assessment with higher order Test Vector Leakage Assessment (TVLA) (Schneider and Moradi, 2015), as it predicts leakage better than basic t-tests on higher order masking.

In the unmasked implementation, the poly_tomsg component has been found to be one of the viable sources of leakage, as it converts the shared key from polynomial to the message domain as a part of decapsulation in pqm4 (Kannwischer et al., 2020). Several attacks (Chang et al., 2022; Mujdei et al., 2022; Ravi et al., 2020b) have been performed on this component for message and secret key recovery. To protect the attacked components from SCA, masking is one of the most used countermeasures. But currently there are also attacks against the masked components (Backlund et al., 2022; Ngo et al., 2021). In our work, we perform the leakage assessment and then use template attacks, that are enhanced by Deep Learning (DL) (Picek et al., 2023) to attack the message decoding component.

The motivation for this work is to understand the implementation of CRYSTALS-Kyber, develop a

Ganesh, B., Ahmed, M. and Mady, A.
Higher Order Leakage Assessment and Neural Network-based Attack on CRYSTALS-Kyber.
DOI: 10.5220/0012715700003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 373-380
ISBN: 978-989-758-709-2; ISSN: 2184-7711

comprehensive methodology for leakage assessment and deploy a set of experiments that can exploit the leakage. The use of Recurrent Neural Networks (RNN) for SCAs is explored, as RNNs are sensitive to the order of the data supplied. This allows to uncover dependencies distributed across different points in time, making it potent against implementations of higher-order masking schemes with multivariate leakages.

1.1 Contribution

Our work demonstrates that the masked message decoding function is leaky under second order t-tests for fixed vs random keys, and is exploitable even with the first order masking. The rationale in this work is to find a source of leakage before launching an attack on the masked implementation. The part of the leakage in masked message decoding is used to launch a side-channel attack. The main contributions of the paper are given as follow:

- Performed Welch's t-tests on multiple components of the unmasked and masked implementations of CRYSTALS-Kyber.
- Performed second order t-tests on the leaky components of the first order masked implementation (Heinz et al., 2022), which has not been done before in literature.
- Performed a template SCA using NN on the masked implementation of Kyber specifically focused on the masked_poly_tomsg step in the decryption using multi-layer perceptrons and recurrent NNs.

After finding an exploitable vulnerability, an NN-based side-channel attack is executed, specifically targeting the masked_poly_tomsg step in the decryption process of the Kyber algorithm. The multi-layer perceptron (MLP) model retrieves 99% of the secret key from just the attack phase. The model is evaluated based on validation accuracy and key retrieval. The performance of the model was achieved through bespoke pre-processing and the error correcting code.

Our MLP model is similar to the model in (Backlund et al., 2022), where the attack was performed on masked and shuffled Kyber, which is not open source. We target mkm4, that needed additional implementation, such as finding specific triggers according to the leakage from the TVLA. Compared to the attack in (Backlund et al., 2022), our method has less preprocessing, from removing their cut-and-join technique.

The paper is outlined as discussed in this section. Sec. 2 discusses the preliminaries required for the rest of the paper including the CRYSTALS-Kyber algorithm, masking, t-tests and the device set-up for the experiments performed. The results of the TVLA experiments are provided in Sec. 3 and further proceed to perform attacks in Sec. 4. Finally, Sec. 5 concludes and provides some possible future works.

1.2 Related Work

The initial phase preceding any attack involves identifying potential vulnerabilities through statistical analysis, and Welch's t-test serves as a valuable tool for this purpose. While there exists a considerable number of studies on attacks targeting CRYSTALS-Kyber, it is noteworthy that, until recently, TVLA on Kyber have been sparingly explored. The pioneering works of (Rajendran et al., 2023; Ravi et al., 2020b; Sim et al., 2022) are among the few that have delved into TVLA specifically on the poly_tomsg operation of Kyber. Two widely adopted implementations that have become standard for cryptographic attacks are pqm4 (Kannwischer et al., 2020) and mkm4 (Heinz et al., 2022).

Numerous attacks on CRYSTALS-Kyber have been conducted using the official pqm4 implementation (Ravi et al., 2022; Ravi et al., 2020b). Following this, there were several attacks focusing on different components of the implementation like the comparison operation plaintext-checking (PC) oracle (Rajendran et al., 2023), but mostly surrounding the message encoding (Sim et al., 2020), the number theoretic transform (Bock et al., 2024; Primas et al., 2017; Yang et al., 2023) and the poly_tomsg components (Chang et al., 2022). Tab. 1 refers to some relevant attacks on Kyber, especially focusing on the components of Decapsulation.

The landscape of side-channel attacks has witnessed a notable evolution with the integration of NNs, as evidenced in recent literature, as illustrated in (Picek et al., 2023). While there exist pre-trained models for AES and RSA, limited availability is noted for other cryptographic schemes. After the release of the masked implementation mkm4 (Heinz et al., 2022), many attacks leveraged various NN architectures, including multi-layer perceptrons (MLP), convolutional neural networks (CNN), and RNN attacking mkm4.

Ngo et al. introduced profiled attacks with NNs in their work on Saber KEM (Ngo et al., 2021) and extended this approach to generic KEMs in (Ngo et al., 2022; Dubrova et al., 2023). Tab. 2 covers relevant attacks on first-order masking and some of these attacks use NNs to accomplish the attack.

Table 1: Side Channel Attacks on CRYSTALS-Kyber.

Paper	Target	Method used
(Ravi et al., 2020b)	poly_tomsg	Chosen Ciphertext – 2560 traces
(Sim et al., 2020)	Barrett reduction	Chosen Ciphertexts - Clustering
(Chang et al., 2022)	poly_tomsg	Template matching – 900 traces
(Mujdei et al., 2022)	Polynomial Multiplication	Hamming weight - 3329^2 guesses
(Primas et al., 2017)	poly_frommsg	Hamming weight - 500 traces
(Rajendran et al., 2023)	re_encrypt - PC oracle	Template matching - 5520 & 72 traces
(Yang et al., 2023)	KeyGen Multiplication	Template matching – 900 traces

Table 2: Attacks on first order masking and use of AI.

Paper	Target	Method used
(Bock et al., 2024)	Polynomial Multiplication	Template attack - no DL
(Backlund et al., 2022)	masked_poly_tomsg	MLP- Hamming distance
(Ngo et al., 2021)	Saber - poly_A2A	MLP & ECC- Hamming weight
(Ueno et al., 2022)	masked_AES	NN Classification

2 PRELIMINARIES

2.1 CRYSTALS-Kyber

Define the ring $R = \mathbb{Z}$ and $R_q = R/qR = \mathbb{Z}_q$ and $K_{\mathbb{R}}$ with the ring-embedding $\sigma : K \to K_{\mathbb{R}}$, with discrete Gaussian and normal distribution used for sampling the vectors. Let $\{0,1\}$ be I. Consider the functions in message encoding and decoding and reduce the key and ciphertext sizes in R_q, $\mathsf{Compress}(\cdot,d) : \mathbb{Z}_q \to \{0,...,2^d-1\}$ and $\mathsf{Decompress}(\cdot,d) : \{0,...,2^d-1\} \to \mathbb{Z}_q$. These functions constitute the poly_frommsg and poly_tomsg in the implementations, and follow invertibility with negligible error.

Algorithm 1: Kyber PKE - Key Generation.

Input: seeds $\rho, \sigma \in I^{256}$

1: Sample $A \sim R_q^{k \times k}$ using seed ρ
2: Sample $(s,e) \sim I^k \times I^k$, using seed σ
3: $t_{\text{comp}} := \mathsf{Compress}(As + e, d_t)$
4: **return** pk $:= (t_{\text{comp}}, \rho)$, sk $:= s$

The Fujisaki-Okamoto (FO) transform is used to upgrade the security of the underlying public key encryption (PKE) in CRYSTALS-Kyber (Bos et al., 2018) from the weaker indistinguishability under Chosen Plaintext Attacks, to the stronger adaptive Chosen Ciphertext Attacks for KEM, by removing ciphertext malleability in the PKE. The modified FO transform re-encrypts the decrypted message and compares the resulting ciphertext against the received one.

Algorithm 2: Kyber PKE - Encryption.

Input: pk, $m \in I^{256}$, $\tau \in I^{256}$

1: $t := \mathsf{Decompress}(t_{\text{comp}}, d_t)$
2: Sample $A \sim R_q^{k \times k}$, for seed ρ
3: Sample $(r, e_1, e_2) \sim I^k \times I^k \times I$, for seed τ
4: $u_{\text{comp}} := \mathsf{Compress}(A^T r + e_1, d_u)$
5: $v_{\text{comp}} := \mathsf{Compress}(t^T r + e_2 + \lfloor \frac{q}{2} \rceil m, d_v)$
6: **return** $c := (u_{\text{comp}}, v_{\text{comp}})$

Algorithm 3: Kyber PKE - Decryption.

Input: $c = (u_{\text{comp}}, v_{\text{comp}})$, s

1: $u := \mathsf{Decompress}(u_{\text{comp}}, d_u)$
2: $v := \mathsf{Decompress}(v_{\text{comp}}, d_v)$
3: **return** $m := \mathsf{Compress}(v - s^T u, 1)$

The pqm4 library serves as a benchmarking and testing framework that targets the ARM Cortex-M4 family of microcontrollers and supports all versions of Kyber. The masking countermeasure, introduced in (Chari et al., 1999), serves as a strategy to conceal side channel leakage for dividing data into multiple shares that can be processed independently, yet when combined, represent the final processed data, so that no individual share reveals any information about the masked secret. Previous research has demonstrated that, through DL, individual secret shares can be extracted and subsequently combined (Backlund et al., 2022; Ueno et al., 2022). The mkm4 library is built upon the M4 implementation of pqm4. The masking is available for the functions that were attacked in pqm4 like message encode and decode, polynomial multiplication and polynomial comparison. mkm4 only supports Kyber 768, which we use from both the

libraries for our attacks. Alg. 3 presents only the Kyber PKE, as given in (Bos et al., 2018) as it covers the necessary background for our attack.

In the decryption and eventually the decapsulation function, one of the important components is the decoding function wherein the polynomial gets converted into the message again, where the message is the shared secret, which are of interest to attack.

2.2 Welch's t-test and TVLA

The Welch's t-test (Welch, 1947) is useful when dealing with unequal sample sizes or variances, enables the comparison of means between the two groups. For SCAs, this helps point out any leakage between the two sets that can be considered for exploitation. Consider having n_A samples from set A and n_B samples from set B. For TVLA typically, set A contains fixed traces, i.e., traces obtained for a function where the keys are fixed and the inputs are random, and set B contains random traces where keys and messages are random. For each group $j = A, B$, let μ_j represent the sample mean in group j, and s_j^2 denote the sample variance.

$$|t| = \frac{|\mu_A - \mu_B|}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}} \quad (1)$$

In the context of masked implementations of any order, the detection of leakage necessitates higher-order TVLA, as articulated in (Schneider and Moradi, 2015). Various techniques can be employed for this purpose, including applying the t-test with higher-order central moments and, the χ^2 test, among others. An essential adaptation involves replacing variance with the order of the moment, aligning with the order of the mask, which may include measures such as skewness and kurtosis. This nuanced approach is imperative for unveiling and addressing potential vulnerabilities in masked implementations, contributing to a comprehensive evaluation of security in cryptographic systems.

3 LEAKAGE ASSESSMENT

First, the leakage assessment is performed with standard t-test and the second order t-test on both the implementations of CRYSTALS-Kyber (Heinz et al., 2022; Kannwischer et al., 2020) to observe the behaviour of the components. Here, key generation or encapsulation procedures are not considered because based on the history of attacks (Ravi et al., 2022), most attacks tend to focus on the decapsulation algorithm to retrieve the secret key, either directly or through the shared secret, i.e., the message. It is worth noting that higher order TVLA has not been employed to assess leakage in masked implementations for PQC so far and our work rectifies this.

3.1 Device Set-up

All experiments were conducted on a Ryzen-7 laptop equipped with 16GB of RAM and approximately 50GB of virtual RAM. The power consumption-based attacks were facilitated using ChipWhisperer Lite, a well-established tool in the field (O'Flynn and Chen, 2014). This tool captures traces of power consumption by monitoring clock cycles. The experiments target an ARM Cortex-M4 processor, specifically utilizing either the STM32F303 or STM32F415, which is commonly employed for trace collection. The F4 series, notable for its inclusion of a True Random Number Generator, is also utilized.

3.2 TVLA

The critical security parameters for Kyber include the secret key or the shared secret, representing the sensitive core of cryptographic systems. The construction of PQC algorithms can also vary across different implementations, urging the adaptation of standards to accommodate these nuances. For each of the experiments, 1000 traces were considered per operation to perform TVLA. Though it would typically require more traces for considering a significant leakage, but a noticeable leakage is observed with such a low trace numbers also. The public key distinguisher method proposed in (Saarinen, 2022), with the as encryption, aims at assessing whether the public key leaks more data than expected during encryption. This experimental study with `pqm4` revealed no discernible leakage for the t-test calculated as in 2.2. This signifies the resilience of the implementation against unintended information disclosure through the public key during the encryption process.

Random vs Mismatch Traces: For assuring the security of the plaintext-checking oracle, with decapsulation of `pqm4`, random vs mismatch traces were used to understand how the oracle reacts to improper decapsulations. From the setup in Sec. 2.2, set A contains valid ciphertexts, while set B comprises ciphertexts encapsulated with a deliberately mismatched public key, $pk' \neq pk$. The subtraction and the comparison operations of decapsulation exhibit leakage when analyzing such random vs mismatched ciphertexts with TVLA, but it is not significant enough. It should be noted that the masking of the comparison operation has made it harder to attack this part of the

algorithm.

Fixed vs Random Traces: For evaluating decryption, traces within set A are generated using a fixed keypair, whereas set B employs random and unique keypairs for each trace. The creation of ciphertexts involves random messages paired with matching keypairs, strategically targeting different components of the decryption process. The leakage found for this variety is not too evident for all operations in decapsulation but present for the poly_tomsg component in pqm4. Though other operations were tested as well, polynomial multiplication showed less than 5% leakage and the rest of the functions did not show any immediately.

As shown in Fig. 2, the leakage is not enough for performing attacks on the function, as it indicates leakage of a single bit uniformly, which may not be exploitable. Given the apparent leakage in the unmasked implementation, the test is replicated with the masked implementation to gauge the area of leakage. The standard fixed vs random TVLA is performed in the masked mkm4 library, and the leakage is evident for the masked counterpart of the decoding function, masked_poly_tomsg in the beginning of the traces.

3.3 Higher Order TVLA

In literature for the leakage assessments done on Kyber (Rajendran et al., 2023; Ravi et al., 2020b; Sim et al., 2020; Sim et al., 2022), the first-order t-tests have revealed leakage on the poly_tomsg component, corresponding to the message decoding process. It is necessary to confirm whether the problem persists specifically within the masked_poly_tomsg operation in mkm4.

Though there are different methods like the χ^2 test or F-test or bi-variate analysis , our approach involves central moments (Schneider and Moradi, 2015). For the higher order TVLA, the mean and standard deviation in the equation in 2.2 are replaced with higher order central moments. For second order TVLA, the mean μ is substituted with CM_2 and the variance s^2 with $CM_4 - CM_2^2$, where CM_2 is the second order central moment and CM_4 is the fourth order central moment.

$$|t| = \frac{|CM_{2A} - CM_{2B}|}{\sqrt{\frac{CM_4 - CM_{2A}^2}{n_A} + \frac{CM_4 - CM_{2B}^2}{n_B}}} \quad (2)$$

For the fixed vs random TVLA, the set A contains traces of the target operation for the encapsulations with a fixed key and set B for random keys. The aim is to enhance and confirm the findings of the previous first order TVLA but with 1000 traces for practical considerations. The captured traces were each of length 280,000 for the masked_poly_tomsg operation. A subset of 44,000 points in the beginning of the trace can clearly be identified as leaky in Fig. 3. This approach facilitated a more manageable yet insightful examination, revealing critical insights into the nature and extent of the leakage within the masked_poly_tomsg operation. The results indicate that the masked implementation exhibits a reduced susceptibility to leakage for the specific component under examination.

4 ATTACK

The target for performing the attack is the masked_poly_tomsg operation in mkm4 as per in Fig. 3. For neural networks, the two phases are training and prediction, that require the training dataset and the test dataset. First the training data is collected and pre-processed. The cleaned data is then segmented and ordered to make the NN optimal with speed and efficiency, then fed into the NN of choice to be trained. Once the model is trained, the saved model is used to make predictions on the test data. The model is adjusted and retrained based on the prediction results.

Moving to the attack phase, traces were collected specifically for chosen ciphertexts. The conceptualization of the approach for the attack originated with Saber and has been extended to Kyber, as detailed in (Backlund et al., 2022). Their attack focuses on a custom masked and shuffled version of CRYSTALS-Kyber, but only masking is used as countermeasure in this work, as there is no official code for the shuffled component available open source. The ECT is strategically employed to extract the secret key from the message guesses associated with different ciphertexts.

4.1 Profiling Phase

For the mkm4 implementation, 50000 traces are captured for the masked_poly_tomsg of the decapsulation for detailed analysis, for ciphertexts known to the attacker. The trace length is set to the first 53000 out of 280,000 points to utilize the source of the leakage as seen in Fig. 3.

4.1.1 Pre-Processing

First, N traces associated with the masked_poly_tomsg operation are gathered with the ChipWhisperer. The traces are split into five

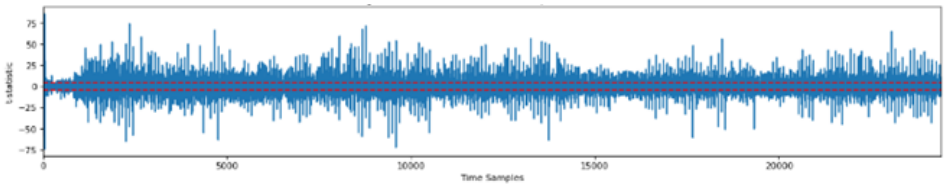

Figure 1: First Order TVLA on poly_tomsg with pqm4.

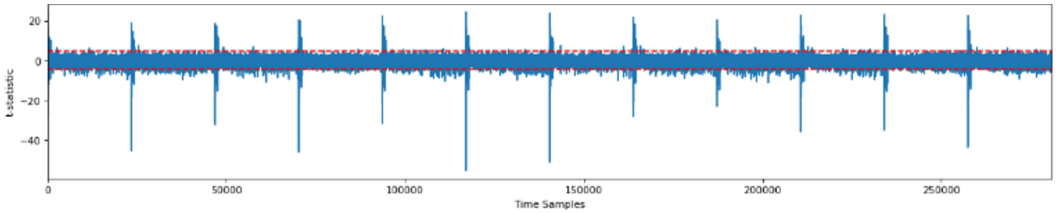

Figure 2: First Order TVLA on masked_poly_tomsg with mkm4.

equal sets. The dataset is carefully prepared for training, with the trace shape set at $(10000, 53000)$. Synchronization is crucial in this process, and a segment length of 225 is chosen. Then, the mean of the end segments for all traces is calculated, and correlations are identified for consistency, resulting in a refined trace shape of $(10000, 57600)$. This is done to ensure a consistent shape for all the traces and eliminate outliers. The subsequent steps involve cutting and shaping the traces, so that they can be split bitwise for better learning in the DL architecture.

Then a unified dataset of 50000 traces is created, combining individual datasets into one to increase the amount of training data. The final trace shape is $(256 * 50000, 3 * 225)$ where the 53000 is split into 256 bit parts (Or 8 bytes). It was decided to keep the three consecutive bit information together as the slicing them would increase the complexity or reduce performance. Standardization and normalization procedures are applied to each trace before the model training phase.

4.1.2 Deep Learning Architecture

We consider a similar architecture to (Backlund et al., 2022) but the algorithm was modified to better fit the mkm4 library. Additionally, the Long Short-Term Memory (LSTM) model, a type of RNN is used as it is known for its proficiency in handling temporal data. Recognizing the importance of trace order and message bit sequencing in key recovery, LSTM is employed in conjunction with batch normalization for standardized traces. The model architecture incorporates ReLU activation for the middle layers and sigmoid for the output layer. The architecture for the RNN is given in Tab. 3. The network can be adjusted with different activation functions and NN layers. There are many factors and metrics like loss, accuracy, etc., that can be calculated in different methods. Setting up different permutations of all such factors is called hyperparametrization and it is very useful in improving the accuracy of a model.

Table 3: Neural Network Architecture.

Layer type	Output shape
Batch Normalization 1	input size
LSTM	512
Batch Normalization 2	32
ReLU	128
Dense 1	16
Batch Normalization 3	16
Sigmoid	16
Dense 2	8

The optimization is carried out using the N-Adam optimizer and RMSprop, with a loss function based on binary cross-entropy. The model undergoes training for 100 epochs, with a batch size set at 128, culminating in a comprehensive evaluation of the proposed attack methodology. The architecture in the paper for RNN, in Table 3, is performed in combination with Dense layers and the activation functions are chosen according to the number of labels to be predicted.

4.2 Attack Phase

The attack phase consists of two parts, first, training the traces and second, using the model to predict the secret key. Constructing the Chosen Ciphertexts (CCT) required for the attack involves the creation of sparse ciphertexts, specifically for a ciphertext (u, v) with a size of 768 bytes as seen in the encryption part of Alg. 3. The computation of $m = v - su$ constructs the message m from the ciphertext and the secret key s. This forms the basis for the construction of CCTs,

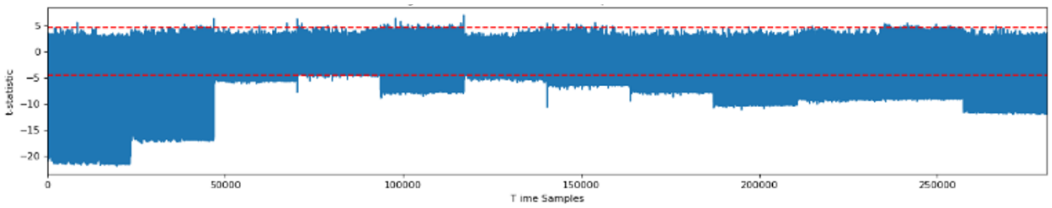

Figure 3: Second Order TVLA on masked_poly_tomsg in mkm4.

wherein the chosen values are determined using Hamming distances with a rotational parameter for ciphertext rotations through shifting (Ravi et al., 2020a).

The enumeration of CCTs is executed in collaboration with the ECT. This tool is very helpful in rounding coefficients that are close enough to the original value in the predicted secret key. The initial step involves generating an error dictionary based on the code distances observed in the chosen ciphertexts. Subsequently, the recovered messages undergo three iterations of error correction through this error dictionary to identify and rectify errors. If no errors are detected after three passes, the corresponding coefficient is deemed enumerable. In conclusion, these coefficients are systematically enumerated over $q = 3329$, facilitating the recovery of the correct secret key coefficient.

4.3 Results

The neural networks were trained with both the trimmed and untrimmed data, but as one can expect, the bitwise trimmed data performs better than the untrimmed data. In current setup, the attacks have not yielded success with the RNN architecture, even though RNN is built to perform better on time-series data. The training phase witnessed a decline in loss rates, indicative of challenges in convergence, and highlighted the dependency on significantly larger datasets compared to MLP. Despite these setbacks, RNN exhibits a notable advantage in terms of reduced prediction runtime. Moreover, the model exhibits enhancements through hyperparameter tuning, indicating the potential for optimization with further exploration.

Increasing the number of traces or the number of layers makes the attack more cumbersome compared to the MLP-based approach, therefore making the MLP approach better, compared to the more sophisticated RNN. For MLP, the convergence was approximately 90% for validation accuracy for the MLP, but not more than 70% for the RNN, even with different permutations of layers. The MLP architecture demonstrates impressive capabilities by successfully retrieving 99% of the coefficients, in 65 hours. However, it's noteworthy that the comprehensive key recovery necessitates the combination of results from different models.

5 CONCLUSION

In this study, the security of the CRYSTALS-Kyber implementations is examined thoroughly. The novelty of the work comes from the execution of higher order TVLA and using recurrent neural network (RNN) SCA to attack the masked Kyber implementation mkm4 using ChipWhisperer. Vulnerabilities in the mkm4 implementation are successfully identified with higher order leakage. This proves that methods other than first order t-tests are needed to attest the security of cryptographic implementations. Though the NTT component did not display any leakage through t-tests, some works (Mujdei et al., 2022) have exploited the vulnerability to perform attacks, due to inherent mathematical properties. The inadequacy of t-tests in directly correlating with successful key recovery is hinted at by related works (Ravi et al., 2022), underscoring the need for a more nuanced approach. In navigating the landscape of modern cryptographic challenges, it becomes imperative to explore modern solutions, such as the integration of DL to introduce noise into the system instead of measures like masking. This line of inquiry addresses the intricacies of cryptographic systems and underscores the evolving nature of cyber security solutions.

REFERENCES

(2023). Nist post-quantum cryptography: Post-quantum cryptography standardization.

Backlund, L., Ngo, K., Gärtner, J., and Dubrova, E. (2022). Secret key recovery attacks on masked and shuffled implementations of crystals-kyber and saber. *IACR Cryptol. ePrint Arch.*

Bock, E. A., Banegas, G., Brzuska, C., Chmielewski, L., Puniamurthy, K., and Sorf, M. (2024). Breaking dpa-protected kyber via the pair-pointwise multiplication. In *Applied Cryptography and Network Security - 22nd*

ACNS 2024, Proceedings, Part II, Lecture Notes in Computer Science.

Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., and Stehle, D. (2018). CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy*.

Chang, Y., Yan, Y., Zhu, C., and Guo, P. (2022). Template attack of lwe/lwr-based schemes with cyclic message rotation. *Entropy*.

Chari, S., Jutla, C. S., Rao, J. R., and Rohatgi, P. (1999). Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology - CRYPTO '99, Proceedings*, Lecture Notes in Computer Science. Springer.

Dubrova, E., Ngo, K., Gärtner, J., and Wang, R. (2023). Breaking a fifth-order masked implementation of crystals-kyber by copy-paste. In *Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop, APKC 2023, Melbourne, VIC, Australia, July 10-14, 2023*. ACM.

Heinz, D., Kannwischer, M. J., Land, G., Pöppelmann, T., Schwabe, P., and Sprenkels, A. (2022). First-order masked kyber on ARM cortex-m4. *IACR Cryptol. ePrint Arch.*

Kannwischer, M. J., Petri, R., Rijneveld, J., Schwabe, P., and Stoffelen, K. (2020). PQM4: Post-quantum crypto library for the ARM Cortex-M4.

Kocher, P. C., Jaffe, J., and Jun, B. (1999). Differential power analysis. In *Advances in Cryptology - CRYPTO '99, Proceedings*, Lecture Notes in Computer Science. Springer.

Mujdei, C., Wouters, L., Karmakar, A., Beckers, A., Mera, J. M. B., and Verbauwhede, I. (2022). Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication. *ACM Trans. Embed. Comput. Syst.*

Ngo, K., Dubrova, E., and Johansson, T. (2021). Breaking masked and shuffled CCA secure saber KEM by power analysis. In *ASHES@CCS: 5th Workshop on Attacks and Solutions in Hardware Security*. ACM.

Ngo, K., Wang, R., Dubrova, E., and Paulsrud, N. (2022). Side-channel attacks on lattice-based kems are not prevented by higher-order masking. *IACR Cryptol. ePrint Arch.*

O'Flynn, C. and Chen, Z. D. (2014). Chipwhisperer: An open-source platform for hardware embedded security research. In *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop*, Lecture Notes in Computer Science. Springer.

Picek, S., Perin, G., Mariot, L., Wu, L., and Batina, L. (2023). Sok: Deep learning-based physical side-channel analysis. *ACM Comput. Surv.*, (11).

Primas, R., Pessl, P., and Mangard, S. (2017). Single-trace side-channel attacks on masked lattice-based encryption. In *Cryptographic Hardware and Embedded Systems - CHES Proceedings*, Lecture Notes in Computer Science. Springer.

Rajendran, G., Ravi, P., D'Anvers, J., Bhasin, S., and Chattopadhyay, A. (2023). Pushing the limits of generic side-channel attacks on lwe-based kems - parallel PC oracle attacks on kyber KEM and beyond. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, (2).

Ravi, P., Bhasin, S., Roy, S. S., and Chattopadhyay, A. (2020a). On exploiting message leakage in (few) NIST PQC candidates for practical message recovery and key recovery attacks. *IACR Cryptol. ePrint Arch.*

Ravi, P., Chattopadhyay, A., and Baksi, A. (2022). Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results. *IACR Cryptol. ePrint Arch.*

Ravi, P., Roy, S. S., Chattopadhyay, A., and Bhasin, S. (2020b). Generic side-channel attacks on cca-secure lattice-based PKE and kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, (3).

Saarinen, M. O. (2022). Wip: Applicability of ISO standard side-channel leakage tests to NIST post-quantum cryptography. In *IEEE International Symposium on Hardware Oriented Security and Trust, HOST*. IEEE.

Schneider, T. and Moradi, A. (2015). Leakage assessment methodology - A clear roadmap for side-channel evaluations. In *Cryptographic Hardware and Embedded Systems - CHES Proceedings*, Lecture Notes in Computer Science. Springer.

Sim, B., Kwon, J., Lee, J., Kim, I., Lee, T., Han, J., Yoon, H. J., Cho, J., and Han, D. (2020). Single-trace attacks on message encoding in lattice-based kems. *IEEE Access*.

Sim, B., Park, A., and Han, D. (2022). Chosen-ciphertext clustering attack on CRYSTALS-KYBER using the side-channel leakage of barrett reduction. *IEEE Internet Things J.*, (21).

Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., and Homma, N. (2022). Curse of re-encryption: A generic power/em analysis on post-quantum kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, (1).

Welch, B. L. (1947). The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*, (1-2).

Yang, B., Ravi, P., Zhang, F., Shen, A., and Bhasin, S. (2023). Stamp-single trace attack on M-LWE pointwise multiplication in kyber. *IACR Cryptol. ePrint Arch.*