Horizontal Side-Channel Vulnerabilities of Post-Quantum Key Exchange and Encapsulation Protocols

FURKAN AYDIN and AYDIN AYSU, North Carolina State University
MOHIT TIWARI, ANDREAS GERSTLAUER, and MICHAEL ORSHANSKY,
The University of Texas at Austin

Key exchange protocols and key encapsulation mechanisms establish secret keys to communicate digital information confidentially over public channels. Lattice-based cryptography variants of these protocols are promising alternatives given their quantum-cryptanalysis resistance and implementation efficiency. Although lattice cryptosystems can be mathematically secure, their implementations have shown side-channel vulnerabilities. But such attacks largely presume collecting multiple measurements under a fixed key, leaving the more dangerous single-trace attacks unexplored.

This article demonstrates successful single-trace power side-channel attacks on lattice-based key exchange and encapsulation protocols. Our attack targets both hardware and software implementations of matrix multiplications used in lattice cryptosystems. The crux of our idea is to apply a horizontal attack that makes hypotheses on several intermediate values within a single execution all relating to the same secret, and to combine their correlations for accurately estimating the secret key. We illustrate that the design of protocols combined with the nature of lattice arithmetic enables our attack. Since a straightforward attack suffers from false positives, we demonstrate a novel *extend-and-prune* procedure to recover the key by following the sequence of intermediate updates during multiplication.

We analyzed two protocols, `Frodo` and `FrodoKEM`, and reveal that they are vulnerable to our attack. We implement both stand-alone hardware and RISC-V based software realizations and test the effectiveness of the proposed attack by using concrete parameters of these protocols on physical platforms with real measurements. We show that the proposed attack can estimate secret keys from a single power measurement with over 99% success rate.

CCS Concepts: • **Security and privacy → Embedded systems security;**

Additional Key Words and Phrases: Lattice-based cryptography, side-channel attacks, RISC-V

ACM Reference format:
Furkan Aydin, Aydin Aysu, Mohit Tiwari, Andreas Gerstlauer, and Michael Orshansky. 2021. Horizontal Side-Channel Vulnerabilities of Post-Quantum Key Exchange and Encapsulation Protocols. *ACM Trans. Embedd. Comput. Syst.* 20, 6, Article 110 (October 2021), 22 pages.
https://doi.org/10.1145/3476799

This work is sponsored in part by Lockheed Martin, National Science Foundation (Award #1453806, #1314709, #1527888, #1441484, #1850373, and CCF-1901446), Semiconductor Research Corporation (SRC), and C-FAR: one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

Authors' addresses: F. Aydin and A. Aysu, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 27606, USA; emails: {faydn, aaysu}@ncsu.edu; M. Tiwari, A. Gerstlauer, and M. Orshansky, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, 78712, USA; emails: {tiwari, orshansky}@austin.utexas.edu, gerstl@ece.utexas.edu.

1539-9087/2021/10-ART110 $15.00
https://doi.org/10.1145/3476799

1 INTRODUCTION

Key exchange protocols and key encapsulation mechanisms establish secret keys between two or more parties. While key exchange protocols allow multiple users to jointly decide on the secret key, a single user transmits its secret key in key encapsulation mechanisms. Traditional key exchange and encapsulation protocols such as Diffie–Hellman rely on the difficulty of solving discrete-logarithm problems, which are widely regarded to be infeasible for large numbers. These problems, however, can be solved in polynomial-time with quantum algorithms [59], generating a significant interest in alternative protocols that will future-proof security systems [41] against a rapidly evolving quantum computing technology [5]. Lattice-based cryptography provides a wide array of such constructions that are resistant to quantum acceleration.

While lattice cryptosystems rely on the theory of quantum-resistant lattice problems [55], their practical implementations have shown vulnerability against power or **electromagnetic** (**EM**) side-channel attacks in the context of public-key encryption or digital signatures [3, 4, 6, 8, 14, 16, 21, 29–32, 45–48, 51, 52, 54, 60–62, 66]. These attacks find the secret key by exploiting the reflection of key-dependent computations on the power/EM consumption. Side-channel attacks are indeed dangerous because they can drastically reduce the number of tests needed to steal a secret key from a cryptosystem. Moreover, side-channel attacks do not need a quantum computer to succeed. We can broadly categorize power side-channel attacks into three groups: **simple power analysis** (**SPA**), **differential power analysis** (**DPA**), and template attacks. SPA is based on purely visual observations to capture large variations related to the secret key with a few traces. By contrast, DPA extracts small correlations by evaluating many traces with statistical methods. We refer to attacks that require a pre-characterization of the target device's power profile for the given application as template attacks [19].

To extract secret-key dependence, prior attacks on lattice problems typically use repeated computations performed with the same secret key. To that end, DPA [34] is a potent side-channel attack that works with a divide-and-conquer principle: It makes an estimation on distinct parts of the key (called *sub-key*) and checks those estimations through multiple tests. These tests are required to remove the noise and reveal the underlying correlation between the sub-key and the power measurement. Once the target sub-key is derived, the adversary can attack the next sub-key by using the same set of measurements. Applying DPA is, however, significantly more challenging on key exchange protocols because these protocols, unlike public-key decryption or digital signatures, work with *ephemeral* secrets: each invocation of the protocol will process a unique value that will result in a new, distinct secret key. Therefore, the adversary is limited to a single power measurement. While key encapsulation mechanisms do not have this limitation, it is still crucial to explore single-trace attacks on them as these attacks do bypass conventional defenses such as masking [51].

Figure 1(a) illustrates the classic DPA, known as *Vertical* DPA. In this attack, the adversary performs a single test on the power trace and collects multiple measurements (each with a different input) to extract sub-key correlations from noise. The intermediate computations on the variable var_1 is selected as the target because its value depends on the public input and a sub-key. *Horizontal* DPA [20], Figure 1(b) , by contrast, performs multiple tests by targeting different computations and combines them to extract the sub-key from a single power measurement. Horizontal DPA thus focuses on intermediate computations that use variables var_i $(i \in 1, 2, \ldots, n)$ that all depend on the public input and the same sub-key. The main challenge of applying Horizontal DPA is in finding multiple tests to be performed within a single computation that will leak the same sub-key. Such approaches have been successfully applied to modular exponentiation [20] and elliptic curve

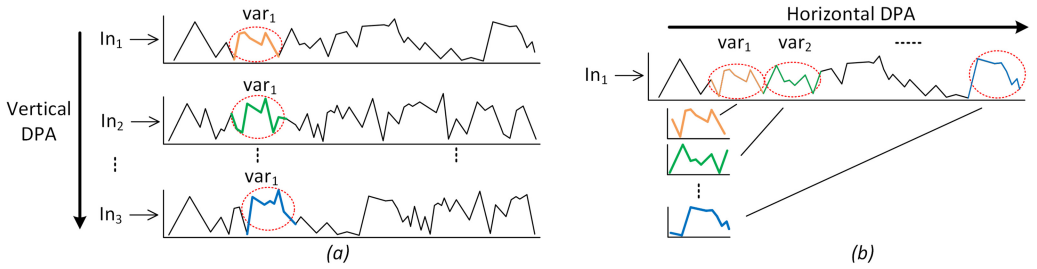

Fig. 1. Vertical DPA (a) targets a single intermediate computation and seeks correlation across multiple traces each using a distinct input, while Horizontal DPA (b) targets multiple intermediate computations within a trace and seeks a correlation among them.

multiplication [12], but lattice arithmetic is based on fundamentally different computations for which the horizontal side-channel vulnerabilities are unknown.

In our prior work [9], we demonstrated, for the first time, that lattice arithmetic used in key exchange protocols is vulnerable to Horizontal DPA attacks.[1] Specifically, we demonstrated that the matrix multiplication used in these protocols have a large number of intermediate computations that depend on the same sub-key. A straightforward attack on these operations, however, causes false positives because similar sub-keys will produce similar multiplication output (e.g., sub-keys of "1" and "2" will generate the same values shifted by one binary digit). We showed that an adversary can address this limitation and still succeed by using a novel attack that targets intermediate state updates of these multiplications starting from the first sub-key to successively recover a chain of subsequent sub-keys. We validated that such an attack effectively removes false positives after the first sub-key.

This article extends our prior work by addressing its limitations in the following ways. First, a new variant of the protocol we previously targeted has since emerged. Second, the attack we proposed was focusing on hardware accelerators while the vulnerability of the software-based designs was unexplored. Third, our previous attack required a large number of sub-traces to succeed and thus may not extend to high-noise platforms.

We analyze the feasibility of the proposed attacks not only on a key exchange protocol (`Frodo` [15]) but also a key encapsulation mechanism (`FrodoKEM` [2]), which both rely on the learning with errors problem over generic, algebraically unstructured lattices. While `Frodo` was the original algorithm implementing a key exchange protocol, `FrodoKEM` is its key encapsulation variant submitted to the NIST post-quantum cryptography standardization process. These protocols are designed to be "conservative (in security level) yet practical (in implementations)" making them high profile candidates. Indeed, the original `Frodo` algorithm has been used in HTTPS/TLS connections and its NIST submission has passed to the final, Round-3 phase as an alternative candidate. Moreover, `FrodoKEM` is currently being recommended by the German Federal Office for Information Security as a desired algorithm to achieve quantum-safe communications [23]. Since KEMs have a dynamic generation mechanism for every key, they are useful for TLS handshakes with perfect forward secrecy [2]. Both `Frodo` and `FrodoKEM` protocols perform matrix multiplications between a secret ephemeral value and a public input, making them susceptible to the proposed attacks. Once the ephemeral secret is discovered with the proposed attack, we show that an adversary can recover the secret key in these protocols. We, furthermore, show that both parties engaged in the key exchange protocol are vulnerable to our attack.

[1]Our earlier work also studies the polynomial multiplication in `NewHope` but it is dropped from this article to focus on Round-3 NIST candidates and to candidates that do not use NTT based multiplications.

To test the practical validity of our attacks, we design the matrix multiplication in both hardware and software realizations using the specific parameter sets of the two protocols, and we apply the attack using real power measurements taken from a SAKURA-G FPGA platform. We validate that the number of horizontal tests, which can be 752 for Frodo and 1344 for FrodoKEM, is sufficient to extract the key. The results show that the proposed attack is able to recover secret key coefficients with over 99% probability for both hardware and software realizations. Our earlier methodology, which we now extend to FrodoKEM and RISC-V based software implementations, had been verified by third-party evaluators in a peer-reviewed publication [16].

Finally, we improve our previous attack [9] by pre-characterizing the power profile of the device. This enhanced attack configures the device with all possible sub-keys and uses template profiling [19] to extract the target device's power measurement behavior, which is then compared with sub-traces of the unknown sub-keys to estimate their value. We augment the efficiency of the template attack with a **Sum of Squared Differences** (**SOSD**) technique to rank and select the time samples that carry more information for the side-channel analysis. Compared with our prior attack, although the enhanced version is less generic, i.e., tuned for the specific target device, it requires fewer sub-traces and thus can also succeed on higher-noise platforms.

In summary, we make the following extensions and contributions in this article:

- We have implemented designs using the latest available Round-3 specification of FrodoKEM and evaluated them under our horizontal attacks. We analyze FrodoKEM and demonstrate that it has a similar vulnerability and can be successfully attacked with our technique.
- We have implemented the targeted computations of FrodoKEM in software and ported it on a RISC-V based softcore processor on the SAKURA-G FPGA. We analyze the generated software assembly code for the RISC-V **Instruction Set Architecture** (**ISA**) and reveal the vulnerability of the software implementation to the proposed attacks. Subsequently, we repeat the horizontal DPA attack on the software realization's power measurements and quantify that the attack still works with a single trace but needs 76% more intermediate computations to achieve the same accuracy.
- We have developed a new attack based on template profiling on the software realization and compared its effectiveness to the horizontal DPA. The results show that the template attack can reduce the number of tests needed to extract a secret key coefficient by 88.6%.

The rest of the article is organized as follows. Section 2 describes the threat model of DPA and provides a background on power side-channels as well as the related work. Section 3 analyzes the protocols under attack. Section 4 outlines the proposed attack. Section 5 shows the details of hardware architectures used in analysis and evaluates the attack efficiency with measurements. Section 6 extends the proposed attacks to a software realization on a RISC-V based softcore processor on the same FPGA and develops a template attack enhancement over horizontal DPA. Section 7 discusses the limitations and countermeasures of our attack and Section 8 concludes the article.

2 THREAT MODEL AND RELATED WORK

In this section, we cover the background material related to our threat model and prior work.

2.1 Threat Model

Our threat model follows the typical power side-channel model of prior work [6, 21, 34, 43, 46, 47, 51, 53, 56, 57, 61, 66]. The adversary in DPA attacks has physical access to the device and can read power measurements as the device computes the cryptographic routine. Therefore, the adversary is equipped with a reasonable measurement setup that can obtain power consumption information several times within a clock period. In our experimental setup, which targets a device operating in

the MHz frequency range, a low-end digital oscilloscope with passive probes is sufficient to apply the attack.

We assume that the attacker in our model can record a single power measurement of the entire application (e.g., HTTPS/TLS) that uses the key exchange protocol or key encapsulation mechanism. We also assume that the DPA adversary knows details of hardware architecture, such as its data flow, parallelization and pipelining, and the details of the software such as its generated assembly. Therefore, unless there is a specific countermeasure to obfuscate the timing of the underlying operations, the adversary can estimate when the targeted computations will *likely* occur and apply the attack around those clock cycles. Engineering aspects of locating cryptographic routines (and specific operations inside those routines) among other applications within a system have been described in prior work, both in the context of physical [10] and digital side-channels [26, 65]. We do not cover this aspect in the scope of this work and assume that the adversary can locate around which clock cycle to start the DPA analysis using prior techniques. Note that the adversary does not have to know the exact timing information of side-channel leaks within the clock period; it can exhaustively evaluate the attack on all sampling points within a period to identify where the side-channel leak occurs.

We follow the assumption that the DPA adversary can eavesdrop on the communication to record the public messages that are transmitted between parties involved in the protocol. We conduct the DPA attack on implementations with no conditional checks based on the value of sub-keys. Therefore, the attacker cannot use timing side-channel leakage due to conditional branches. Yet the DPA can still extract information from the data flow variations.

2.2 Related Work

Although power-based side-channel attacks have been an active area of research for the last two decades [34], analysis of lattice-based constructions is relatively limited [3, 4, 6, 8, 14, 16, 21, 29, 31, 32, 43, 45–48, 51, 52, 54, 56, 57, 60–62, 66]. The scope of this work is power side-channels but we will include EM side-channels in the related work as well since they both stem from the same source of data-dependent **Complementary Metal Oxide Silicon** (**CMOS**) switching activity. These earlier attacks typically consider other applications with lattices such as public-key decryption or digital signatures while executing multi-trace attacks. In the conference version of this paper, we demonstrated, for the first time, that lattice arithmetic used in key exchange protocols is vulnerable to Horizontal DPA attacks that require a single power/EM measurement trace to break the system [9]. But our earlier approach had three limitations. First, the algorithm we analyzed has been changed when it was submitted for the NIST's post-quantum standardization contest. Second, we had exclusively focused on hardware accelerator design, leaving the feasibility of the attack on software implementations unknown. Third, we only had implemented a DPA-style attack, leaving the capabilities of more powerful, template-based attacks unexplored. We therefore extend our previous conference paper in this work by addressing these three limitations.

Earlier *single-trace attacks* focus on other components such as the **Number Theoretic Transform** (**NTT**) [32, 48, 51], discrete Gaussian sampling [21, 33, 64], rejection [45], message encoding/decoding [3, 42, 54, 60, 62], and other related components used in lattice cryptosystems such as the hash function [42, 54]. These attacks are orthogonal to our work. Single-trace attacks targeting multiplication corroborates the results of our conference paper [16] and extends such attacks to digital signature schemes [53] or other key encapsulation candidates [25]. Our work, furthermore, evaluates the security of shuffling against single-trace attacks on lattice cryptography multiplications. By contrast, earlier defenses on lattice-based cryptosystems primarily applies masking [43, 56, 58] which is known to be insufficient against single-trace attacks [51].

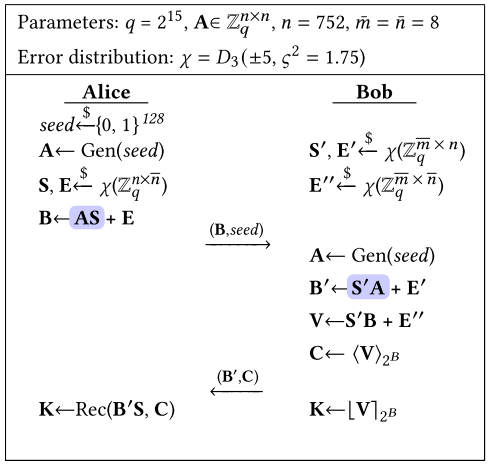

Fig. 2. Frodo key exchange protocol and the parameters we select for its instantiation. We highlight the target of DPA attack; the ephemeral secrets (**S** and **S′**) are multiplied with a public value (**A**).

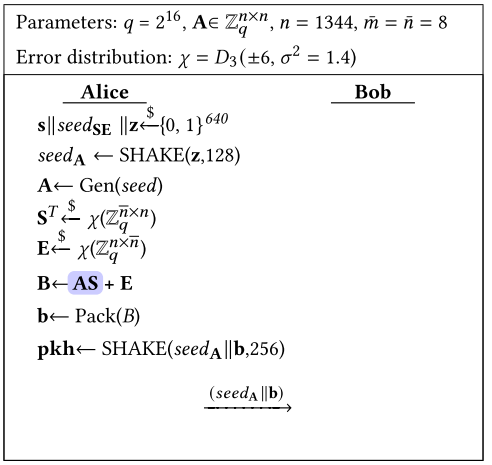

Fig. 3. FrodoKEM key generation and the parameters we select for its instantiation. We highlight the target of DPA attack; the ephemeral secret (**S**) is multiplied with a public value (**A**).

3 THE TARGETED POST-QUANTUM PROTOCOLS

In this section, we summarize the key exchange protocol and the key encapsulation mechanism that we attack.

3.1 Frodo: Post-Quantum Key Exchange Protocol

Key exchange protocols establish a unique, symmetric key between two or more parties.[2] Both parties in these protocols use an ephemeral secret to generate some public information and share it with the other party to successfully agree on the same key. The protocol is designed in such a way that the adversary who eavesdrops on the public information cannot capture the established key or recover the ephemeral secret values.

Figure 2 gives the description of Frodo, which relies on the **learning-with-errors** (**LWE**) problem. The algorithm deliberately picks LWE over R-LWE in case there is a future reduction in the assumed difficulty of underlying R-LWE problems (this can occur due to the algebraic structure in R-LWE). Therefore, Frodo works with matrices rather than polynomials. The matrix symbols are denoted with capital Latin letters.

Alice starts by generating a public parameter (**A**) from a random *seed*, sampling the secret error terms (**S** and **E**) from a specific distribution, and computing the message (**B**), which is sent, together with the *seed* to Bob. Given **B** and **A**, it should not be feasible, either with a conventional or a quantum computer, to compute the values of small error terms **S** and **E** due to the LWE problem [55]. Bob then generates his share of the secret key (**B′**) by using his ephemeral error samples (**S′** and **E′**) and sends it back to Alice. Alice and Bob now can agree on the secret value by evaluating each others' terms with their ephemeral secret shares. Since Alice and Bob will achieve a similar but noisy term, they can *reconcile* by recovering from this noise through a thresholding scheme. Interested readers can refer to the detailed description of the protocols [15].

Frodo has several parameter options depending on the desired security level. For our analysis, the parameters for the Frodo are selected from the "Recommended" scheme [15]. This set uses

[2]Frodo is defined for two parties, hence we cover two party key exchange.

matrices of sizes $n \times n$, $n \times \bar{n}$, $\bar{m} \times \bar{n}$, and $\bar{m} \times \bar{n}$, where n, $\bar{n}$, and $\bar{m}$ are, respectively, 752, 8, and 8 with integer elements modulo 2^{15}. The error distribution is a Rényi divergence approximation to a rounded continuous Gaussian with variance $\varsigma^2 = 1.75$, which is centered at 0 and has the output range from −5 to 5.

3.2 FrodoKEM: Post-Quantum Key Encapsulation Mechanism

Key encapsulation mechanisms too establish symmetric keys between two parties. But the difference is that in key encapsulation protocols, a single party decides on the secret key and transmits it securely by encrypting/encapsulating it with the destination's public key. The designated party then can use its secret key to decapsulate and obtain the secret key send for the session.

Figure 3 illustrates the key generation steps of FrodoKEM. The algorithm does not take any inputs and generates the public key for the encapsulation and the secret key for the decapsulation processes. The scheme first picks random seeds for the error terms ($\mathbf{s}$ and $seed_{\mathbf{SE}}$), and the public values ($seed_{\mathbf{A}}$), and uses them to generate the error matrices ($\mathbf{S}^T$ and $\mathbf{E}$) and the large matrix ($\mathbf{A}$) for the LWE instance ($\mathbf{B} = \mathbf{AS} + \mathbf{E}$). The packed value of $\mathbf{B}$ is published along with the seed of $\mathbf{A}$ ($seed_{\mathbf{A}}$) as the public key, while the secret key components are $\mathbf{s}$, ($seed_{\mathbf{A}}$), $\mathbf{b}$, $\mathbf{S}^T$, and $\mathbf{pkh}$.

FrodoKEM has multiple parameter options based on the desired security level. We chose FrodoKEM-1344 for our analysis which is the most secure version and its security matches or exceeds the brute-force security of AES-256 [2].

FrodoKEM-1344 uses matrices of sizes $n \times n$, $n \times \bar{n}$, $\bar{m} \times \bar{n}$, and $\bar{m} \times \bar{n}$, where n, $\bar{n}$, and $\bar{m}$ are, respectively, 1344, 8, and 8 with integer elements modulo 2^{16}. The error distribution is a Rényi divergence approximation to a rounded continuous Gaussian with variance σ^2=1.4, which is centered at 0 and has the output range from -6 to 6.

Note that we chose the most theoretically-secure versions of both Frodo and FrodoKEM, yet these will interestingly turn out to be the least secure versions for side-channel analysis as we describe in Section 4.

4 THE PROPOSED SIDE-CHANNEL ATTACKS

In this section, we highlight the target operation of our Horizontal DPA. Then, we give a high-level description of our attack and discuss how to address its challenges.

4.1 Protocol Operations Under Attack

Figure 2 highlights the possible points of attacks for our Horizontal DPA on Frodo. We focus on the multiplication of public value $\mathbf{A}$ with the ephemeral secret $\mathbf{S}$ or $\mathbf{S}'$. The target operation for Frodo is therefore a matrix multiplication of the public value $\mathbf{A}$ with the ephemeral secret $\mathbf{S}$ or $\mathbf{S}'$. This is multiplication of a size 752×752 matrix with a size 752×8 or 8×752 matrix.

Note that both Alice and Bob in Frodo can be attacked using this approach. Once the adversary recovers the ephemeral secret, extracting the exchanged secret key becomes trivial. For Frodo, an adversary attacking Alice can compute $\mathbf{K}$ =Rec($\mathbf{B'S}$,$\mathbf{C}$) and an adversary attacking Bob can first recover $\mathbf{V}$ (without the small error of $\mathbf{E''}$), and then generate $\mathbf{K} = \lfloor \mathbf{V} \rceil_{2^B}$ since $\lfloor \mathbf{V} \rceil_{2^B} = \lfloor \mathbf{V} - \mathbf{E''} \rceil_{2^B}$.

Figure 3 highlights the possible point of attack for FrodoKEM. Our attack, again, aims at the matrix multiplication of the public value $\mathbf{A}$ with the secret key's component $\mathbf{S}$. This time, however, the size of the matrices are larger and hence each element of $\mathbf{S}$ gets multiplied with 1344 coefficients of $\mathbf{A}$. If the adversary recovers $\mathbf{S}$, it can repeat the sequence of decapsulation operations and extract the shared secret $\mathbf{ss}$ in FrodoKEM [2] for all sessions using a valid ciphertext. Since each key generation will use fresh randomness, the attack still has the single-trace limitation. Using this attack, encapsulation or decapsulation parts of FrodoKEM can also be targeted. The decapsulation may

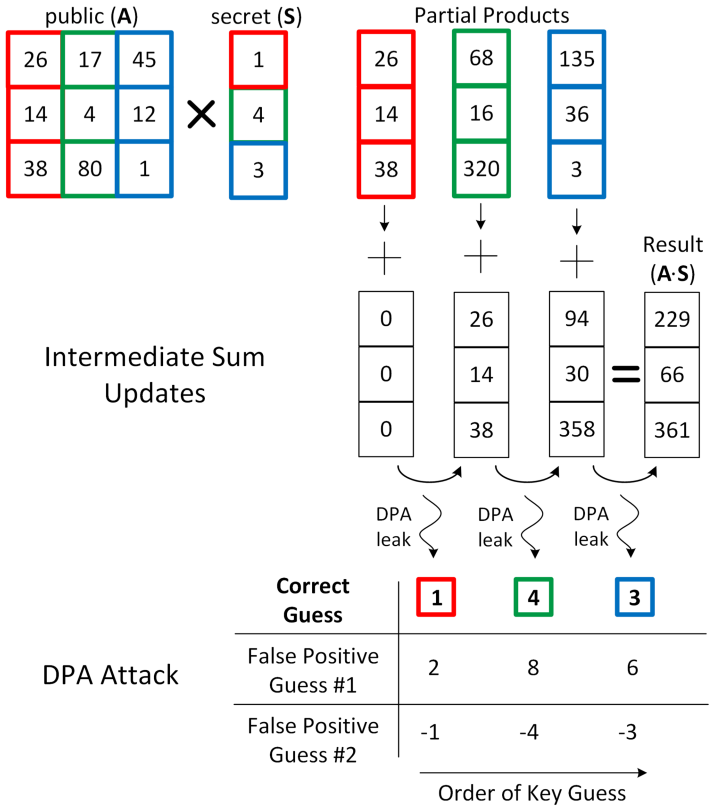

Fig. 4. An example of Horizontal DPA attack on the matrix multiplication. The correct guess reveals the coefficients of the secret matrix **(S)** by targeting the intermediate sum updates.

use the same secret key more than once but the single-trace attacks are still valuable as they can bypass masking style defenses that intends protecting multi-trace attacks on decapsulation [48].

We do not claim that the single-trace vulnerabilities are only at the matrix multiplication. Indeed, other parts of the algorithm such as the hash function [27] or message encoding [60] can also be vulnerable to singe-trace attacks but these are orthogonal threats that have already been shown and thus are out of our scope.

4.2 The Crux of Horizontal DPA on Multiplication

Figure 4 illustrates the main idea behind our Horizontal DPA attack on the matrix multiplication. For simplicity, this example uses a matrix of size 3×3 and 3×1. The figure reflects the matrix multiplication of the public matrix (e.g., **A**) with an ephemeral secret matrix (e.g., **S)**. The matrix multiplication has two main parts:

(1) **Generating Partial Products:** Each column of multiplication in Figure 4 corresponds to the product of a single secret coefficient of **S** with all coefficients of public matrix **A**; we refer to this method as column-wise multiplication. We omit the reduction modulo q again and negative numbers for simplicity.

(2) **Updating Intermediate Sum:** The result of matrix multiplication is the addition of all column-wise computations. Therefore, after a partial product is computed, its value has to be accumulated into the intermediate sum that holds the result of previous column's

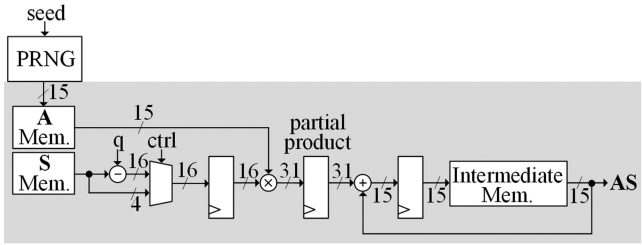

Fig. 5. Hardware architecture of operations under attack for Frodo.

computations. After all columns are processed, the value of the intermediate sum will be the result of this operation.

Figure 4 highlights, in bold red, the first column of computations and its dependence on the first secret coefficient (second and third column computations are depicted in green and blue, respectively). The crux of our attack is to observe that all of these column operations rely on the same coefficient of the secret matrix. The adversary, therefore, can effectively apply a Horizontal DPA using these operations: it can make a hypothesis guess on a secret coefficient, compute the hypothesis value for each intermediate computation, and test correlations between those values and corresponding activity within the single power trace.

The main problem of applying DPA on the target lattice-based constructions is the false positives of multiplication. Unlike the case of AES or other block ciphers where an S-BOX maximally diffuses similar inputs, multiplications with similar values generate a correlated output. For example, if a secret coefficient is "1" vs. "2", the output of the multiplication will be shifted by one binary digit, resulting in an output having the same Hamming weight unless there is a modular reduction. Even when there is a modular reduction, the output changes by a single overflow bit if the modulo value is a *power of two*, which is exactly the case for the modulo 2^{15} and 2^{16} multiplication of Frodo and FrodoKEM, respectively.

The solution to this problem is to target the intermediate sum update and to extract one key coefficient at a time, successively, starting from the first coefficient of the key. Since the intermediate sum for every coefficient is "0" in the first row, attacking these sums will be equivalent to attacking partial product generation, hence resulting in false positives. However, due to the modular reductions, after the first row, there will only be a single guess that yields a high correlation for each previous distinct guess. Therefore, our attack will form a *string* of possible keys, rather than forming a tree having multiple independent false positives at each row of computations.

We elaborate on this phenomena with an example. For instance, assume that the operations are modulo 256, the first and second known coefficients are 100 and 100, and the first and second secret coefficients are 1 and 1, respectively. The first row multiplication then returns 100 ($100 \times 1 = 100$). Since the intermediate memory initially stores 0, the intermediate memory update also returns 100 ($100 + 0 = 100$). Therefore, when targeting first computations, secret coefficient guesses between "1" vs. "2" causes false positives (100 vs. 200 has the same Hamming weight). For the second row, the true multiplication results will be 100 and the intermediate update will be 200 ($100 + 100$). But the guessed multiplication result for the guessed secret coefficient "2" will be 200 and the intermediate update will be 54 ($100 + 200 = 54 \bmod 256$). Thus, the predicted activity of switching from 100 to 200 vs. switching from 100 to 54, corresponding to the guessed secret coefficients of 1 vs. 2, will have different Hamming distances (i.e., the attacker will no longer observe the false positive). Note that many other changes in the datapath and control signals occur along with the targeted change of the accumulator register. But these changes add noise to the measurement, they do not eliminate the observed leakage.

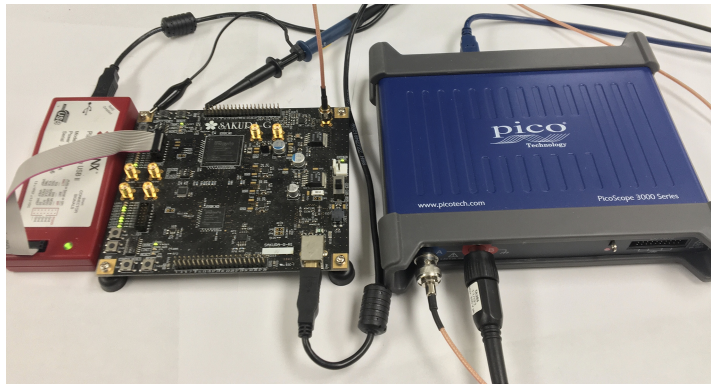

Fig. 6. The evaluation setup for the side-channel experiments with the Picoscope 3206D oscilloscope.

In Figure 4, the correct string of guesses is $1 \rightarrow 4 \rightarrow 3$, whereas the false strings are $2 \rightarrow 8 \rightarrow 6$ and $-1 \rightarrow -4 \rightarrow -3$. There can only be k strings in total, where k is the number of true and false positives at the first coefficient (i.e., 11 strings at the worst-case for Frodo). The correct key can be brute forced among these k possible options.

The number of possible horizontal tests depends on the size of the matrices. Since the columns $\mathbf{A}$ in Frodo have 752 coefficients, our attack can perform 752 tests for each key guess. For the FrodoKEM key generation, our attack can conduct 1344 tests.

Note that our attack does not require enforcing specific patterns or values in the input (as in, e.g., Ravi et al. [54], Fouque et al. [24], and Xu et al. [62]) to estimate the secret key. This corresponds to a known-plaintext attack instead of a chosen-plaintext. An adversary using our attack, therefore, does not have to invoke or modify an encryption request. Instead, it can simply eavesdrop on the communication (in addition to recording power consumption) to extract the secret key, which is less likely to be detected with a higher-level detector in the system. Our attack is also possible on both column-wise and row-wise matrix multiplications—Sections 5 and 6, respectively, apply the attacks on these settings.

Another important feature of our attack is the implication of using larger keys to improve theoretical security. In both protocols, a future update on the parameter set, which typically occurs due to disclosure of better attacks, can increase the matrix sizes of the keys—this improves the effectiveness of our attack as it increases the number of tests for the horizontal analysis.

5 ATTACKING HARDWARE IMPLEMENTATIONS

In this section, we evaluate the effectiveness of our proposed Horizontal DPA attack on a dedicated hardware implementation on FPGAs using real power measurements.

5.1 The Targeted Hardware Architecture

We primarily focus on resource-constrained applications in embedded devices, such as RFID, smart cards, or IoT nodes. As such, we design and analyze coefficient-serial architectures that compute a single coefficient of multiplication in a clock cycle. This design uses one multiplier as its main processing unit. Therefore, it takes approximately $\bar{m} \times n \times n$ clock cycles to compute a matrix multiplication of $\mathbf{A} \cdot \mathbf{S}$.

Figure 5 shows the details of the hardware architecture for the matrix multiplications. As discussed in Section 4, both Frodo and FrodoKEM uses the same type of multiplications with some differences in the sizes due to parameter specifications. The figure reflects the hardware design for Frodo. The main processing unit is a multiplier to compute the partial products. The result of

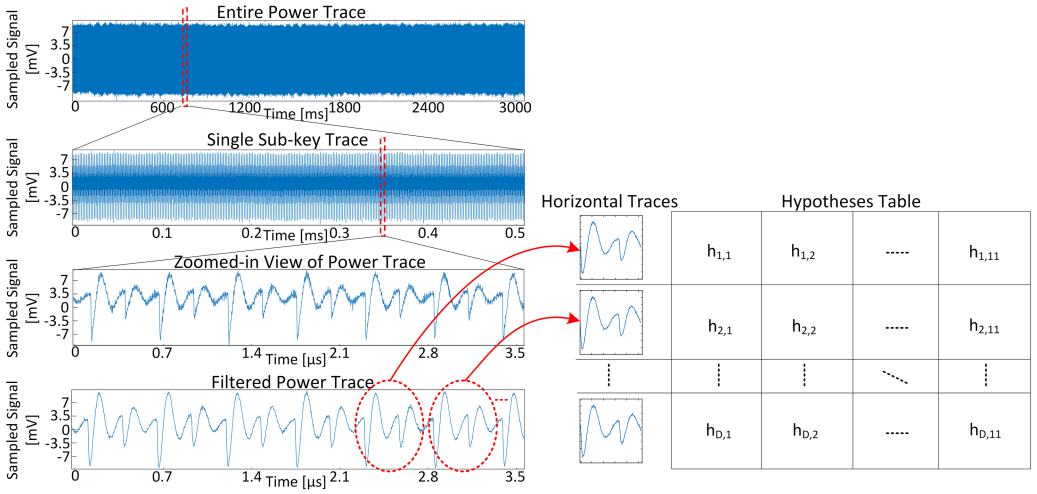

Fig. 7. Pre-processing of power traces for the Horizontal DPA for Frodo and generation of the hypotheses table for one sub-key.

the partial product is accumulated to the previous value stored in intermediate memory, which updates the intermediate sum. Since this instantiation use a modular reduction with a power of two, the modulo operation is free, which is simply truncation of the adder output to $\log_2 q$ bits.

The size of matrix **A** creates a problem for the implementations. Since our target FPGA cannot store the entire **A** due to BRAM limitations, it has to generate parts of it on-the-fly during the computation of **A**·**S**. To do so, our architecture follows the guidelines of Frodo specifications [15], and generates one column of **A** at a time and multiplies it with one row of **S** to compute the partial results for the entire **A**·**S** matrix. Only then, does the hardware generate the next column of **A** and repeat the same process until all the columns of **A** are swept. This approach minimizes the required amount of storage for **A**.

The hardware architecture uses a generic modulo q sign conversion on secret key (**S**) coefficients to handle the sign arithmetic; a similar approach is also taken by prior work on an area-optimized lattice-hardware design [50]. There are two reasons for this approach. First, it allows to mitigate zero-value attacks on lattice arithmetic ([57], Appendix A) when a secret coefficient is equal to 0, and second, it enables achieving a modular design, independent of the size of coefficients. Since the multiplication and additions are mapped to a DSP unit, which can compute up to a 18-bit multiplication and 48-bit addition, converting **S** coefficients into $\log_2 q$ bits does not carry an area overhead.

The hardware architecture we propose is implemented in Verilog HDL and mapped on to the Xilinx Spartan-6 XC6SLX75 FPGA. The synthesis, placement, and routing of the proposed designs to the target FPGA is performed using Xilinx **Integrated Synthesis Environment** (**ISE**) version 14.6.

5.2 Evaluation Setup

To evaluate the power attacks in a real environment, we ported our hardware implementation on the SAKURA-G board, which includes a Xilinx Spartan-6 (XC6SLX75-2CSG484C) FPGA. We will, specifically, evaluate the hardware design for the Frodo algorithm. Figure 6 shows our evaluational setup. We measure the voltage drop on a 1-Ω resistance and make use of the on-board amplifiers on the SAKURA-G platform to measure power consumption. The measurements for DPA analysis

are taken using a low-end Oscilloscope (SDS1102X Digital Oscilloscope) that can sample at 2 ns intervals (500 MS/s) with two active channels. We use the first channel for power measurements and the second channel to trigger the oscilloscope to start recording. The design is clocked at a constant 1.5 MHz operating frequency.

Prior to DPA, the adversary has to pre-process the power measurement and divide it into smaller parts for the DPA targeted operations. Figure 7 shows the entire power trace, which is then zoomed into the regions of interest. We empirically find applying a 20 MHz low pass filter on the measured signal to be very useful as it reduces noise and achieves better detection result. We then divide the power traces into pieces of one clock period, which we refer to as *sub-trace*. To synchronize these divided sub-traces, we find the minimum point within each sub-trace and synchronize by fixing that point in the sub-trace to a certain clock index. Moreover, to avoid missing sub-traces, we divided the sub-traces 10% shorter than the actual cycle length and searched for the next minimum point (i.e., the beginning of the next sub-trace) starting there. Note that this does not affect the results since the leakages are at the beginning of the clock cycle where the registers are updated at the rising edge. Figure 7 also shows an example hypotheses table for one sub-key of `Frodo`. Note that there are 11 possible values for each sub-key of $\mathbf{S}$ and as many sub-trace as there are modular multiplication, i.e., there are 752 possible tests for each key guess.

Power models in our analysis use the Hamming distance of the registers and we consider all registers in the datapath. We use the Pearson correlation coefficient based distinguisher for the differential side-channel attack [17]. This test aims at differentiating populations through their covariance, i.e., by checking if deviations from mean occur in a similar fashion. Correlation trace $r_{i,j}$ for a sub-key guess i is defined as follows:

$$r_{i,j} = \frac{\sum_{d=1}^{D}\left[\left(h_{d,i} - \overline{h_i}\right)\left(t_{d,j} - \overline{t_j}\right)\right]}{\sqrt{\sum_{d=1}^{D}\left(h_{d,i} - \overline{h_i}\right)^2 \sum_{d=1}^{D}\left(t_{d,j} - \overline{t_j}\right)^2}}, \tag{1}$$

where D is the number of traces each having T data points,[3] $t_{d,j}$ is a power trace with $0 < d \leq D$ and $0 < j \leq T$, $\overline{t_j}$ is the mean power trace, $h_{d,i}$ is a power estimate (Hamming distance of the target register) of trace d for the key guess value i, and $\overline{h_i}$ is the mean power estimate. The result $r_{i,j}$ returns a correlation trace with values between [−1,1] that estimates the linear relationship between the sub-key guess r_i and the power measurement for each guess i and time j. This trace depicts the significance and the timing information of the DPA leak.

5.3 Empirical Validation of Attacks on Hardware

Figure 8 presents the evaluation results of our Horizontal DPA attack on the `Frodo` key exchange protocol and validates the effects we discussed in Section 4.2. Figure 8(a) evaluate attacking the first row of coefficients. As expected, targeting partial product generation results in false positives due to modular reduction with a power of two. However, the attacker has to subsequently extract the key coefficients, starting from the first row, by targeting the intermediate memory update. The results in Figure 8(b) illustrates that attacking the second row removes false positives; there is only a single sub-key guess that crosses the threshold. Hence, from this row onwards, there is a single true positive of the correlation test. This result validates our attack's hypothesis by providing empirical evidence.

[3]In our experiments, D can range between 0 and 1344 (i.e., the ring size), and T can range between 0 and 333 (i.e., clock frequency divided by the oscilloscope sampling rate). To improve the success of our attack, we use 1344 for D. For convenience, we chose 250 for T when attacking hardware and 300 for T when attacking software (since the leakage is at the beginning of the clock cycle, we do not need to include the full clock period and any number higher than 200 would work).

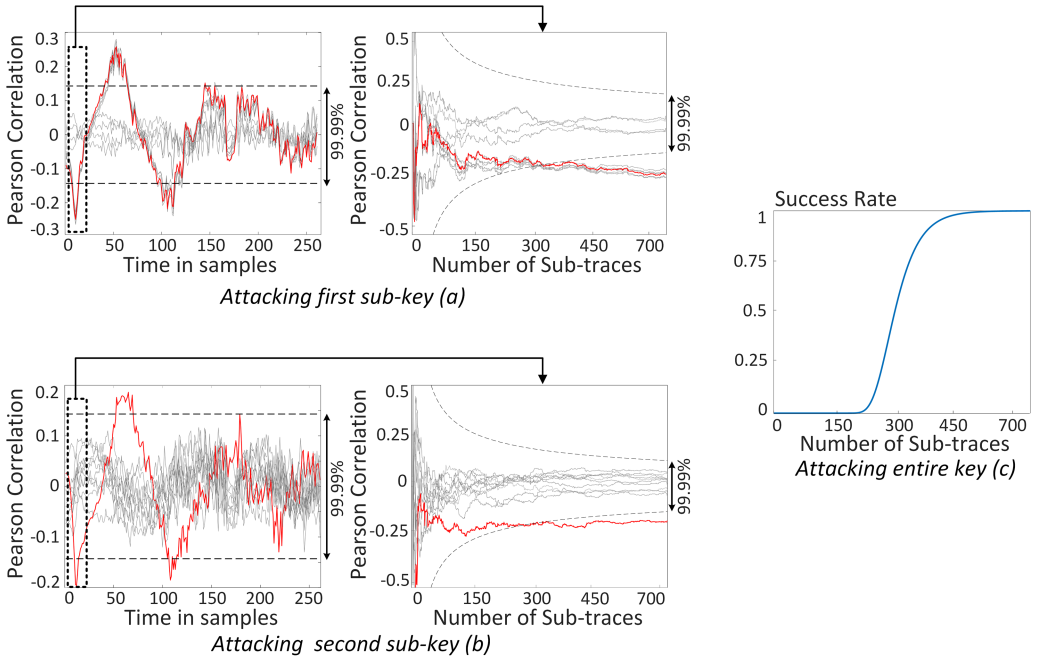

Fig. 8. Evaluation results of the proposed attack on Frodo. Correct key guesses are marked in bold red and dashed lines mark the confidence interval of 99.99%. Attacking the first row of key (a) results in both true and false positives on Frodo. Starting from the second row (b), false positives are eliminated. In all cases (c), entire keys can be successfully extracted using our attack.

The success rate, shown in Figure 8(c), reflects the probability of successfully estimating coefficients of an entire key, e.g., 752×8 elements of matrix $\mathbf{S}$ in Frodo. This value is calculated by performing a hypothesis test on the difference of correlation coefficient for the correct guess vs. incorrect key guess having the highest correlation—the details of computing this test between two correlation coefficients are given in Mangard et al. [36]. The number of possible horizontal tests is statistically sufficient to estimate the entire key.

6 ATTACKING SOFTWARE IMPLEMENTATIONS

In this section, we extend the proposed attacks to a software realization, build stronger attacks with template profiling, and validate effectiveness on the same FPGA using real power measurements. Although we are attacking a software implementation, the attack itself still abuses hardware behavior of power activity.

6.1 The Targeted Software Implementation

We have also implemented a software realization of the matrix multiplication, and applied this in the context of executing the operations of FrodoKEM key generation as described in Section 3.2. Keeping the resource-constrained applications in mind, we have opted for a low-area RISC-V based architecture and selected the PICORV32[4] for the implementation. PICORV32 has a simple, area-optimized architecture with a single-issue, in-order core. We can implement the PicoRV32 core to support RV32E, RV32I, RV32IC, RV32IM, or RV32IMC configurations, which respectively

[4]Code obtained from https://github.com/cliffordwolf/picorv32 commit version e308982e18fc952a8d446ddb7ea8b70433a998c2.

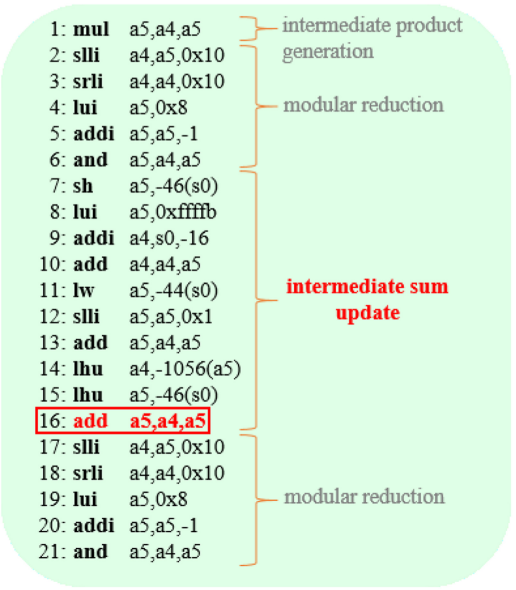

Fig. 9. The related part of the RISC-V assembly code for the FrodoKEM matrix multiplication. The highlighted line in red shows the target operation of our DPA attack.

supports the instructions sets for 32-bit base integer (embedded), base integer, standard extension for compressed instructions, and standard extension for integer multiplication and division. For our analysis, we used the RV32IM, which provides the integer multiplication instruction. We also set `ENABLE_FAST_MUL` parameter to 1. This enables `picorv_32_pcpi_fast_mul` core that provides a single cycle multiplier.

We compile the matrix multiplication software C code for the `PicoRV32` core using the RISC-V software toolchain. We first compile the source files with `riscv-gnu tool-chains`, which are referenced in the corresponding git repositories. The RISC-V gcc compiler compiles the software from C/C++ source language code to object file. Based on target architecture and instruction set support, `PicoRV32` linker links the object files and generates the executable output. The `PicoRV32` design requires dumping the resulting hex files to memory. Therefore, we use `objcopy` command and a Python script to format the output of `.tmp` file into a `.hex` file, which is then dumped to the memory of `PicoRV32`.

Figure 9 shows the related parts of the assembly dump file corresponding to the RISC-V assembly instructions for the matrix multiplication kernel. As described before, the matrix multiplication loop consists of two operations: generating partial products and updating intermediate sum. These are succeeded by modular reductions (even though modulus is a power of two, a few operations are needed to enforce bit alignment and overflow prevention). In RISC-V assembly, the entire computations during the calculation of an intermediate sum (excluding initial load and storing the resulting value) corresponds to a sequence of 21 instructions starting with the modular multiplication. The figure highlights the targeted operation of our attack in RISC-V assembly with the red color. The 16th operation is the targeted computation of our attack: the update of the intermediate sum by adding currently stored value and the generated partial product. This performs the same add operation in the dedicated hardware design. Note that there are other potentially leaky instructions and we conduct a more comprehensive analysis in Section 6.3 via template attacks.

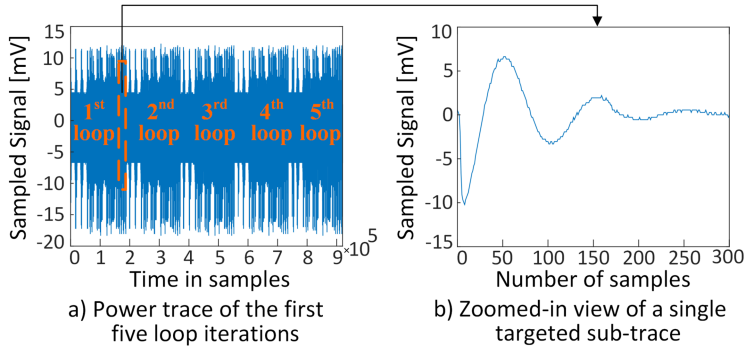

Fig. 10. The power trace of the first five loop iterations (a) and the Zoomed-in view of a single targeted sub-trace (b).

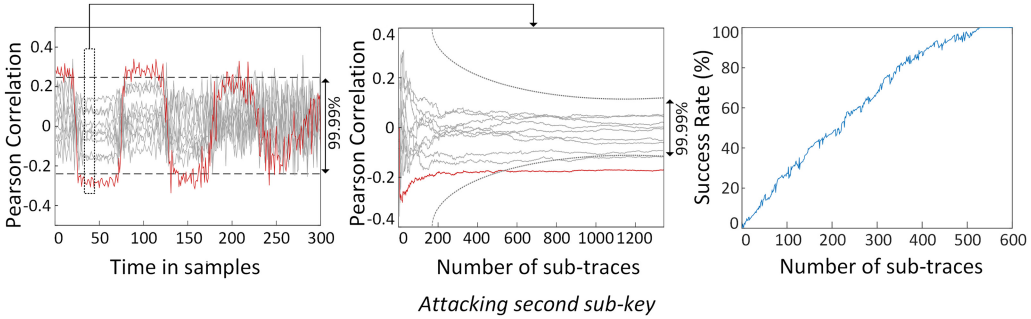

Fig. 11. Evaluation results of the proposed attack on FrodoKEM. Correct key guesses are marked in bold red and dashed lines mark the confidence interval of 99.99%. The sub-key can be successfully extracted using our attack.

6.2 Empirical Validation of Attacks on Software

We use the same FPGA as in Section 5.2 but we obtain the power measurements with a different oscilloscope—the Picoscope 3206D—because the software implementation takes much longer to execute creating longer power measurements that cannot be stored with the on-device memory of SDS1102X Digital Oscilloscope. The sampling frequency of the Picoscope 3206D oscilloscope is set to 500 MHz. A higher sampling frequency leads to memory storage challenges.

Figure 10(a) shows the power measurements corresponding to the first five loop iterations. The figure also marks where within this execution the targeted "add" instructions occur. The sub-traces of 1 cycle in length associated with these operations are isolated with post-processing (see Figure 10(b)) and the horizontal DPA table is organized as in Section 5.2.

Figure 11 quantifies our attack's results on the software implementation. This attack aims at the second row of the secret matrix **S** and thus removes the false positives just like the attack on the dedicated hardware. The results here show that the available 1,344 tests are indeed statistically sufficient to extract the secret coefficient as around 530 sub-traces the correlation coefficient crosses the 99.99% significance threshold. However, the attack here is less successful (by about 76%) compared with the attack on hardware. Given that the core multiplication operations are similar in the two protocols, we believe that the primary reason for this reduction could be the added control overhead and non-secret-data-dependent units (e.g., instruction fetch and decode) of the

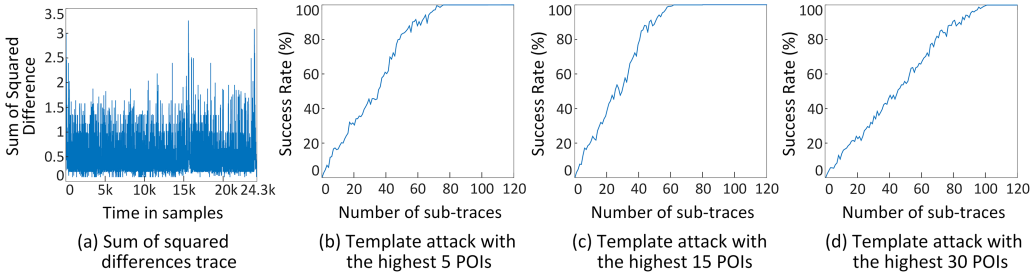

Fig. 12. Sum of squared differences trace (a) that corresponds to the code lines 7–21 in Figure 9(b) . TA result (b) using the highest 5 POIs. TA result (c) using the highest 15 POIs. TA result (d) using the highest 30 POIs.

processor increasing microarchitectural noise. The effect of control overhead and non-secret-data-dependent unit may change across sub-traces. But if there are certain distinctive features of these activities, e.g., in the frequency or a higher dimensional domain, some more advanced attacks such as frequency-based [37] or ML-based [63] can be implemented.

6.3 Improving Attack Effectiveness with Template Profiling

We next develop template based attacks to further improve the accuracy. This attack, however, needs configuring the device with possible keys and profiling their power measurement behavior. Once these templates are derived, the side-channel attack then calculates the probability of a given power measurement belonging to each template and chooses the one that maximizes the probability.

Our template attack is expected to outperform the DPA because it uses more than a single time stamp in the power measurement profile whereas DPA *independently* tests each sample in the time domain. This is especially important for software implementation as there are multiple instructions that correspond to the processing of the secret-dependent data and thus multiple points in time that are potentially leaky.

Unfortunately, a template attack using all time samples is typically inefficient [22]. A successful attack has to select a subset of these samples and the selection should not be random. We therefore conduct a SOSD based technique to rank the points that are important for the side-channel analysis [22]. These points are called the **Points of Interests** (**POIs**). This technique applies the sum of squared operations to means of all power traces. For every sample i in a number of power traces $t_1..t_k$ that are collected using different secret keys k, we find mean power M in Equation (2). Then, we calculate SOSD in Equation (3) for each sample points.

$$M_{k,i} = \frac{1}{T_k} \sum_{j=1}^{T_k} t_{j,i}, \tag{2}$$

$$SOSD_i = \sum_{k_1,k_2} (M_{k_1,i} - M_{k_2,i})^2. \tag{3}$$

We select trace samples that correspond to indexes of the highest SOSD values. The high SOSD value means that the variation of power consumption is high at these samples potentially due to the higher leakage. We use these selected trace samples at both the template profiling and attack stages. For the profiling step, we calculate variance of power v_{s_i} in Equation (4), and covariance in Equation (5) c_{s_i,s_i^*} at every pair of the selected trace (s_i and s_i^*) for creating the covariance matrix S_i with the Equation (6).

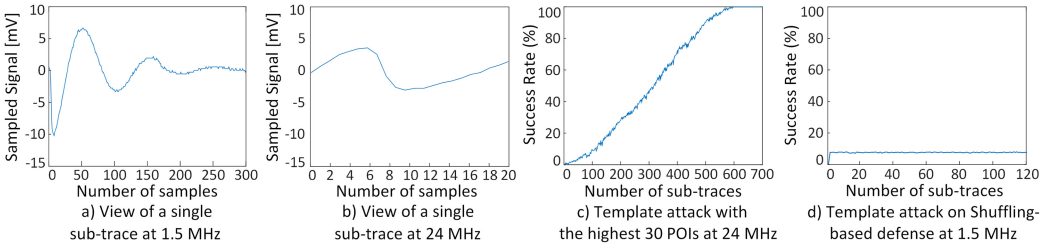

Fig. 13. The view of single sub-trace at 1.5 MHz clock frequency (a) and at 24 MHz clock frequency (b). TA result (c) using the highest 30 POIs at 24 MHz clock frequency. TA result on Shuffling-based defense (d) using the highest 30 POIs.

$$v_{s_i} = \frac{1}{T_k} \sum_{j=1}^{T_k} (t_{j,s_i} - M_{s_i})^2, \tag{4}$$

$$c_{s_i,s_i^*} = \frac{1}{T_k} \sum_{j=1}^{T_k} (t_{j,s_i} - M_{s_i})(t_{j,s_i^*} - M_{s_i^*}), \tag{5}$$

$$S_{s_i} = \begin{pmatrix} v_1 & c_{1,2} & c_{1,3} & \dots \\ c_{2,1} & v_2 & c_{2,3} & \dots \\ c_{3,1} & c_{3,2} & v_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{6}$$

During the attack stage, we apply the **multivariate normal probability density function** (**MVNPDF**) using the selected samples of attacked traces r_{s_i} and calculated mean M_{s_i} and covariance matrix S_{s_i} values at the profiling stage. We finally sum the log of the normal distribution $\mathcal{N}$ as in Equation (7) to avoid numerical errors that occur if the results of MVNPDF are too large. The index of the matrix P_k with the highest value corresponds to the predicted sub-key guess.

$$P_k = \sum_{j=0}^{k} \log \mathcal{N}(r_{j,s_i}, M_{s_i}, S_{s_i}). \tag{7}$$

Figure 12 shows the SOSD trace (a) along with the results of our template attack with 5 (b), 15 (c), and 30 (d) POIs. As expected, the template attacks do outperform the horizontal DPA attacks. While the DPA achieves a 100% success rate after 530 sub-traces, the template attack only needs 60–100 sub-traces based on the number of POIs, corresponding to an 81.1%–86.6% improvement. The results also quantify that selecting 15 POIs seems to be a better option than 5 or 30 POIs. The most likely explanation of this is that adding more POIs would eventually cause adding points that show a high variation but those variations do *not* relate to secret data processing.

6.4 Template Attack at Higher Operating Frequency

We evaluate the template attacks at a higher operating frequency of 24 MHz since the frequency of many embedded RFID/MCU processors is about or below 24 MHz [13, 38]. Figure 13 shows the view of a single sub-trace at 1.5 MHz clock frequency (a) and at 24 MHz clock frequency (b). The length of the sub-trace in Figure 13(b) is less than (a) because the oscilloscope stores less samples when the operating frequency of the FPGA increases. Figure 13(c) shows template attacks with 30 POIs at 24 MHz can still successfully break the system; however, it requires much more number of sub-traces than at the low frequency implementation as expected.

7 DISCUSSIONS

7.1 Attacking Other Lattice-Based Cryptosystems

Although the scope of this article is the matrix multiplication in lattice-based cryptosystems, our Horizontal DPA has the potential to also break other lattice-based constructions such as the public-key decryption. Even though the baseline design for these applications does not necessarily require single-trace attacks as they work with long-term secret keys, masking, blinding, and re-keying based DPA countermeasures can fail against single-trace attacks [51]. Therefore, our attack can be used in the presence of such countermeasures. However, a potential problem may occur when implementing our attack on schemes that work with smaller degree polynomials. These polynomials would only allow a smaller number of horizontal tests and may thus require a better oscilloscope or EM probing to reduce noise.

Our attack can also be applied on R-LWE schemes that use polynomial multiplication instead of matrix multiplication. But such an attack would be possible on the regular (i.e., schoolbook) polynomial multiplication. An alternative method to implement *polynomial* multiplication is using NTT, which is essentially an arithmetic transformation possible to trade-off area for performance in high-end application scenarios. Indeed, prior works favor schoolbook polynomial multiplication over NTT for area-constrained platforms [50] while some works comment that it may even have better performance than NTT in some corner cases [49] or yield to a higher operating frequency due to its simplified control [18]. We note that our proposed attack is not directly applicable on NTT-based multiplications but other single-trace attacks on NTT-based multiplications have also been shown [45, 48]. Extensions of our proposed attack on various forms of NTT implementations [28, 39, 40, 44] are out of the scope of this work.

There are three remaining lattice-based post-quantum key-encapsulation finalists at NIST's standardization process. KYBER, whose security is based on the hardness of solving the LWE problem over module lattices [7] uses NTT to efficiently multiply polynomials. Therefore, our attack is not applicable to KYBER in a straightforward manner. NTRU is another lattice-based finalist. Although NTRU originally is not anticipated for NTT multiplication, recent works have shown how to retrofit NTT into NTRU [1, 35]. SABER, whose security relies on the hardness of the Module Learning With Rounding problem [11], is another remaining finalist. SABER uses schoolbook multiplications and thus resembles similarities to `FrodoKEM`. SABER is, therefore, potentially vulnerable to our attack but will likely yield worse results given that SABER uses a smaller polynomial ring. Extensions of our attack to NIST candidates or more broadly to other lattice-cryptosystems are out of the scope of this work.

7.2 Possible Countermeasures

A common method to mitigate DPA attacks is to introduce randomness into the computations. In our case, this can be achieved by randomizing the order of computations since the result of matrix and polynomial multiplication is independent of the order in which partial products are generated. Another option might be to add dummy steps in the computation. These countermeasures would encumber the adversaries' capability to distinguish sub-traces within a power trace and to associate them with corresponding sub-key guesses.

The defense proposed by Bos et al. is another alternative [16] to protect the system against single-trace attacks. This method shuffles the order of operations (i.e., intermediate value updates) with a fixed but random sequence. Therefore, it can work if the threat model relaxes the assumption on the adversary by expecting that the hardware implementation details will be kept secret. We applied an emulated version of shuffling-based defense to the `FrodoKEM` software-based implementation in order to observe the effectiveness of our template attack. Figure 13(d) shows the results of template

attack with 30 POIs at 1.5 MHz. The success rate of the template attack drops to about 7.69% (random guess rate for −6 to 6 sub-key range).

8 CONCLUSION

Key exchange protocols and key encapsulation mechanisms are important cryptographic routines for large-scale communication protocols. Just because these protocols may work with one-time secrets or include masking defenses against multi-trace attacks, their single-trace side-channel analysis cannot be overlooked. In this article, we validate that it is indeed a mistake to assume such limitations would, by default, prevent single-trace side-channel attacks. As new key exchange and encapsulation protocols are being formulated and deployed, their single-trace side-channel evaluation (and not just SPA leaks) should play a role in the decision of their implementation choices. This article shows that matrix multiplication is vulnerable to a novel side-channel attack. Therefore, some form of countermeasure is required in their implementation.

ACKNOWLEDGMENTS

We thank Ashay Rane for helpful discussions and Youssef Tobah for his contributions to the conference version of this work. We thank anonymous reviewers for their feedback.

REFERENCES

[1] Erdem Alkim, Dean Yun-Li Cheng, Chi-Ming Marvin Chung, Hülya Evkan, Leo Wei-Lun Huang, Vincent Hwang, Ching-Lin Trista Li, Ruben Niederhagen, Cheng-Jhih Shih, Julian Wälde, and Bo-Yin Yang. 2020. Polynomial multiplication in NTRU prime: Comparison of optimization strategies on cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021, 1 (2020), 243–268. DOI: https://doi.org/10.46586/tches.v2021.i1.217-238

[2] Erdem Alkim, Joppe W. Bos Léo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, and Douglas Stebila. 2020. FrodoKEM Learning With Errors Key Encapsulation Algorithm Specifications And Supporting Documentation. https://frodokem.org/files/FrodoKEM-specification-20200930.pdf.

[3] Dorian Amiet, Andreas Curiger, Lukas Leuenberger, and Paul Zbinden. 2020. Defeating newhope with a single trace. In *Proceedings of the International Conference on Post-Quantum Cryptography*. 189–205.

[4] Soojung An, Suhri Kim, Sunghyun Jin, HanBit Kim, and HeeSeok Kim. 2018. Single trace side channel analysis on NTRU implementation. *Applied Science* 8, 11 (2018), 1–17.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A. Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[6] Ali Can Atici, Lejla Batina, Benedikt Gierlichs, and Ingrid Verbauwhede. 2008. Power analysis on NTRU implementations for RFIDs: First results. In *Proceedings of the Workshop on RFID Security*. 128–139.

[7] Roberto Avanzi, Léo Ducas Joppe Bos, Eike Kiltz, Tancréde Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 2021. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation. Retrieved June 9, 2021 from https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf.

[8] Furkan Aydin, Priyank Kashyap, Seetal Potluri, Paul Franzon, and Aydin Aysu. 2020. DeePar-SCA: Breaking parallel architectures of lattice cryptography via learning based side-channel attacks. In *Proceedings of the International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*. Springer, 262–280.

[9] Aydin Aysu, Youssef Tobah, Mohit Tiwari Andreas Gerstlauer, and Michael Orshansky. 2018. Horizontal side-channel vulnerabilities of post-quantum key exchange protocols. In *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*. 81–88. DOI: https://doi.org/10.1109/HST.2018.8383894

[10] Josep Balasch, Benedikt Gierlichs, Oscar Reparaz, and Ingrid Verbauwhede. 2015. *DPA, Bitslicing and Masking at 1 GHz*. Springer Berlin Heidelberg, Berlin, 599–619. DOI: https://doi.org/10.1007/978-3-662-48324-4_30

[11] Andrea Basso, Jose Maria Bermudo Mera, Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Michiel Van Beirendonck, and Frederik Vercauteren. 2020. *SABER: Mod-LWR based KEM*. Technical report. Retrieved from https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround3.pdf.

[12] Aurélie Bauer, Eliane Jaulmes, Emmanuel Prouff, and Justine Wild. 2014. *Horizontal Collision Correlation Attack on Elliptic Curves*. Springer Berlin Heidelberg, Berlin, 553–570. DOI: https://doi.org/10.1007/978-3-662-43414-7_28

[13] STMicroelectronics 8 bit MCUs. 2020. Retrieved June 9, 2021 from https://www.st.com/en/microcontrollers-microprocessors/stm8-8-bit-mcus.html.

[14] Jonathan Bootle, Claire Delaplace Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. 2018. LWE without modular reduction and improved side-channel attacks against BLISS. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 494–524.

[15] Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. 2016. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1006–1018.

[16] Joppe W. Bos, Simon Friedberger, Marco Martinoli, Elisabeth Oswald, and Martijn Stam. 2018. Assessing the feasibility of single trace power analysis of frodo. In *Proceedings of the Selected Areas in Cryptography*. Springer, 216–234.

[17] Eric Brier, Christophe Clavier, and Francis Olivier. 2004. Correlation power analysis with a leakage model. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 16–29.

[18] Johannes Buchmann, Florian Göpfert, Tim Güneysu, Tobias Oder, and Thomas Pöppelmann. 2016. High-performance and lightweight lattice-based public-key encryption. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*. ACM, New York, NY, 2–9. DOI: https://doi.org/10.1145/2899007.2899011

[19] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. 2002. Template attacks. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 13–28.

[20] Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. 2010. Horizontal correlation analysis on exponentiation. In *Proceedings of the International Conference on Information and Communications Security*, Vol. 6476. Springer, 46–61.

[21] Thomas Espitau, Pierre-Alain Fouque, Benoit Gerard, and Mehdi Tibouchi. 2017. *Side-Channel Attacks on BLISS Lattice-Based Signatures – Exploiting Branch Tracing Against strongSwan and Electromagnetic Emanations in Microcontrollers*. Cryptology ePrint Archive, Report 2017/505. Retrieved June 9, 2021 from http://eprint.iacr.org/2017/505.

[22] Guangjun Fan, Yongbin Zhou, Hailong Zhang, and Dengguo Feng. 2014. How to choose interesting points for template attacks more effectively? In *Proceedings of the International Conference on Trusted Systems*, Vol. 9473. 168–183.

[23] Federal Office for Information Security. 2020. BSI TR-02102-1: "Cryptographic Mechanisms: Recommendations and Key Lengths" Version: 2020-1. Retrieved from https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html.

[24] Pierre-Alain Fouque and Frédéric Valette. 2003. The doubling attack-why upwards is better than downwards. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, Vol. 2779. Springer, 269–280.

[25] Wei-Lun Huang, Jiun-Peng Chen, and Bo-Yin Yang. 2020. Power analysis on NTRU prime. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 1 (2020), 123–151. DOI: https://doi.org/10.13154/tches.v2020.i1.123-151

[26] Mehmet Sinan İnci, Berk Gulmezoglu, Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar. 2016. *Cache Attacks Enable Bulk Key Recovery on the Cloud*. Springer Berlin Heidelberg, Berlin, 368–388. DOI: https://doi.org/10.1007/978-3-662-53140-2_18

[27] Matthias J. Kannwischer, Peter Pessl, and Robert Primas. 2020. Single-trace attacks on keccak. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 3 (2020), 243–268. DOI: https://doi.org/10.13154/tches.v2020.i3.243-268

[28] Emre Karabulut and Aydin Aysu. 2020. RANTT: A RISC-V architecture extension for the number theoretic transform. In *Proceedings of the 30th International Conference on Field-Programmable Logic and Applications*. 26–32. DOI: https://doi.org/10.1109/FPL50879.2020.00016

[29] Emre Karabulut and Aydin Aysu. 2021. *Falcon Down: Breaking Falcon Post-Quantum Signature Scheme through Side-Channel Attacks*. Cryptology ePrint Archive, Report 2021/772. Retrieved June 9, 2021 from https://eprint.iacr.org/2021/772.

[30] Priyank Kashyap, Furkan Aydin, Seetal Potluri, Paul Franzon, and Aydin Aysu. 2020. 2Deep: Enhancing side-channel attacks on lattice-based key-exchange via 2D deep learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40, 6 (2020), 1217–1229. DOI: https://doi.org/10.1109/TCAD.2020.3038701

[31] Il-Ju Kim, Tae-Ho Lee, Jaeseung Han, Bo-Yeon Sim, and Dong-Guk Han. 2019. *On Security of Fiat-Shamir Signatures over Lattice in the Presence of Randomness Leakage*. Cryptology ePrint Archive, Report 2019/715. Retrieved June 9, 2021 from http://eprint.iacr.org/2019/715.

[32] Il-Ju Kim, Tae-Ho Lee, Jaeseung Han, Bo-Yeon Sim, and Dong-Guk Han. 2020. *Novel Single-Trace ML Profiling Attacks on NIST 3 Round candidate Dilithium*. Cryptology ePrint Archive, Report 2020/1383. Retrieved June 9, 2021 from http://eprint.iacr.org/2020/1383.

[33] Suhri Kim and Seokhie Hong. 2018. Single trace analysis on constant time cdt sampler and its countermeasure. *Applied Sciences* 8, 10 (2018), 1809.

[34] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Proceedings of the Annual International Conference on Advances in Cryptology*. Springer, 789–789.

[35] Vadim Lyubashevsky and Gregor Seiler. 2019. NTTRU: Truly fast NTRU using NTT. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019, 3 (2019), 180–201. DOI: https://doi.org/10.13154/tches.v2019.i3.180-201

[36] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. 2007. *Statistical Characteristics of Power Traces*. Springer US, Boston, MA, 61–99. DOI: https://doi.org/10.1007/978-0-387-38162-6_4

[37] Edgar Mateos and Catherine H. Gebotys. 2010. A new correlation frequency analysis of the side channel. In *Proceedings of the 5th Workshop on Embedded Systems Security*. ACM, 1–8.

[38] MaximIntegrated Secure MCUs. 2020. Retrieved from https://para.maximintegrated.com/en/search.mvp?fam=micros&1233=Secure.

[39] Ahmet Can Mert, Emre Karabulut, Erdinc Ozturk, Erkay Savas, and Aydin Aysu. 2020. An extensive study of flexible design methods for the number theoretic transform. *IEEE Transactions on Computer*. 1–1. https://doi.org/10.1109/TC.2020.3017930

[40] Ahmet Can Mert, Emre Karabulut, Erdinc Ozturk, Erkay Savas, Michela Becchi, and Aydin Aysu. 2020. A flexible and scalable NTT hardware: Applications from homomorphically encrypted deep learning to post-quantum cryptography. In *Proceedings of the 2020 Design, Automation Test in Europe Conference Exhibition*. 346–351. DOI: https://doi.org/10.23919/DATE48585.2020.9116470

[41] National Institute of Standards and Technology. 2015. Workshop on Cybersecurity in a Post-Quantum World. Retrieved from https://www.nist.gov/news-events/events/2015/04/workshop-cybersecurity-post-quantum-world.

[42] Kalle Ngo, E. Dubrova, Q. Guo, and T. Johansson. 2021. A side-channel attack on a masked IND-CCA secure saber KEM. *IACR Cryptology ePrint Archive* 2021, 4 (2021), 676–707. DOI: https://doi.org/10.46586/tches.v2021.i4.676-707

[43] Tobias Oder, Tobias Schneider, Thomas Pöppelmann, and Tim Güneysu. 2018. Practical CCA2-secure and masked ring-LWE implementation. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018, 1 (2018), 142–174. DOI: https://doi.org/10.13154/tches.v2018.i1.142-174

[44] Erdem Ozcan and Aydin Aysu. 2020. High-level synthesis of number-theoretic transform: A case study for future cryptosystems. *IEEE Embedded Systems Letters* 12, 4 (2020), 133–136. DOI: https://doi.org/10.1109/LES.2019.2960457

[45] Apostolos P. Fournaris, Charis Dimopoulos, and Odysseas Koufopavlou. 2020. Profiling dilithium digital signature traces for correlation differential side channel attacks. In *Proceedings of the International Conference on Embedded Computer Systems:Architectures, Modeling, and Simulation*. Springer, 281–294.

[46] Aesun Park and Dong-Guk Han. 2016. Chosen ciphertext simple power analysis on software 8-bit implementation of Ring-LWE encryption. In *Proceedings of the IEEE Asian Hardware-Oriented Security and Trust*. 1–6. DOI: https://doi.org/10.1109/AsianHOST.2016.7835555

[47] Peter Pessl. 2016. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In *Proceedings of the 17th International Conference on Progress in Cryptology*. Springer, 153–170.

[48] Peter Pessl and Robert Primas. 2019. More practical single-trace attacks on the number theoretic transform. In *Proceedings of the International Conference on Cryptology and Information Security in Latin America Progress in Cryptology*. Springer, 130–149.

[49] Thomas Pöppelmann and Tim Güneysu. 2012. Towards efficient arithmetic for lattice-based cryptography on reconfigurable hardware. In *Proceedings of the 2nd International Conference on Cryptology and Information Security in Latin America* . Springer-Verlag, Berlin, 139–158. DOI: https://doi.org/10.1007/978-3-642-33481-8_8

[50] Thomas Pöppelmann and Tim Güneysu. 2014. Area optimization of lightweight lattice-based encryption on reconfigurable hardware. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. 2796-2799. DOI: https://doi.org/10.1109/ISCAS.2014.6865754

[51] Robert Primas, Peter Pessl, and Stefan Mangard. 2017. Single-trace side-channel attacks on masked lattice-based encryption. In *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 513–533.

[52] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay. 2020. *Drop by Drop you break the rock - Exploiting generic vulnerabilities in Lattice-based PKE/KEMs using EM-based Physical Attacks*. Cryptology ePrint Archive, Report 2020/549. Retrieved June 9, 2021 from http://eprint.iacr.org/2020/549.

[53] Prasanna Ravi, Mahabir Prasad Jhanwar, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. 2018. *Side-channel Assisted Existential Forgery Attack on Dilithium-A NIST PQC candidate*. Cryptology ePrint Archive Report 2018/821. Retrieved June 9, 2021 from https://eprint.iacr.org/2018/821.pdf.

[54] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. 2020. Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 3 (2020), 307–335. DOI: https://doi.org/10.46586/tches.v2020.i3.307-335

[55] Oded Regev. 2009. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM* 56, 6 (2009), 1–40.

[56] Oscar Reparaz, Ruan de Clercq, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. 2016. Additively homomorphic ring-LWE masking. In *Proceedings of the International Workshop on Post-Quantum Cryptography*. Springer, 233–244.

[57] Oscar Reparaz, Sujoy Sinha Roy, Ruan de Clercq, Frederik Vercauteren, and Ingrid Verbauwhede. 2016. Masking ring-LWE. *Journal of Cryptographic Engineering* 6, 2 (2016), 139–153.

[58] Oscar Reparaz, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. 2015. A masked ring-LWE implementation. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 683–702.

[59] Peter W. Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. 124–134. https://doi.org/10.1109/SFCS.1994.365700

[60] Bo-Yeon Sim, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Taeho Lee, Jaeseung Han, Hyojin Yoon, Jihoon Cho, and Dong-Guk Han. 2020. *Single-Trace Attacks on the Message Encoding of Lattice-Based KEMs*. Cryptology ePrint Archive, Report 2020/992. Retrieved June 9, 2021 from https://eprint.iacr.org/2020/992.

[61] An Wang, Xuexin Zheng, and Zongyue Wang. 2013. Power analysis attacks and countermeasures on NTRU-based wireless body area networks. *KSII Transactions on Internet and Information Systems* 7, 5 (2013), 1094–1107.

[62] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, and David Oswald. 2020. *Magnifying Side-Channel Leakage of Lattice-Based Cryptosystems with Chosen Ciphertexts: The Case Study of Kyber*. Cryptology ePrint Archive Report 2020/912. Retrieved June 9, 2021 from https://eprint.iacr.org/2020/912.

[63] Guang Yang, Huizhong Li, Jingdian Ming, and Yongbin Zhou. 2018. Convolutional neural network based sidechannel attacks in time-frequency representations. In *Proceedings of the International Conference on Smart Card Research and Advanced Applications*. Springer, 1–17.

[64] Cong Zhang, Zilong Liu, Yuyang Chen, Jiahao Lu, and Dongsheng Liu. 2020. A flexible and generic Gaussian sampler with power side-channel countermeasures for quantum-secure internet of things. *IEEE Internet of Things Journal* 7, 9 (2020), 8167–8177.

[65] Yinqian Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2012. Cross-VM side channels and their use to extract private keys. In *Proceedings of the ACM Conference on Computer and Communications Security*. ACM, New York, NY, 305–316. https://doi.org/10.1145/2382196.2382230

[66] Xuexin Zheng, An Wang, and Wei Wei. 2013. First-order collision attack on protected NTRU cryptosystem. *Microprocessors and Microsystems* 37, 6 (2013), 601–609.

Received February 2021; revised July 2021; accepted July 2021