Power-based Side Channel Attack Analysis on PQC Algorithms

Tendayi Kamucheka, Michael Fahr, Tristen Teague, Alexander Nelson, David Andrews, Miaoqing Huang
Department of Computer Science and Computer Engineering
University of Arkansas
{*tfkamuch,mjfahr,tdteague,ahnelson,dandrews,mqhuang*}*@uark.edu*

***Abstract*—Power-based side channel attacks have been successfully conducted against proven cryptographic algorithms including standardized algorithms such as AES and RSA. These algorithms are now supported by best practices in hardware and software to defend against malicious attacks. As NIST conducts the third round of the post-quantum cryptography (PQC) standardization process, a key feature is to identify the security candidate algorithms have against side channel attacks, and the tradeoffs that must be made to obtain that level of protection. In this work, we document the development of a multi-target and multi-tool platform to conduct test vector leakage assessment of the candidate algorithms. The long-term goals of the platform are to 1) quantify test vector leakage of each of the primary and alternate candidates, 2) quantify test vector leakage of each of the candidates when adjustments and adaptations (e.g., masking) are applied, and 3) assess the equivalent security levels when tools of varying sophistication are used in the attack (e.g., commodity vs. specialized hardware). The goal of this work is to document the progress towards that standardized platform and to invite discussion in how to extend, refine, and distribute our tools.**

1. Introduction

Cryptographic schemes developed for classical computers have relied on the assumption that factoring large integers is computationally intractable. Shor's algorithm showed that integers could be factored in polynomial time given a sufficiently powerful quantum computer. Consequently, the need for newer quantum-resistant cryptographic standards has given rise to a new class of cryptography now commonly known as Post Quantum Cryptograpghy (PQC).

Post Quantum Cryptography represents a class of quantum-proof or quantum-resistant cryptographic schemes not known to be susceptible to quantum computers. Plainly, no significant advantage is gained by executing a cryptanalytic attack on a quantum computer over a classic computer. At the time of this writing, promising PQC schemes are either code-based, hash-based, lattice-based, multivariate-based, and supersingular elliptic curve isogeny-based. It is not yet known which of these schemes will emerge as future standards. Furthermore, although large-scale quantum computing attacks still remain outside the scope of modern engineering capabilities, preparation has already began to develop alternative schemes resistant to quantum attacks.

In 2009, the National Institute of Standards and Technology (NIST) launched the PQC initiative to standardize one or more quantum-resistant cryptographic schemes [1]. NIST is a non-regulatory government agency that develops technology, standards, and guidelines to help federal agencies meet requirements of the Federal Information Security Modernization Act of 2004 (FISMA) [2]. NIST is also responsible for producing Federal Information Processing Standards (FIPS) in accordance with FISMA. In 2016, NIST announced the NIST PQC competition in which contestants from all around the world were invited to submit candidate quantum-resistant cryptoghraphic schemes for evaluation and ultimately standardization. NIST does not intend to pick a single winner out of the competition [3]. The successful candidate(s) will be selected for standardization and will augment the cryptographic algorithms specified in Federal Information Processing Standard (FIPS) 186-4 [4], [5].

By November 2017, 82 candidate algorithms were submitted for consideration of which 69 met the minimum requirements for entry [6]. The first round submissions included public-key encryption (PKE), key-establishment mechanisms (KEM), and digital signature (DS) schemes. Twenty six candidate schemes were selected for the second round of competition. NIST narrowed the selection to 9 PKE/KEM schemes and 6 DS schemes for the third round in July of 2020 [4], [6]. Most of the earlier candidates fell out the race because they were significantly compromised by cryptanalytic attacks while some merged to become stronger contenders. In the second round of the competition NIST evaluated the candidates schemes on cost and performance as well as security.

During the third round NIST has incorporated performance and side channel resistance as features in the selection process. A standardized platform to analyze these features is necessary to prove these features for a given candidate. In this work, we present the development of a platform to analyze side channel power analysis of Round 3 finalist and alternate candidate algorithms. The work presented in this paper is in pursuit of our overall objective to quantify the side channel security of PQC candidates and their derivatives to an array of power-based techniques ranging from commodity hardware to specialized tools.

TABLE 1: NIST PQC competition round 3 finalists grouped by algorithm category.

Finalists	Public-key Encryption & Key-encapsulation Mechanisms	Code-based	Classic McEliece [7]
		Hash-Based	-
		Lattice-Based	CRYSTALS-Kyber [8] NTRU [9] SABER [10]
		Multivariate PKE	-
	Digital Signatures	Code-based	-
		Hash-Based	-
		Lattice-Based	CRYSTALS-Dilithium [11] FALCON [12]
		Multivariate PKE	Rainbow [13]
Alternate Candidates	Public-key Encryption & Key-encapsulation Mechanisms	Code-based	BIKE [14] HQC [15], [16]
		Hash-Based	-
		Lattice-Based	FrodoKEM [17] NTRU Prime [18]
		Multivariate PKE	-
	Digital Signatures	Code-based	-
		Hash-Based	SPHINCS+ [19]
		Lattice-Based	-
		Multivariate PKE	GeMSS [20]

2. Background

2.1. Round 3 Candidates

In the third round of the PQC competition, the selected candidate algorithms are designated as either finalists or alternate candidates. The finalists are the more likely schemes to be considered for standardization while the alternates are schemes advanced into the third round with some but very low likelihood of being standardized [21].

Submissions to the competition have formally been separated into two classes, public-key encryption and key-establishment schemes, and digital signature schemes. In addition to these classes, the candidate algorithms represent multiple categories of cryptographic schemes by their underlying mathematical formulation. These categories are: 1) code-based, 2) hash-based, 3) lattice-based, and 4) multivariate PKE-based cryptography. Table 1 shows Round 3 candidates and their placement in the various algorithm categories.

Lattice-based Schemes are the most common among third round finalists. Because they are similar in their underlying mathematics and would therefore be susceptible to similar attacks, it is likely that at most one lattice-based algorithm will be selected for standardization [21]. Lattice-based schemes have emerged as popular candidates due to their relatively simple construction and robust security guarantees given by their underlying problems even in the worst case scenarios. Typical lattice problems that form the basis for their security include Shortest Integer Solution (SIS), Shortest Vector Problem (SVP), Learning With Errors (LWE), Ring-LWE (R-LWE), and Module-LWE (M-LWE). R-LWE and M-LWE are potentially reducible to SVP [22]. Thus far, lattice cryptosystems have been resistant to theoretic attacks. However, there has been less focus regarding the robustness of lattice-cryptography to side channel assisted attacks. More scrutiny of finalist candidates may reveal potential concerns. These concerns must be evaluated in a standardized and fair manner across all finalist candidates.

2.2. Side Channel Attack Methods

Typical side channel attack methods include cold boot attack, fault attack, timing attack, and power analysis.

Cold boot attacks require the attacker to have physical access to the device. In this attack, the adversary applies coolant to the DRAM to freeze it and slow down the rate of information decay. The data on the DRAM is then dumped onto an external device for further analysis. A general attack targeting the number theoretic transform (NTT), which is commonly used by many ring- and module-LWE PQC algorithms like Kyber and NewHope, was demonstrated [23]. The attack recovered between 60 and 90% of the secret coefficients depending on the parameters used for algorithm strength.

Another common method of attack used for extracting secret information from cryptographic devices is the fault attack. This attack operates by having the adversary induce a fault into the cryptographic device, causing unintended operations to reveal secret information that can lead to a key recovery.

Timing attacks are yet another common method to extract secret information. Timing attacks are accomplished by analyzing the amount of time required to process cryptographic algorithms on varying inputs and using this data to recover secret information. A major countermeasure to this attack is to implement the algorithms in constant-time. However, this can incur significant overhead for the algorithm. Timing attacks have been found to be effective against many of the PQC algorithms including FrodoKEM, HQC, and Falcon.

Power analysis attacks are broader and more varied than other types of side channel attacks. One type of power

analysis attack that has been demonstrated is a simple power analysis (SPA) attack. In most cases, this type of attack requires a direct analysis of observed power or electromagnetic (EM) radiation traces [24], [25]. The other form of attack is the differential power analysis (DPA), which applies additional analysis techniques like statistical correlation or device profiling and templating [25]. The analysis is performed on several, sometimes thousands of traces in conjunction with in-depth knowledge of the inner workings of the algorithm.

3. Power Analysis on Selected PQC algorithms

3.1. Equipment and Platforms

We are currently evaluating both hardware and software implementations of the target PQC algorithms. We collect the power traces using a Tektronix oscilloscope (model: MDO34 3-BW-200, bandwidth: 200 MHz, sample rate: 2.5 GS/s) and a commodity off-the-shelf NewAE ChipWhisperer-Lite.

FPGA Platform: Hardware implementations are evaluated on a Xilinx Virtex®-7 (XC7VX485T-2FFG1761C) VC707 FPGA board. Xilinx Vivado 2019.1 and SDK tools are used for design and synthesis of HDL (hardware description language) implementations. The VC707 FPGA board is equipped with one Texas Instruments PTD08A020W, one PTD08A010W, and five PTD08D210W DC-to-DC power modules. These power modules convert the 12V main input and supply the various power rails, which power the internal circuitry of the FPGA [26], [27]. The PTD08A020W and PTD08A010W modules each have a single voltage output and a corresponding current output for VCCINT and VCC_ADJ respectively. Each of the PTD08D210W modules has two voltage outputs, each of which has a corresponding current measurement output. Small copper hooks were soldered on the exposed pads of the current outputs corresponding to VCCINT, VCCAUX_IO, and VCCBRAM. The oscilloscope analog probes are attached to the copper hooks (see Figure 1) with the ground of the probes attached to the common ground of the FPGA board. Since the VC707 does not have any dedicated GPIO pin headers, the GPIO pins located inside the 20-pin XADC connector are used to get trigger signals, which are attached to the digital probes of the oscilloscope. For a more realistic attack where no trigger signals available to the attacker, it is possible to achieve the same result by soldering a suitable wire on the exposed RX pin of the UART port. Some profiling may be required to narrow the search space of the measured trace.

Microcontroller Platforms: Software implementations are evaluated on a STM32F4 Discovery (STM32F407G-DISC1) evaluation kit. The evaluation board includes an STM32F407VGT6 microcontroller, which features a 32-bit ARM Cortex-M4 embedded processor. The software implementations are obtained from the "pqm4" library [28]. The Discovery board was modified using a solder rework station to enable direct contact to perform power analysis

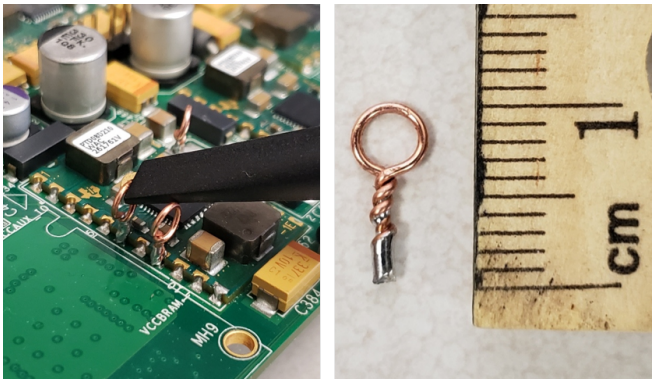


Figure 1: Xilinx VC707 FPGA board probe attachment.

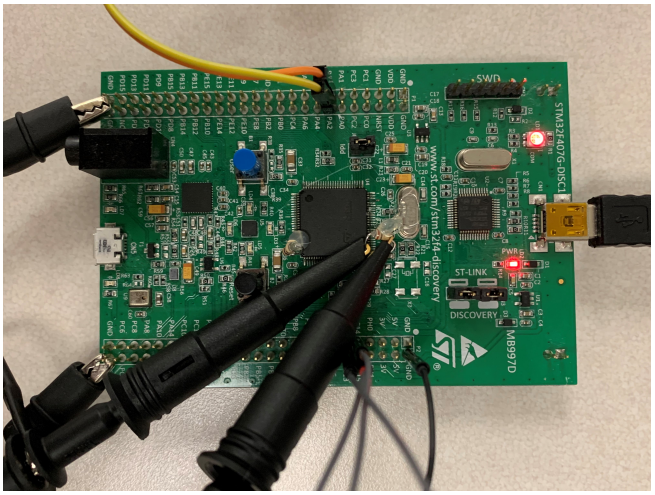


Figure 2: STM32F407 microcontroller setup.

(see Figure 2). Specifically, on the LQFP100 package that houses the STM32 controller, pins 11, 19, 22, 28, 50, 75, and 100 were lifted from the pads, and 1Ω resistors were attached in series between pins 25 and 50 and their pads respectively. Hot glue was affixed to the resistors to stabilize the components so that the oscilloscope probes could be attached. A USB-TTL converter was used with a 2 pin cable connected to pins PA2 and PA3. These pins assisted in communication with the algorithms on the host machine. A logic probe was used with GPIO pins PC13 to PC15 to assist in timing the oscilloscope captures with specific parts of the algorithms. The oscilloscope trigger was configured to automate capturing the waveform based on output of the logic probe. To enable single-ended passive analog probes with the digital logic, a common ground was needed. Therefore, two analog probes were used. The probe tips were connected to each side of the shunt resistor and referenced to ground. A virtual math channel available on the Textronix oscilloscope calculated the difference between the potential of the probes to generate the power trace. Note that this is a digital differential not an analog differential. In future work, we plan to explore the use of differential or active probes in order to quantify their benefit to SPA/DPA

side channel analysis.

Our final capture platform is the NewAE ChipWhisperer-Lite 32-Bit synchronous power capture tool with a removable STM32F303 microcontroller featuring an ARM Cortex-M4 processor. This tool is to add a point of comparison to demonstrate the capability of commodity hardware when compared with higher-end tools (such as oscilloscopes). The capture device uses a Xilinx Spartan-6 FPGA to obtain 10-bit ADC captures at 105 MS/s. Our setup currently captures power traces from the default target board (i.e., STM32F303 mocrocontroller). Future work will capture traces from the STM32F4 Discovery board to make direct comparisons between capture methods. At this stage we can capture power traces of candidate algorithms, but do not have a basis for comparing these traces to those captured by the Tektronix oscilloscope. Therefore analysis of these traces is left to future work.

3.2. Test Vector Leakage Assessment

The Test Vector leakage Assessment (TVLA) [29] has become the de facto standard in the evaluation of side channel measurements. TVLA identifies differences between two sets of side channel measurements, such as power and EM traces, by computing the uni-variate Welch's t-test for the two sets of measurements. The test can be used to detect side channel leakages that are not associated with any specific leakage model [30]. Two sets of measurements are taken, the first with fixed inputs and the second with random inputs, which we will refer to as T_f and T_r, respectively. At each time step a pass/fail decision is given by testing for a null hypothesis such that the means of the two sets is equivalent. The TVLA at each time step is calculated as follows:

$$TVLA = \frac{\mu_r - \mu_f}{\sqrt{\frac{\sigma_r^2}{n_r} + \frac{\sigma_f^2}{n_f}}}, \tag{1}$$

where μ_r, σ_r, and n_r are the mean, standard deviation, and number of traces collected for T_r; likewise for trace set T_f. The null hypothesis is rejected with a confidence level of 99.9999% if the absolute value of the t-test is greater than 4.5 [29]. Assuming independent leakage, any variation in the power traces are the result of the underlying computation not other factors such as hardware architecture features [31]. Therefore, a rejected null hypothesis —which constitutes a fail decision— suggests that the two trace sets T_r and T_f are different and as such might leak some information about the underlying computation. We use TVLA to confirm the presence and/or absence of side leakages for power traces measured on the oscilloscope and the ChipWhisperer.

3.3. Implementation and Results

The power traces of software implementation were captured for the Round 3 NIST PKE/KEM and DS algorithms from the "pqm4" library. Currently, the schemes CRYSTALS-KYBER, SABER, NTRU, CRYSTALS-DILITHIUM, and FALCON are all implemented on the microcontroller. Additionally, a masked version of the CRYSTALS-KYBER decapsulation procedure was implemented and tested on the microcontroller. Multiple different implementations existed for each scheme, including a clean reference implementation, a reference implementation submitted to NIST, an optimized implementation in plain C, and an implementation with Cortex-M4 specific optimizations. For our power analysis, the software traces were captured from the M4 optimized implementation. This library uses the PQClean API and is required to implement three different functions. The KEMs are required to implement key generation, encryption, and decryption functions. The DS algorithms are required to implement key generation, sign, and sign open functions. Triggers were used to assist in timing the start and end of these operations. Several binaries were available for each of the implementations, including a test, speed, hashing, and stack binary. The test binary was used to verify the algorithms were implemented correctly. For KEMs, it tested that the same shared key was derived for both an Alice and Bob. For signature schemes, it tested the generated signature to ensure it could be verified correctly. Traces for each of the schemes were obtained during the execution of the test binaries.

In addition, we are in the progress to design pure hardware implementations of PQC algorithms on FPGAs. Currently, we have completed the hardware implementation of CRYSTALS-KYBER algorithm.

3.3.1. CRYSTALS-KYBER. We implemented the Kyber-512 variant based on Huang et al [32] on the FPGA. The parameters for Kyber512 are defined as $k = 2$, $n = 256$, $q = 3329$, $\eta_2 = 2$. Other parameters η_1, d_u, and d_v are positive integers. $M = \{0,1\}^n$ is the message space and $m \in M$. χ_η is the centered binomial distribution and $\chi_{n,\eta}$ is the distribution of n-degree polynomials with each element independently sampled from χ_n [33]. The functions $Compress$ and $Decompress$ are defined as follows:

$$\text{Compress}_q(x,d) := \lfloor \frac{2^d}{q} \cdot x \rceil \text{mod}^{(+)} 2^d, \tag{2}$$

$$\text{Decompress}_q(x,d) := \lfloor q/2^d \rceil \cdot x, \tag{3}$$

Algorithm 1: Kyber.INDCPA.KEYGEN

Input: $(\rho, \sigma) \xleftarrow{\$} \{0,1\}^{256} \times \{0,1\}^{256}$
Input: $A \leftarrow U(q)^{k \times k}$
Result: return PK $\leftarrow (\hat{t}||\rho), SK \leftarrow \hat{s}$
A $\xleftarrow{\rho} U(q)^{k \times k}$
(s, e) $\xleftarrow{\sigma} \chi_{m,eta_1}^{k \times k}, \chi_{m,eta_1}^{k \times k}$
$\hat{\text{s}} \leftarrow \text{NTT}(s)$
$\hat{\text{e}} \leftarrow \text{NTT}(e)$
$\hat{\text{t}} \leftarrow A \circ \hat{s} + \hat{e}$

Algorithm 2: Kyber.INDCPA.ENC

Input: $PK = (\hat{t}||\rho)$
Input: $m \in M$
Input: r $\xleftarrow{\$} \{0,1\}^{256}$
Result: return c $\leftarrow (c_1||c_2)$
A $\xleftarrow{\rho} U(q)^{k \times k}$
(r, e_1, e_2) $\xleftarrow{r} \chi^k_{n,\eta_1} \times \chi^k_{n,\eta_2} \times \chi^k_{n,\eta_2}$
$\hat{\text{r}} \leftarrow \text{NTT}(r)$
$\text{u} \leftarrow \text{INTT}(A \circ \hat{r}) + e_1$
$\text{v} \leftarrow \text{INTT}(\hat{t} \circ \hat{r}) + e_2 + \lfloor \frac{q}{2} \rceil \cdot m$
$\text{c}_1 \leftarrow \text{Compress}(u, d_u)$
$\text{c}_2 \leftarrow \text{Compress}(v, d_v)$

Algorithm 3: Kyber.INDCPA.DEC

Input: $SK = \hat{s}$
Input: $c_1, c_2 \leftarrow c$
Result: return m $\in M$
$\text{u} \leftarrow \text{Decompress}(c_1, d_u)$
$\text{v} \leftarrow \text{Decompress}(c_2, d_v)$
$\text{m} \leftarrow \text{Compress}_q(v - \text{INTT}(\hat{s} \circ \text{NTT}(u)), 1)$

Algorithm 4: Kyber.CCAKEM.ENC

Input: $PK = (\hat{t}||\rho)$
Result: c $\leftarrow (c_1||c_2)$
Result: Shared key $K \in \{0,1\}^{256}$
$\text{m} \leftarrow \{0,1\}^{256}$
$\text{m} \leftarrow \text{H}(m)$
$(\overline{K}, r) \leftarrow \text{G}(m||\text{H}(PK))$
$\text{c} \leftarrow \text{Kyber.INDCPA.ENC}(PK, m, r)$
$K \leftarrow \text{H}(\overline{K}||\text{H}(c))$

Algorithm 5: Kyber.CCAKEM.DEC

Input: c $\leftarrow (c_1||c_2)$
Input: SK $\leftarrow \hat{s}$
Result: Shared key $K \leftarrow \{0,1\}^{256}$
$\text{PK} \leftarrow SK + 13 \cdot k \cdot \frac{n}{8}$
$\text{h} \leftarrow SK + (13 + d_t) \cdot k \cdot \frac{n}{8} + 32 \in \{0,1\}^{256}$
$\text{z} \leftarrow SK + (13 + d_t) \cdot k \cdot \frac{n}{8} + 64$
$\text{m}' \leftarrow \text{Kyber.CPAPKE.DEC}(SK, (c_1, c_2))$
$(\overline{K}', r') \leftarrow \text{G}(m'||h)$
$\text{c}' \leftarrow \text{Kyber.CPAPKE.ENC}(PK, m', r')$
if $c = c'$ **then**
  $K \leftarrow \text{H}(\overline{K}'||\text{H}(c))$
else
  $K \leftarrow \text{H}(z||H(c))$
end

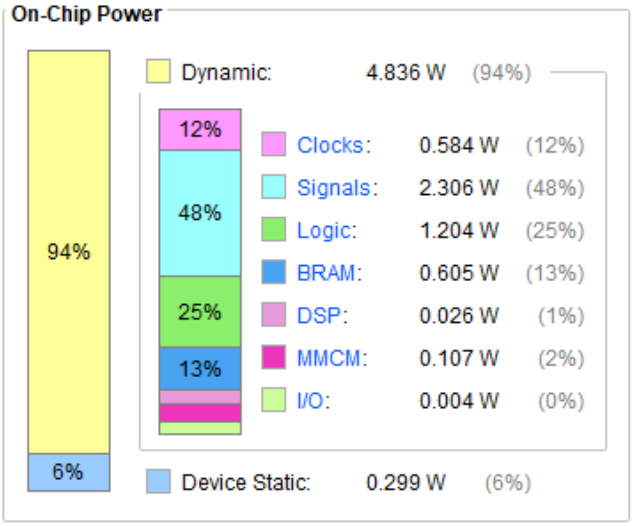


Figure 3: Kyber-512 estimated power consumption report.

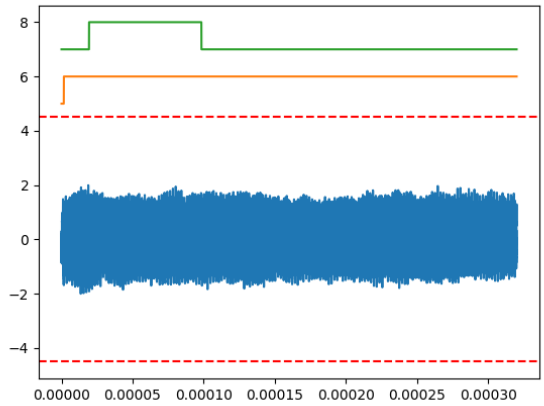
TABLE 2: Kyber-512 resource utilization on VC707 FPGA.

Resource	Available	Utilization	Utilization %
LUT	303600	168953	55.65
FF	607200	143412	23.62
BRAM	1030	264	25.63
DSP	2800	53	1.89

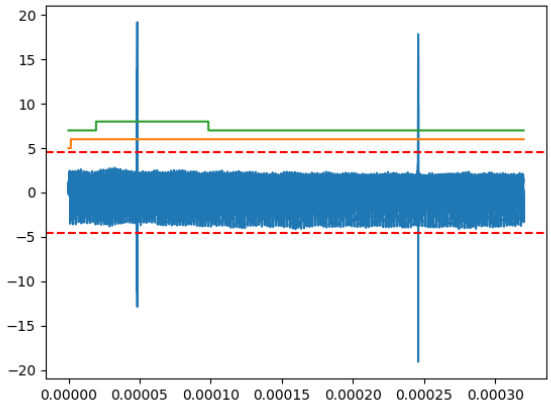
Our implementation is designed around the AXI-Lite IP protocol to take advantage of the AXI bus for communication between host PC and the Kyber512 IP core, which is clocked at 100 MHz. The host PC is equipped with an Intel i7-8700 CPU clocked at 3.2GHz and 16 GB of system memory. A script on the host PC written in Ruby 3.0.0p0 communicates with a control program running on the FPGA's Microblaze soft-core over UART. The control program is responsible for communicating inputs and outputs with the Kyber IP core.

For the purposes of measurement, two trigger signals are placed in the design, one to mark the beginning and the end of the algorithm and another to pick out specific sections of the algorithm like the Number Theoretic Transform (NTT) module. Trace collection is automated. The host PC is connected to the oscilloscope via the USB VISA interface. A Python script is responsible for configuring, arming, and capturing traces from the oscilloscope. The trigger signals from the FPGA enable trace capture when the oscilloscope is armed.

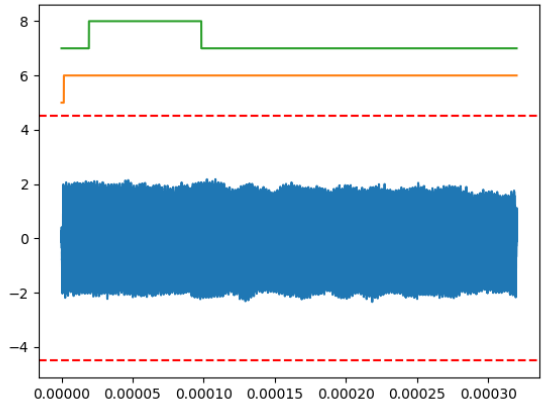
Regarding probe placement on the FPGA board, the power consumption estimation shown in Figure 3, which were given by the Vivado synthesis tool, was used to determine the best locations to probe. According to Figure 3, signals, logic and BRAMs are most likely to draw the most power. The circuitry of those features is powered by VCC_INT, VCCAUX_IO, and VCC_BRAM power rails on the FPGA [26], [34], [35]. From our experiments, we noted that the power rails on the FPGA are not isolated from each other. Hence it is difficult or impossible to guarantee the



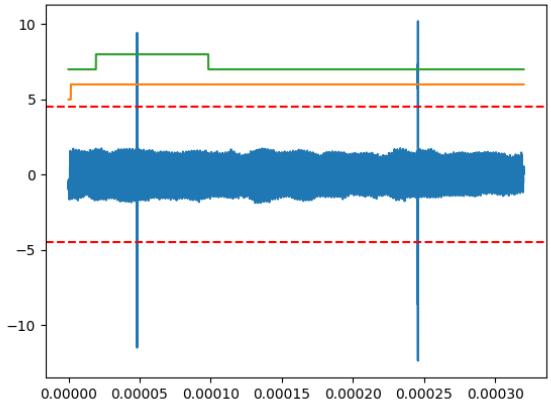
(a) VCCBRAM(I) - Result of fixed vs. fixed TVLA control test. Traces were measured from current output of VCCBRAM power rail.



(b) VCCBRAM(I) - Result from fixed vs. random TVLA. Traces were measure from current output of VCCBRAM power rail.



(c) VCCAUX_IO(I) - Result of fixed vs. fixed TVLA control test. Power traces were measured from current output of VCCAUX_IO power rail.



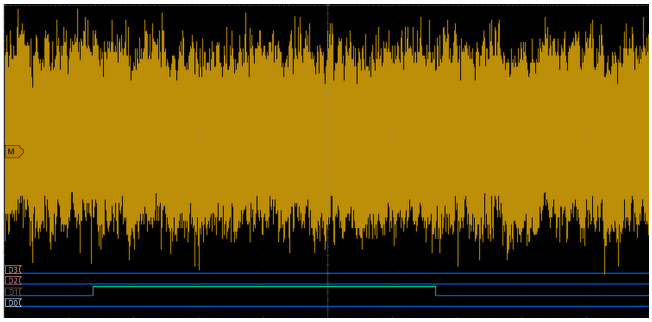
(d) VCCAUX_IO(I) - Result of fixed vs. random TVLA. Power traces were measured from current output of VCCAUX_IO power rail.

Figure 4: The results of non-specific Test Vector Leakage Assessment for Kyber-512 on Xilinx Virtex-7 FPGA platform. The x-axis shows time in seconds. Yellow and green lines are digital trigger signals.

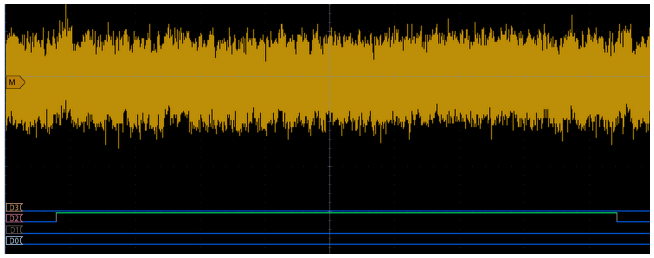
independent leakage assumption. The current draw output of the VCCBRAM and VCCAUX_IO were selected for measurement because they appeared to draw the most power.

In addition, two different software implementations of Kyber were analyzed. Namely, the implementation provided by the "pqm4" repository of Kyber-512, and a masked version of the Kyber-512 decryption based on work from Pessl and Prokop [36]. The Pessl implementation was converted from the C++ implementation to C to interoperate with the "pqm4" software, and was selected because it features a masked decoder in the decryption. The primary motivation for this masking is to prevent the Fujisaki-Okamoto (FO) transformation from leaking information. The FO transformation is performed in decapsulation and has been found to reveal information in schemes that do not use error correcting codes. It has been shown that through this leak, a plaintext checking oracle can be instantiated [37]. Therefore, it is **crucial** that masking can properly circumvent this attack, and it follows that any candidates that use the FO transformation will be similarly affected.

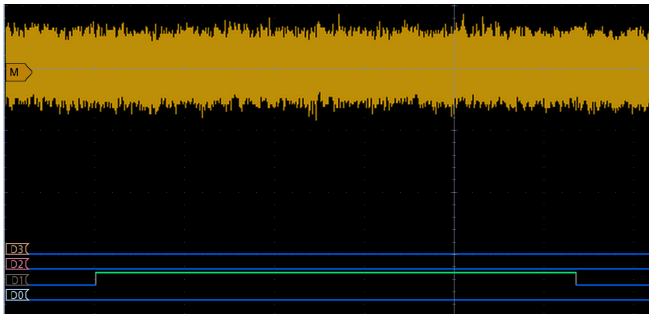
The power traces of the non-masked Kyber-512 software implementation are illustrated in Figures 5a and 5b. Figure 4 shows the preliminary result of the non-specific t-test for Kyber-512 on the FPGA. The result was obtained from 4000 traces (2000 fixed input traces vs. 2000 random input traces). A control experiment is set up with fixed vs. fixed inputs that we expected to show no leakage. Another experiment with fixed vs. random inputs was also setup to identify the



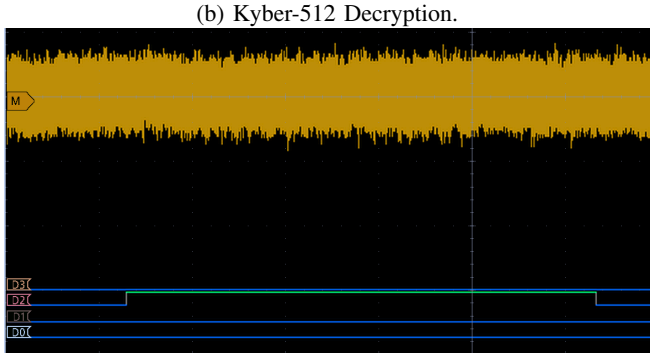
(a) Kyber-512 Encryption.



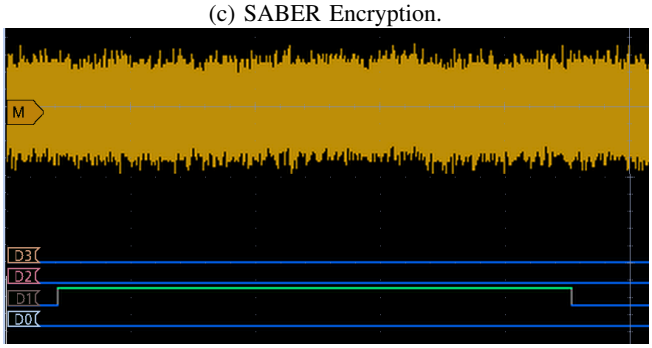
(b) Kyber-512 Decryption.



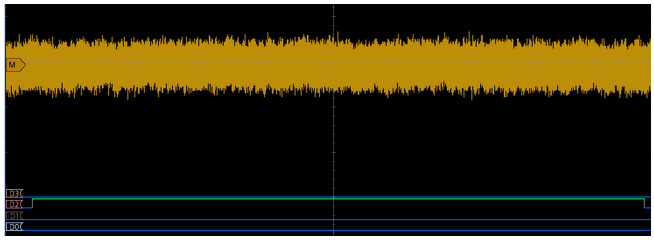
(c) SABER Encryption.



(d) SABER Decryption.



(e) NTRU Encryption.



(f) NTRU Decryption.

Figure 5: Power traces of software implementations of PQC algorithms collected on oscilloscope.

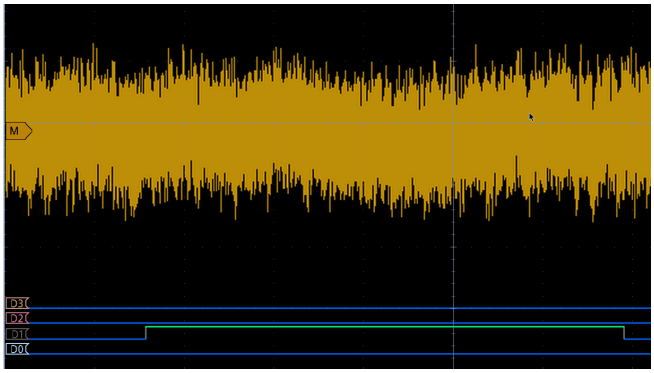


Figure 6: Power trace of software implementation of CRYSTALS-DILITHIUM Sign.

presence of leakage. As shown in figure 4, the results of the control tests for both current outputs on VCCBRAM and VCCAUX_IO showed no leakage. On the other hand, results of the t-test for current measured from both VCCBRAM and VCCAUX_IO are agreement and show presence of some leakage in two locations on the trace. Further analysis is required to identify the specific source of the leakage.

3.3.2. A Few Other PQC Algorithms. Figure 5 also includes the power traces of other two Round 3 finalists, i.e., SABER and NTRU. SABER is a LWR-based KEM. The NTRU implementation is "ntruhps2048509". In addition, the power trance of the signature scheme CRYSTALS-DILITHIUM is shown in Figure 6. We are working on the implementations of masked versions of these algorithms. Then we will evaluate their security robustness based on power analysis.

4. Conclusions

Power-based side channel attacks can be used to reveal the leakage of cryptography algorithms. Therefore, it is critical to perform a thorough power analysis on implementations of Round 3 PQC finalists and alternate candidate algorithms to evaluate their security robustness. In this work, we have built a multi-target and multi-tool platform to collect power traces from both hardware and software implementations of PQC algorithms using oscilloscope and commodity ChipWhisperer-Lite. This platform provides us the capability to perform power analysis on PQC algorithms based on Test Vector Leakage Assessment (TVLA).

Given the traces from both fixed and random inputs on the hardware implementation of Kyber-512, TVLA has shown the presence of some leakage at two locations on the traces. We are in the process of applying TVLA on hardware and software implementations of other Round 3 PQC algorithms to demonstrate the presence of leakages and identify their specific sources.

Acknowledgment

This work was supported in part by NIST Award 60NANB20D016. We also like to thank Daniel Apon of NIST for his help on project discussion and manuscript writing.

References

[1] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016, vol. 12.

[2] U. Congress, "Federal information security modernization act of 2014," *Public Law*, pp. 113–283, 2014.

[3] D. Moody, "Let's get ready to rumble. the nist pqc "competition"," in *Proc. of First PQC Standardization Conference*, 2018, pp. 11–13.

[4] S. H. Standard, "Federal information processing standard (fips) 180-2," *National Institute of Science and Technology*, 2002.

[5] C. F. Kerry and C. R. Director, "Fips pub 186-4 federal information processing standards publication digital signature standard (dss)," 2013.

[6] G. Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta *et al.*, *Status report on the first round of the NIST post-quantum cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology . . . , 2019.

[7] R. McEliece, "A public key cryptosystem based on algebraic coding theory," 1978.

[8] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM," in *Proceedings - 3rd IEEE European Symposium on Security and Privacy, EURO S and P 2018*. Institute of Electrical and Electronics Engineers Inc., jul 2018, pp. 353–367.

[9] J. Hoffstein, J. Pipher, and J. Silverman, "Ntru: A ring-based public key cryptosystem," in *ANTS*, 1998, pp. 267–288.

[10] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, "Saber. proposal to nist pqc standardization, round2, 2019."

[11] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Dilithium : A Lattice-Based Digital Signature Scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, feb 2018. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/839

[12] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon: Fast-fourier lattice-based compact signatures over ntru.(2018)," *Submission to the NIST PQC project*, vol. 3, 2018.

[13] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Lecture Notes in Computer Science*, vol. 3531. Springer Verlag, 2005, pp. 164–175. [Online]. Available: https://link.springer.com/chapter/10.1007/11496137_12

[14] N. Aragón, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Güneysu, C. Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J. Tillich, and G. Zémor, "Bike: Bit flipping key encapsulation," 2017.

[15] C. A. Melchor, O. Blazy, J. Deneuville, P. Gaborit, and G. Zémor, "Efficient encryption from random quasi-cyclic codes," *CoRR*, vol. abs/1612.05572, 2016. [Online]. Available: http://arxiv.org/abs/1612.05572

[16] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, and I.-C. Bourges, "Hamming quasi-cyclic (hqc)," *NIST PQC Round*, vol. 2, pp. 4–13, 2018.

[17] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE," in *Proceedings of the ACM Conference on Computer and Communications Security*, vol. 24-28-October-2016. New York, NY, USA: Association for Computing Machinery, oct 2016, pp. 1006–1018. [Online]. Available: https://dl.acm.org/doi/10.1145/2976749.2978425

[18] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "Ntru prime: reducing attack surface at low cost," in *International Conference on Selected Areas in Cryptography*. Springer, 2017, pp. 235–260.

[19] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The sphincs+ signature framework," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2129–2146.

[20] A. Casanova, J.-C. Faugere, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem, "Gemss: A great multivariate short signature," *Submission to NIST*, 2017.

[21] N. P. Standardization, "Pqc third round candidate announcement," 7 2020. [Online]. Available: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

[22] S. Chowdhury, A. Covic, R. Y. Acharya, S. Dupee, F. Ganji, and D. Forte, "Physical security in the post-quantum era: A survey on side-channel analysis, random number generators, and physically unclonable functions," *arXiv preprint arXiv:2005.04344*, 2020.

[23] M. R. Albrecht, A. Deo, and K. G. Paterson, "Cold boot attacks on ring and module lwe keys under the ntt," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 173–213, 2018.

[24] Z. Xu, O. Pemberton, S. S. Roy, and D. Oswald, "Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber," Cryptology ePrint Archive, Report 2020/912, 2020. https://eprint. iacr. org . . . , Tech. Rep.

[25] A. Aysu, Y. Tobah, M. Tiwari, A. Gerstlauer, and M. Orshansky, "Horizontal side-channel vulnerabilities of post-quantum key exchange protocols," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2018, pp. 81–88.

[26] Xilinx and Inc, "Xilinx XTP135 – VC707 Schematics (Rev 1.0)," Tech. Rep. [Online]. Available: http://www.xilinx.com/warranty.htm.

[27] ——, “Xilinx XTP135 – VC707 Schematics (Rev 1.0),” Tech. Rep. [Online]. Available: http://www.xilinx.com/warranty.htm.

[28] M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, “PQM4: Post-quantum crypto library for the ARM Cortex-M4,” https://github.com/mupq/pqm4.

[29] B. J. Gilbert Goodwill, J. Jaffe, P. Rohatgi *et al.*, “A testing methodology for side-channel resistance validation,” in *NIST non-invasive attack testing workshop*, vol. 7, 2011, p. 115–136.

[30] D. Heinz and T. Pöppelmann, “Combined fault and dpa protection for lattice-based cryptography,” Cryptology ePrint Archive, Report 2021/101, 2021, urlhttps://eprint.iacr.org/2021/101.

[31] M. V. Beirendonck, J.-P. D’Anvers, and I. Verbauwhede, “Analysis and comparison of table-based arithmetic to boolean masking,” Cryptology ePrint Archive, Report 2021/067, 2021, urlhttps://eprint.iacr.org/2021/067.

[32] Y. Huang, M. Huang, Z. Lei, and J. Wu, “A pure hardware implementation of crystals-kyber pqc algorithm through resource reuse,” *IEICE Electronics Express*, pp. 17–20 200 234, 2020.

[33] D. Heinz and T. P
”oppelmann, “Combined fault and dpa protection for lattice-based cryptography,” Cryptology ePrint Archive, Report 2021/101, Tech. Rep., 2021.

[34] D. Canny, “Power-supply solutions for xilinx fpgas,” *Power*, 2012.

[35] “Vivado Design Suite Properties Reference Guide UG912 (v2018.1),” Tech. Rep., 2018. [Online]. Available: www.xilinx.com

[36] P. Pessl and L. Prokop, “Fault attacks on cca-secure lattice kems,” Cryptology ePrint Archive, Report 2021/064, 2021, https://eprint.iacr.org/2021/064.

[37] P. Ravi, S. S. Roy, A. Chattopadhyay, and S. Bhasin, “Generic side-channel attacks on cca-secure lattice-based pke and kem schemes.” *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 948, 2019.