Evaluating the effectiveness of Test Vector Leakage Assessment when performed on Kyber running on a softcore RISC-V processor on an FPGA

M.J. Becker

Abstract—**With the rise of quantum computing an update to current cryptography standards is needed, as quantum algorithms break the problems underlying the current encryption standards. There is a draft standard for post-quantum cryptography called ML-KEM, previously known as Kyber, described in the FIPS-203 draft standard. Kyber is a scheme based on Learning With Errors over module lattices. One possible vector of attack on practical implementation of cryptographic schemes is through side-channel analysis. To find out whether sensitive information is leaked through a side-channel, the Test Vector Leakage Assessment method can be used to prove correlation between for example power consumption and sensitive information. This work looks at the effectiveness of TVLA when performed on Kyber running on a softcore processor implemented on an FPGA. After collecting a few thousand traces no definitive conclusion can be drawn based on the TVLA metric alone: either a lot more measurements are needed, a definitive way to discern an intermediate bit or byte or for example filtering or other methods to significantly improve the signal-to-noise ratio. A non-specific t-test based on a fixed vs random ciphertext alone does not give guarantees for finding leakage, due to the extra noise added by the FPGA.**

I. Introduction

Currently the main standards for encryption consist of algorithms such as the Advanced Encryption Standard (AES) set by the National Institute of Standards and Technology (NIST). However, it has been proven that once sufficiently powerful quantum computers will be put to use, the underlying problem of AES can be solved much faster [1]. As this compromises encryption, the NIST has been running a project on standardising encryption algorithms which are quantum resistant, known as the Post-Quantum Cryptography Standardization Project. Recently the NIST has published the first draft of FIPS-203, a draft standard for post-quantum Key Encapsulation Mechanism (KEM) based on CRYSTALS-Kyber [2]. It is a KEM based on the hardness of the Learning With Errors problem, over module lattices. As this scheme is relatively new, more research can be done into the implementation side of things.

One problem related to the implementation of such a cryptographic scheme, is the possibility of attack through Side-Channel Analysis (SCA), where for example power consumption or EM-waves radiated by the chip are measured in order to derive secret information. Simply put, if the power consumption of a chip correlates to secret information, a hacker could devise a method of deducing the secret-key through that side channel [3]. As embedded devices are still on the rise [4], the chance of a hacker or adversary to have access to a device to measure power consumption or put an EM-probe near a device is only ever increasing.

A well-known way of identifying such leakage of information is Test Vector Leakage Assessment (TVLA) [5]. This method uses statistical analysis in conjunction with supplied inputs and power consumption measurements to tell whether the power consumption of a chip correlates to the input supplied or even the secret-key itself. The idea is that a set comprised of traces, measurements of power consumption over time, of a single input has a different mean and variance when compared to a set comprised of traces of random inputs, at least for the non-specific version.

With AES, the sets are usually discerned by a certain intermediate value. This can be done, as AES is deterministic: given a certain input, intermediary values will be the same. As in this work a different cryptographic algorithm is looked at, this is not possible due to inherent randomness within the algorithm [6].

In this work, the effectiveness of TVLA in a less than ideal situation will be looked at. The NIST draft version of Kyber is put on a softcore processor on an FPGA, and the power consumption is measured, in order to see whether leakage is identified. Only the decryption step is looked at, as that is the most interesting part for an adversary in case of a static key, as then the device can be polled with multiple ciphertexts after one another.

II. Background

CRYSTALS-Kyber is a scheme designed to allow for cryptography in the post-quantum era [6]. It is built on the LWE problem, but uses polynomial vectors from a ring to allow for faster computation. For the full INDistinguishable under Chosen Ciphertext Attack (IND-CCA) version of Kyber the following algorithms are defined: key generation, encapsulation, encryption, decapsulation, and decryption. For an adversary trying to find the secret key, the decapsulation and decryption are the most interesting steps of the algorithm. For one, because the secret key is present in these algorithms, and two because this algorithm is run many times with the same key in a static key setting, allowing for the collection of multiple traces. In this work, focus was put on the decryption algorithm, and especially the final step of the inverse Number Theoretic Transform (NTT). The NTT is a special case of the Discrete Fourier Transform, and allows for coefficient wise multiplication of polynomials on a ring. This significantly reduces the resources needed to multiply two polynomials, hence its use within the Kyber algorithm. In the final step of going from the NTT-domain back to polynomials there is a multiplication with a constant. This makes it easier to see the impact of different inputs on this part of the algorithm, and therefore reason to suspect side-channel leakage.

Side-Channel analysis (SCA) is the process of looking for secret information leaked by the practical implementation of cryptographic systems. This is usually done either by placing a EM probe near the processing unit, or by measuring the power consumption of the chip. In the past, visual inspection of the measurement of power over time (also called a trace) could be enough to decipher the secret key, if for example a zero or a one took differing amounts of time to compute. Nowadays more sophisticated methods are used, such as oracle-based attacks or correlation power analysis, see [3] for a more extensive list. In order to identify the possibility of a side-channel attack without building a full attack, it is enough to identify the leakage of information through a side-channel, e.g. to show that different states within a chip have statistically different traces, a different power consumption over time. The industry standard of identifying leakage is Test Vector Leakage Assessment (TVLA) [5], also defined as ISO/IEC 17825:2024.

TVLA is a process of providing different ciphertexts to a cryptographic system, and seeing if different ciphertext inputs giving different intermediary bits show statistical differences in for example the power consumption of the chip [5]. This is done by choosing ciphertexts in such a way, that part of them give a certain intermediate bit or byte, and the other part the opposite. Many traces of both sets are collected, and then a Welch's t-test is used to see whether the two sets differ from one another in a statistically significant way. If there is a difference, it can be concluded that an adversary could find information and thus recover the secret key through the use of a side-channel without creating a full attack. If more traces are needed for a positive t-test for one implementation over the other, it could be concluded that the implementation with more traces needed is harder to attack than the implementation which needs fewer traces [3].

III. Related work

Avanzi et al. [6] published the submission for Kyber to the Post-Quantum cryptography standardisation effort of the NIST. In that document, the theoretical background to the scheme is given, as well as a few reference implementations for performance evaluation. A newer version of the scheme is given in the FIPS-203 draft of the NIST [7]. In the PQCLEAN project [8] an effort was made to improve upon these implementations, and give a clean platform-independent implementation of Kyber and other candidates for standardisation. As these are reference implementations, and platform-independent, a lot of optimisation can be done for individual platforms. In the thesis by T. Fritzmann [9] an effort is made to implement Kyber in RISC-V in a more performant manner, by making use of the possibility to add custom extensions to the ISA. In the thesis by J. Meijer [10], the PQCLEAN implementation is analysed through power consumption on both ARM and RISC-V, and concludes that there is leakage in the final step of the inverse NTT on both ISAs. In the paper by Xu et al. [11] a practical attack on Kyber is presented, using chosen ciphertexts. By choosing the ciphertexts in a special manner, a relation between secret information and side-channel leakage of the reference C implementation from PQClean [8] was shown. In a conference paper by A. Barenghi, G. Pelosi and Y. Teglia [12] digital signal processing was used to greatly increase the chance of finding leakage in a noisy part of AES running on an FPGA.

IV. Methodology

In order to determine whether information leaks through side-channels a so called Test Vector Leakage Assessment (TVLA) was done. TVLA is a method of conducting tests and corresponding analysis designed to show whether a DUT is susceptible to side-channel attacks [5]. In this paper, a non-specific t-test is done. This means the following: either a fixed or random ciphertext is supplied to the DUT, and the power consumption is measured during the following decryption operation. The trace, a measurement of power over time, is then sorted into either set A if it was a fixed ciphertext or set B if it was a random ciphertext. Then, for each point in time, set A and B are compared using a two-tailed Welch's t-test, see equation 1 [5]. In the equation, μ_0 is the mean of set A, s_0^2 the variance of set A and n_0 the cardinality of set A. The same characters with subscript 1 are the same concepts but for the other set.

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}}} \tag{1}$$

The mean and variance of the two sets are compared, and if the two sets differ significantly, the t value will be higher. A t-value of 4.5 corresponds to a confidence of over 0.99999 to reject the null hypothesis: the samples in each of the two sets are drawn from the same population [5]. In order to get the mean and variance of set A and B, a single-pass incremental algorithm is used, so that for each measurement added to a set only the current value is updated. This saves a considerable amount of time, when compared to the traditional method of using all values in a set to calculate the mean and variance [5]. The first raww moment M can be updated as follows, where Q is either set A or B, Q' is the updated set, and n the cardinality of that set:

$$\begin{gathered} M_{1,Q'} = M_{1,Q} + \frac{\Delta}{n}, \\ \Delta = y - M_{1,Q} \end{gathered} \tag{2}$$

Then for the variance s^2 the following equation is used, where CS_2 is the second central sum, CM_2 is the second central moment and n is again the cardinality of the set:

$$\begin{gathered} CS_{2,Q'} = CS_{2,Q} + \frac{\Delta^2(n-1)}{n}, \\ s^2 = CM_2 = \frac{CS_2}{n} \end{gathered} \tag{3}$$

The implementation of the TVLA procedure can be read in the following section, where more practical details will be explained.

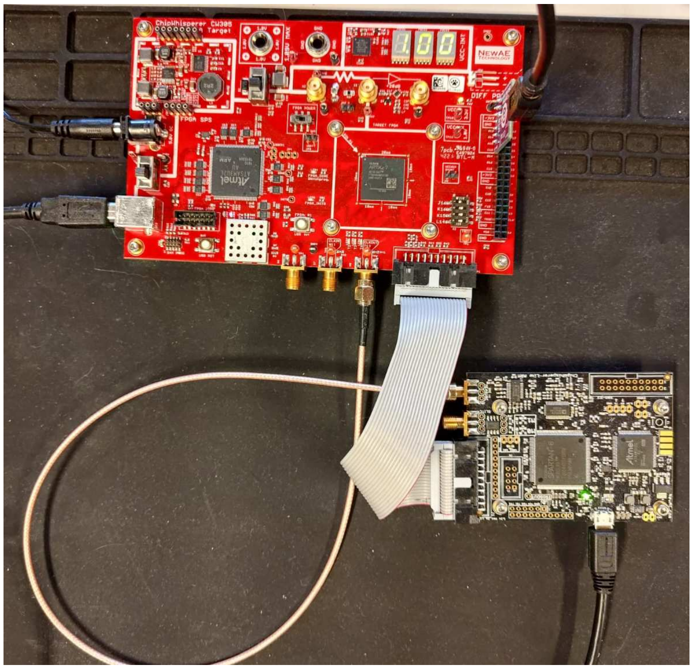

Fig. 1. Experimental setup

V. Procedure

First off the Kyber-PKE key generation algorithm is run on the host computer, and the secret key (sk) is sent to the DUT. Then, the same secret and public key are used to generate a ciphertext with a random message, which will be the fixed ciphertext. Now a loop is entered: first a coinflip to determine whether the fixed or a random ciphertext will be sent. Then, in the case of a random ciphertext, encryption is run with a random message to give a random valid ciphertext. The ciphertext will be sent over UART to the DUT. On the DUT, Kyber-PKE decryption will be run with the received ciphertext, until the final step of the inverse NTT. As the buffer of the CW-Lite is not large enough to fit a measurement of the entire fqmul operation, the operation will be captured in eight chunks of 32 loop iterations. The trigger pin is set to high, followed by 32 loop iterations. Then the trigger pin is set to low, and the DUT waits for a character to be received over UART, as that signifies that the CW-Lite is ready to capture again. Repeat until entire fqmul operation is calculated, then finish decryption and wait for new ciphertext. Once a trace has been captured, the host computer runs the mean and variance update algorithm. If you were to follow the TVLA procedure completely, you calculate the t-value as well and stop measuring once the t-value exceeds 4.5 [5]. In this paper I did not do this, due to several reasons discussed in section VII. Instead, I set out to capture a set amount of traces per set, although due to the nature of a coinflip both sets are often not exactly the same size.

VI. Experimental

For a diagram of the experimental setup, see figure 2. A ChipWhisperer Lite and a ChipWhisperer CW305 FPGA Target are connected to a host computer through USB. I put the NEORV32 softcore processor on the FPGA [13]. It is an open-source RISC-V processor designed to be able to run out of the box on a large range of hardware. In order to keep the runtime of the Kyber algorithm small, the multiply extension is enabled. DSP blocks are used to implement the multiply extension for further time reduction. Furthermore, a USB UART interface is connected to the FPGA as well, to be used to communicate between the host computer and the NEORV32 soft-core processor directly. NEORV32 allows programs to be synthesized within memory, so once the bitstream is uploaded to FPGA it boots directly into my program. The program itself is simple: receive secret-key once, then enter a loop where everytime it waits for a ciphertext and then runs the decryption algorithm. Some small changes where made to the decryption algorithm in order to speed up the capture and analysis portion: the final step of the inverse NTT is a loop which calls the fqmul function 256 times, and there some code was added to allow for segmented capture, see below.

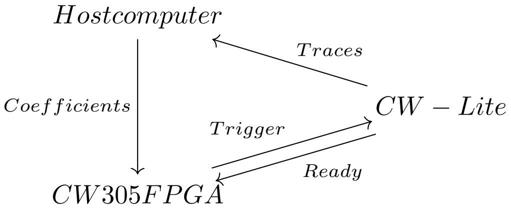

Fig. 2. Experimental diagram

```
for (j = 0; j < 256; j++) {
    int result = j%32;
    if(result == 0){
        neorv32_gpio_port_set(0);
        neorv32_uart0_getc();
        neorv32_gpio_port_set(1);
    }
    r[j] = fqmul(r[j], f);
}
```

The experiments consist of the same procedure each time. One experiment was to capture a thousand traces, repeated four times, in order to see the influence of a different fixed ciphertext on the resulting figures. Another experiment was to capture 5000 traces, to see how adding more measurements influences the maximum t-value.

VII. Results & Discussion

As can be seen in figure 3, the maximum t-value found after any amount of traces does not show a clear trend. Usually, on a microcontroller with for example AES, the maximum t-value would keep rising as more measurements are added, as the sets at a certain point in time definitely do differ. That is why in the TVLA-test procedure proposed by [5] the measurements are stopped once any point goes above the threshold value of 4.5, as then it should only become more obvious leakage afterwards. However, that is not the case here. Instead, the maximum t-value goes above and then below the threshold value. This can be ascribed to several different phenomenon. First of all, because the FPGA and the softcore processor add

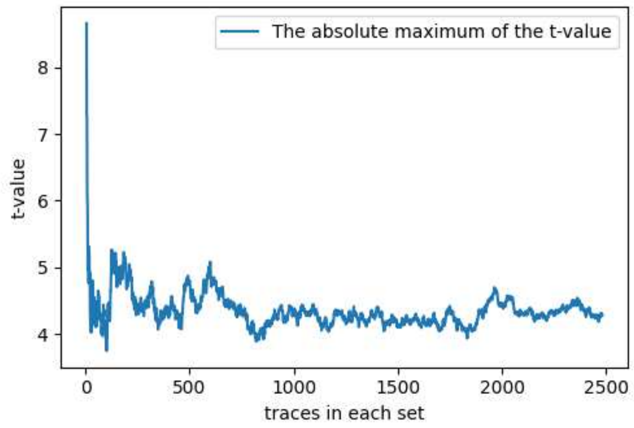

Fig. 3. The maximum value of the absolute t-value as traces get added to each set

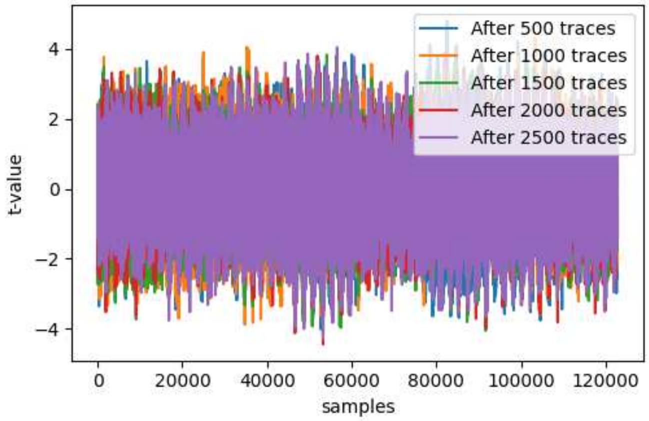

Fig. 4. T-value of entire trace at different amount of traces added to each set

noise on the power rails [14]. This means that at any point the influence of the secret information on the measurement is a lot lower. In [12] a multi-bandpass filter was proposed in order to reduce noise and increase the likelihood of finding leakage significantly. This method was cited in a paper where they attacked the encryption of a bitstream uploaded to an FPGA, which signifies its effectiveness on a similar platform as was used here [14]. Furthermore, due to the randomness in the Kyber algorithm, intermediary values will not always be the same. This is due to for example the compression and decompression of the ciphertext, but also the influence of randomness on the polynomials before they are compressed to the message by rounding. As mentioned, TVLA is an ISO standard (ISO/IEC 17825:2016), but during the writing of this paper ISO/IEC 17825:2024 was published, perhaps to address some shortcomings put forward by [15]. In figure 5 the influence of adding more measurements to each set on the t-value can be seen for the first fifty samples, and in figure 6 a few samples are shown over the amount of measurements in each set.

In order to see the difference a different fixed ciphertext makes, see figures 9, 7, and 8. Even though the individual graphs in figure 9 look way different the result is about similar: after a certain amount of traces get added to each set the maximum t-value seems to stay within 4 and 5, sometimes crossing the 4.5 threshold but also dipping below it again.

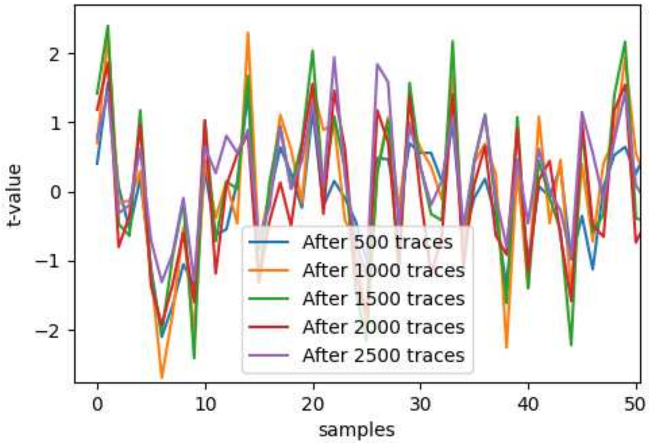

Fig. 5. T-value of first fifty samples, shown at different amount of traces added to each set

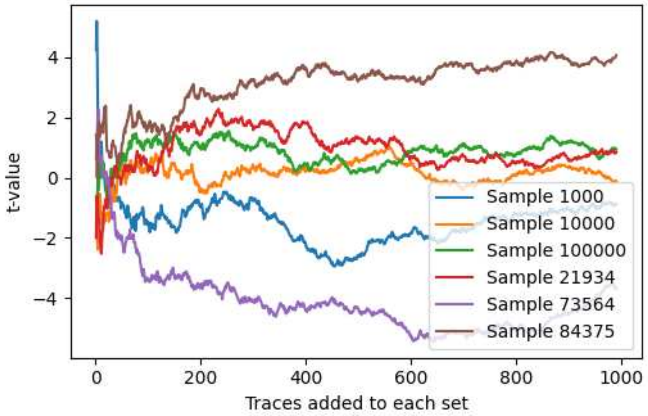

Fig. 6. Various samples shown at different amount of traces added to each set

Either way, in the way I have done measurements here, there is no solid conclusion to be drawn. The signal to noise ratio is too high, and the correlation between information and power consumption is not obvious. By filtering and choosing malicious ciphertexts the correlation should become more obvious if it is there. Another option is to add a lot more measurements, even though the expectation put forward by microcontroller measurements is that a few thousand measurements should be enough to clearly show leakage [16]. Furthermore, the trends seen in figure 3 do not suggest that simply adding measurements will show definitive leakage. Unfortunately, the TVLA procedure in itself only really gives a reason to reject the null hypothesis of both sets being the same once the threshold value is clearly above the threshold, and no conclusion if otherwise [15]. More comments have been made on the procedure itself, such as [15] questioning the validity and significance of TVLA test results and recommending further statistical tools to be used to reduce false positives/false negatives depending on

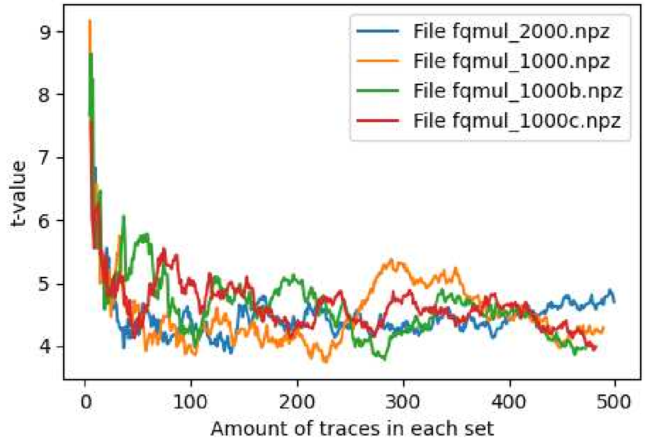

Fig. 7. Maximum t-value for different fixed ciphertexts as traces get added to each set

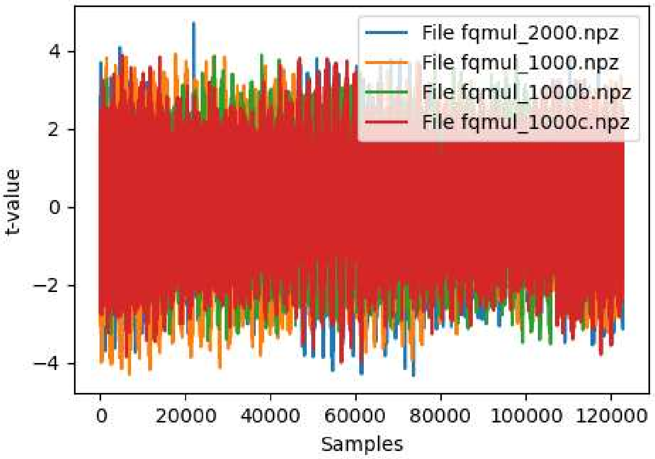

Fig. 8. T-value of four different fixed ciphertexts across all samples

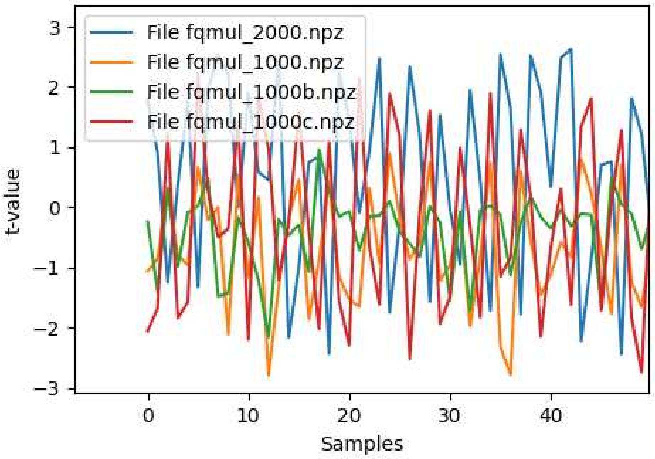

Fig. 9. T-value of four different fixed ciphertexts across the first 50 samples

the use case. The ISO standard (ISO/IEC 17825:2024) was updated during the final days of writing this report, so perhaps the test procedure recommendation has been changed from what was done here, so it is definitely recommended to see what was changed in the newest version.

VIII. Conclusion

To summarize, the expectation was that a few thousand traces captured would lead to a definitive conclusion about there being side-channel leakage on a reference implementation of Kyber running on a softcore processor. After implementing Kyber on the NEORV32 softcore processor on the ChipWhisperer CW305 FPGA, that expectation was not met. After 5000 total measurement the trend in the maximum t-value does not suggest definitive leakage. There are ways in which leakage might show up within this set of measurements: for example by filtering, to improve signal-to-noise ratio, or by crafting ciphertexts, to highlight secret-key coefficients. The most straightforward approach is to keep adding measurements to both sets, although the trend seen in figure 3 does not suggest adding more measurements will show definitive leakage. It is recommended to look into the newer version of the ISO specification (ISO/IEC 17825:2024) to see if the procedure was changed in a significant manner, as most likely that also addresses some concerns about the effectiveness of the test procedure.

References

[1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing*, vol. 26, p. 1484–1509, Oct. 1997.

[2] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. M. Kelsey, J. Lichtinger, Y.-K. Liu, C. A. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone, “Status report on the third round of the nist post-quantum cryptography standardization process,” September 2022. https://doi.org/10.6028/NIST.IR.8413-upd1.

[3] P. Ravi, A. Chattopadhyay, J. P. D’Anvers, and A. Baksi, “Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results.” Cryptology ePrint Archive, Paper 2022/737, 2022. https://eprint.iacr.org/2022/737.

[4] IOT Analytics. IoT Connections Market Update. 2022. https://iot-analytics.com/number-connected-iot-devices/ (accessed on 16 January 2024).

[5] T. Schneider and A. Moradi, “Leakage assessment methodology - a clear roadmap for side-channel evaluations.” Cryptology ePrint Archive, Paper 2015/207, 2015. https://eprint.iacr.org/2015/207.

[6] R. M. Avanzi, J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “Crystals-kyber algorithm specifications and supporting documentation,” 2017. https://api.semanticscholar.org/CorpusID:198992527.

[7] National Institute for Standards and Technology, “Module-lattice-based key-encapsulation mechanism standard (draft)..” nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.ipd.pdf, 2023.

[8] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers, “Improving software quality in cryptography standardization projects,” in *IEEE European Symposium on Security and Privacy, EuroS&amp;P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*, (Los Alamitos, CA, USA), pp. 19–30, IEEE Computer Society, 2022.

[9] T. Fritzmann, G. Sigl, and J. Sepúlveda, “Risq-v: Tightly coupled risc-v accelerators for post-quantum cryptography,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, p. 239–280, Aug. 2020. https://tches.iacr.org/index.php/TCHES/article/view/8683.

[10] J. Meijer, “Towards future proof cryptographic implementations: Side-channel analysis on post-quantum key encapsulation mechanism crystals - kyber,” June 2023. http://essay.utwente.nl/96514/.

[11] Z. Xu, O. Pemberton, S. S. Roy, D. Oswald, W. Yao, and Z. Zheng, “Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber,” *IEEE Transactions on Computers*, vol. 71, no. 9, pp. 2163–2176, 2022.

[12] A. Barenghi, G. Pelosi, and Y. Teglia, “Improving first order differential power attacks through digital signal processing,” in *Proceedings of the 3rd International Conference on Security of Information and Networks, SIN 2010, Rostov-on-Don, Russian Federation, September 7-11, 2010* (O. B. Makarevich, A. Elçi, M. A. Orgun, S. A. Huss, L. K. Babenko, A. G. Chefranov, and V. Varadharajan, eds.), pp. 124–133, ACM, 2010.

[13] S. Nolting and A. the Awesome Contributors, “The NEORV32 RISC-V Processor,” Aug. 2023. https://github.com/stnolting/neorv32.

[14] A. Moradi, A. Barenghi, T. Kasper, and C. Paar, “On the vulnerability of fpga bitstream encryption against power analysis attacks - extracting keys from xilinx virtex-ii fpgas.” Cryptology ePrint Archive, Paper 2011/390, 2011. https://eprint.iacr.org/2011/390.

[15] C. Whitnall and E. Oswald, “A critical analysis of iso 17825 (‘testing methods for the mitigation of non-invasive attack classes against cryptographic modules’).” Cryptology ePrint Archive, Paper 2019/1013, 2019. https://eprint.iacr.org/2019/1013.

[16] T. Teague, “Side-channel analysis on post-quantum cryptography algorithms.,” *Computer Science and Computer Engineering Undergraduate Honors Theses*, 2022. Retreived from https://scholarworks.uark.edu/csceuht/106.