Post-Quantum Cryptography: Challenges and Opportunities for Robust and Secure HW Design

Davide Bellizia, Nadia El Mrabet, Apostolos Fournaris, Simon Pontié, Francesco Regazzoni, François-Xavier Standaert, Élise Tasso, Emanuele Valea

Post-Quantum Cryptography: Challenges and Opportunities for Robust and Secure HW Design

Davide Bellizia[*], Nadia El Mrabet[†], Apostolos P. Fournaris[‡], Simon Pontié[%,§],
Francesco Regazzoni[¶,$], François-Xavier Standaert[*], Élise Tasso[%,§], Emanuele Valea[#]
[*]*Université Catholique de Louvain*, Leuven, Belgium. [†]*Mines Saint-Étienne, CEA-Tech, Centre CMP*, F-13541 Gardanne, France
[‡]Industrial Systems Institute, Research Center ATHENA, 26504 Marousi, Greece
[%]*CEA-Leti, Univ. Grenoble Alpes*, F-38000 Grenoble, France; [#]*Univ. Grenoble Alpes, CEA, List*, F-38000 Grenoble, France
[§]*CEA Tech, Centre CMP, Équipe Commune CEA Tech - Mines Saint-Etienne*, F-13541 Gardanne, France
[¶]*University of Amsterdam*, Amsterdam, The Netherlands; [$]*Università della Svizzera Italiana*, Lugano, Switzerland

***Abstract*—Post-Quantum Cryptography (PQC) will become soon the standard for many systems of the future. With the advent of quantum computers, all encrypted communications based on traditional asymmetric cryptography (e.g., RSA, ECC) will become insecure. The definition the PQC standards is an on going process proceeding at a fast pace, involving new and largely unexplored cryptographic primitives. For this reason, the design of hardware implementations of PQC algorithms is still under study. In this paper, we introduce the fundamentals of PQC, with a focus on lattice-based cryptography and its hardware security issues, namely side-channel and fault-based attacks. Then, we focus on isogeny-based cryptography and the SIKE algorithm. We highlight the importance of fault-tolerant design choices through the presentation of a fault attack, based on the electromagnetic injection of transient faults, targeting this cryptographic primitive. Finally, we show an interesting idea that starts from the observation that some PQC algorithms have an intrinsic probabilistic behavior. We argue that this characteristic is a clear opportunity that paves the way for the application of approximate (or inexact) computing to the implementation of PQC cryptography.**

I. Introduction

The advent of quantum computing represents a menace for the security of modern communication systems. In fact, most communication protocols that are used over the internet rely on asymmetric cryptography for exchanging secret keys. Asymmetric cryptography standards place their security on the hardness of specific mathematical problems, such as the factorization of long integers (i.e., RSA). Shor's algorithm has been proven to be able to solve these problems in polynomial time on a quantum computer that is powerful enough [1]. For this reason, the scientific community found a new interest in studying asymmetric cryptography based on mathematical problems that preserve their hardness even against a quantum computer, namely Post-Quantum Cryptography (PQC). In 2017, the National Institute of Standards and Technology (NIST) started a competition with the aim of defining the new PQC standards. These standard proposals are based on different families of cryptographic primitives, such as lattice-based cryptography and isogeny-based cryptography.

This work is partially supported by the European Union Horizon 2020 research and innovation program under CPSoSAware project (grant No. 871738)

The theoretical security of PQC primitives is being extensively scrutinized, giving a good confidence in their robustness from the mathematical point of view. However, this is not the case for physical security evaluation. In fact, research on the hardware implementation of PQC algorithms is still at the beginning, leaving many problems still open. For instance, the resistance of PQC implementations against side-channel and fault attacks is still partially unexplored [2]. In addition, the PQC seems to be a promising application that could rely on emerging computing paradigms (e.g., approximate computing [3], in-memory computing [4]) in order to achieve highly efficient hardware implementations.

In this paper, we provide an overview of different challenges and innovative solutions related to the hardware implementation of PQC algorithms. In Section II, we focus on lattice-based cryptography and we summarize the main physical attacks affecting this family of PQC primitives and we discuss possible countermeasures. In Section III, we focus on SIKE and we present a possible fault attack with a related countermeasure. In Section IV, we show how resorting to *inexact computing* can be a valid option in order to achieve highly efficient hardware implementations of PQC primitives.

II. Lattice-based cryptography in hardware

Lattice based schemes, a class of quantum resistant algorithms, are quite promising because of their performance and their flexibility. For practical deployment of post quantum primitives, it is necessary to ensure that these primitives reach an adequate level of performance and that their implementation is robust against physical attacks. In this section, we summarize research efforts on these two aspects for lattice based primitives.

Performance wise, the bottlenecks of lattice based cryptography is the polynomial multiplication and the noise sampler function. Multiplication can be optimally performed using the Number Theoretic Transform (NTT): the polynomials to be multiplied are converted into the spectral domain, reducing the polynomial multiplication to a simple point-wise multiplication of the two polynomials. This approach is followed by several finalists of the NIST contest [5], [6].

Aiming to render the NTT operation efficient for real-world usage, there have been works in regards to hardware implementations that incorporate several optimization techniques, targeting either FPGA platforms [7] [8] [9] [10] [11] or even low-power ASIC designs, such as [12], that exhibits a low-cost power dissipation of 30%. The hardware implementation of NTT in general follows the butterfly structure described in the Cooley-Tukey (CT), or Gentleman-Sande (GS) NTT/inverse NTT algorithm. Some hardware implementations have dedicated butterfly units [13] and/or they use systolic arrays consisting of several small NTT based Processing Elements (PEs) [7]. In general, the Hardware structure follows a combination of RAM (or BRAM in FGPA implementation) elements that are used for storage of the coefficients of the NTT input polynomials, followed by some parallel processing logic (e.g., using PEs) that handles the butterfly structure of the algorithm. However, using a fully parallel butterfly structure implementation lead to excessive chip covered area within the FPGA and it is not very practical. To reduce the LUT number on the FPGA fabric, a sequential multiplier structure is used [10] [7]. As can be seen in the work of [10], the multiplier uses a dedicated ROM to store all the twiddle factors (precomputed) which are required during the NTT computation before performing the actual NTT multiplication. To remove this ROM access overhead, in other approaches, the precomputations are replaced by repeated multiplications that are used in order to compute the twiddle factors at run-time [11]. The process can be further optimized by re-arranging the nested loops in the NTT computation as shown in [14].

Noise sampling can be done using uniform or binomial distributions (that can be implemented easily) but usually discrete Gaussian distributions are used. Sampling in such distributions with high precision is challenging and non-trivial. High-precision floating-point arithmetic operations are required to perform a high-precision Gaussian sampling with negligible statistical distance. In practice, Box-Muller and/or Ziggurat sampling [15] and sampling rejection algorithms are used including precomputed values stored in BRAMS (or ROMs) as lookup tables, followed by a few floating point multipliers and multiplexers [16] [17].

Despite that they are quite novel schema, physical attack security of lattice based constructions has been already explored, since the ease of protecting against side channel attacks is indicated by NIST as one of the criteria for the selection of the standard. The most straightforward attack goal is to use a side channel attack in order to retrieve the PQC secret key either in a Key Encapsulation Mechanism scheme (e.g., New Hope, Kyber, Saber, McEliece, NTRUPrime, NTRUEncrypt, etc.) or in a PQC digital signature scheme (e.g., Dilithium, Falcon, LAC, etc.). The dominant mathematical problem behind most of the above schemes, is the Learning With Error (LWE) problem appearing in several variations (e.g, Ring LWE, Module LWE, etc).

Timing attacks against NTRUEncrypt implementation ([18], [19]) were probably the first side channel attacks applied to lattice based constructions. The attack exploits the fact that the execution time of the hash function in the decryption process is depending on the ciphertext. By carefully selecting the ciphertexts and by analyzing the time needed to decrypt them, an adversary could be able to recover the secret key. As other timing side channel attacks, also this one could be counteracted by ensuring a constant time of operation.

Timing side channel have also been used to attack efficient implementations of discrete Gaussian samplers based on lookup tables. Despite the searching algorithms have a constant number of steps, a non constant execution time could come from the role played by caches. This is exploited, for instance, in the Flush+Reload cache-attacks [20] or on the Ring-TESLA algorithm [21]. Power analysis attacks have been used against lattice based algorithms in many forms: simple power analysis was demonstrated on an 8-bit microcontroller [22]; differential power analysis was used to attack a RFID implementation of NTRU [23], higher order attacks have been used to attack the convolution step of NTRU [24], horizontal attacks were used to attack low area designs of the NTT [25], and template attacks have been exploited to attack the Gaussian sampler of lattice signatures [26] or the NTT of the RLWE decryption [27]. Finally, the resistance against fault attacks of lattice based constructions have been analyzed too. Bindel et al. investigated the robustness of signatures schema such as BLISS, ring-TESLA and GLP signatures [28]). Valencia et al. [29] systematically evaluated the robustness of RLWE against different type of faults, including randomization faults, skipping faults and zeroing faults. Resistance against fault sensitivity analysis of arithmetic operators used in lattice based cryptography has also been explored [30]. Attacks against lattice based signature schema have also been practically demonstrated using ARM Cortex-M4 as target [31]. Similar principles have been exploited to recover the key of the FALCON algorithm [32].

In [33] the authors provide a theoretical modelling of the side-channel information that can leak from LWE based crypto algorithms. According to [33], side-channel information can be described in the form of hints that are provided to the attacker. Assuming that $\mathbf{v}$, l, k and σ are known to the attacker and that s is the secret in a given Lattice L, then four types of hints can be identified:

- Perfect hints: $< s, \mathbf{v} >= l$ (intersecting the lattice with a known hyperplane)
- Modular hints: $< s, \mathbf{v} >= lmodk$ (provide a mechanism that sparsify the lattice)
- Approximation hints: $< s, \mathbf{v} >= l + e_\sigma$ (decrease the covariance of the secret)
- Short vector hints: $\mathbf{v} \in L$ (Lattice is projected orthogonally to $\mathbf{v}$)

III. Introduction to SIKE, plus related fault attack and countermeasure

SIKE is the only submission to the NIST PQC Standardization Process based on isogenies between elliptic curves. It is characterized by a relatively slow speed in comparison to other candidates and a small key size. Following a brief

SIKE description, the hardware attack threat is introduced. We will especially focus on our experimental validation of a fault injection attack proposed by Ti in 2017 [34]. We manage to recover the secret thanks to electromagnetic fault injection on an ARM Cortex A53 using a correct and an altered public key generation. We will show that countermeasures to detect this fault attack in SIKE implementations have a low overhead due to existing redundancy. This section is a short version of [35].

A. An Introduction to SIKE

1) Preliminaries: First, we are going to present a few mathematical tools and concepts that are used in SIKE. We will start by briefly introducing elliptic curves and isogenies using [36]. The latter is presented for cryptography use in [37].

Definition 1. *Let K be a finite field such that char$(K) \neq 2$ and A, $B \in \mathbb{F}_{p^2}$ such that $B(A^2-4) \neq 0$. The Montgomery (elliptic) curve $E_{A,B}$ consists of a point at infinity O and the set of points $(x, y) \in \mathbb{F}_{p^2}$ such that $By^2 = x^3 + Ax^2 + x$.*

An addition law can be defined on the set of points of an elliptic curve, hence this set has a group structure. In particular, we are interested in two kinds of points. Let E be such a curve, P, Q points on E, t and k positive integers. P is a t-torsion point if $tP = O$ and Q is of order k if k is the smallest integer such that $kQ = O$.

From now on, we will suppose that $B = 1$, as in SIKE. As shown in [38], Montgomery curves can be represented by a triplet of x-coordinates (x_P, x_Q, x_R) of points P, Q and R such that $P \neq Q$ and $R = P - Q$. To improve readability, we will however use points instead of x-coordinates on figures only. Moreover, a j-invariant can be defined [38].

Definition 2. *Let E be a Montgomery curve as above. Then the j-invariant of E is*

$$j(E) = \frac{256(A^2-3)^3}{A^2-4}.$$

Thus we get equivalence elliptic curves classes [36, § III.1].

Proposition 3. *Two elliptic curves are isomorphic over the algebraic closure of their definition field if and only if they have the same j-invariant.*

Isogenies are maps between these equivalence classes. More precisely, let E and F be two elliptic curves defined over a finite field K. An isogeny ϕ between E and F is a non-trivial group morphism between E and F. The isogenies used in SIKE are separable and thus can be uniquely defined by their respective kernels. It is possible to compute the expression of an isogeny knowing said kernel with formulas proposed by Vélu [39]. After the above mathematical review the main building blocks of SIKE are described.

2) SIDH: The goal of the supersingular isogeny Diffie-Hellman (SIDH) key exchange is for Alice and Bob to share a secret. They have at their disposal public data: a supersingular elliptic curve E_0 defined on $\mathbb{F}_{p^2}$ with $p = 2^{e_2}3^{e_3} - 1$, points P_2, Q_2 of order 2^{e_2} and R_2 such that $R_2 = P_2 - Q_2$ and points P_3, Q_3 of order 3^{e_3} and R_3 such that $R_3 = P_3 - Q_3$.

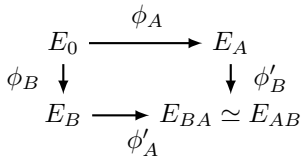

Fig. 1. The SIDH key exchange.

Their secret keys are scalars $\mathrm{sk}_2 \in [0, 2^{e_2 \log_2(2)} - 1]$ and $\mathrm{sk}_3 \in [0, 2^{e_3 \log_2(3)} - 1]$. The associated secret isogenies are ϕ_A and ϕ_B such that $\mathrm{Ker}(\phi_A) = \langle P_2 + \mathrm{sk}_2 Q_2 \rangle$ and $\mathrm{Ker}(\phi_B) = \langle P_3 + \mathrm{sk}_3 Q_3 \rangle$, and ϕ'_A and ϕ'_B such that $\mathrm{Ker}(\phi'_A) = \langle \phi_B(P_2) + \mathrm{sk}_2 \phi_B(Q_2) \rangle$ and $\mathrm{Ker}(\phi_B) = \langle \phi_A(P_3) + \mathrm{sk}_3 \phi_A(Q_3) \rangle$. By applying each of these isogenies to E_0 as shown on Figure 1, Alice and Bob will obtain two isomorphic elliptic curves E_{AB} and E_{BA}. Thus, as seen in Section III-A1, the j-invariant of these curves will be the shared secret.

SIKE is a key encapsulation mechanism based on SIDH. We are only going to focus on the public key computation step from now on as seen on Figure 2 and give all explanations for points P_3, Q_3 R_3 without loss of generality.

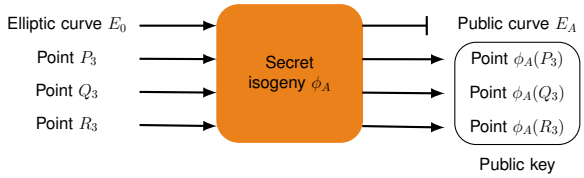

Fig. 2. Public key computation in SIKE.

A public key in SIKE is a triplet of x-coordinates of points $\phi_A(P_3), \phi_A(Q_3), \phi_A(R_3)$ composed of the images of the three public points P_3, Q_3 and R_3. Do note that the image of starting curve E_0 by isogeny ϕ_A is also computed in SIKE (see Figure 2), but not sent with the coordinate triplet as the public key because these three x-coordinates enable to compute the public curve coefficient.

After this presentation of the various components of SIKE, we will have a look at existing hardware attacks, more precisely at fault attacks.

B. The Hardware Attack Threat

Since the inception of SIDH in 2011 [40], there have been two fault attacks on isogeny-based cryptography. The attack by Gélin et al. [41] consists in stopping prematurely the loop of the shared secret isogeny computation to recover the secret. Countermeasures for such loop-abort attacks are presented in [42]. The attack we are going to focus on is a 2017 theoretical attack on SIDH by Ti [34]. We start by presenting the threat model of this attack. Using the same secret, the attacker will ask for two public key generations: the first one will be carried out correctly, while the second one will be altered by a fault. The attack occurs during the public key computation as shown on Figure 2. For the altered public key generation, instead of letting the key generator compute the

image of the three fixed public points by the secret isogeny, the attack will create a fault during the computation so that at least one image of a random point on the starting curve E_0 is computed instead of only the images of the fixed points. This altered image point $\phi_A(P_3)$ has a high probability to contain leaked information about the secret and will then enable a secret recovery by performing an analysis described in [34].

In the following section, we will show that this attack is practically exploitable in a laboratory.

C. Experimental Validation of Ti's Attack and Countermeasure

After this overview of existing hardware attacks on SIKE, we will focus on Ti's attack and show how we validated it in practice, and present the countermeasure we found.

1) Set up of an attack campaign and experimental results: To check the feasibility of the attack in a laboratory, we decided to use the ARMv8-A implementation with x64 assembly optimizations of the public key generation of the SIKE round 3 submission [43]. We chose to attack a system on chip (SoC) with four Cortex-A53 cores at the maximum frequency of 1.2 GHz, the computations being only performed by CPU 3. While skipping a chosen instruction is difficult as there are latency issues in SoCs [44], it is not a problem when performing Ti's attack because we do not need a great precision as we only need to inject a fault during the public key generation, as seen in Section III-B. The set up of our attack campaign can be seen on Figure 3.

The control computer communicates with the oscilloscope, the target and its power supply, the pulse generator and the motorized stage. Upon receiving a trigger signal from the target, the computer launches the attack through the pulse generator that generates a tension pulse creating an electromagnetic field on top of the target thanks to the electromagnetic probe. Width, amplitude and delay of the pulse, i.e., the time between the beginning of the public key generation, in our case, and the injection, can be modified. The probe can be moved using the motorized XYZ stage. We decided on a fixed probe position and a pulse width of 6 ns during the campaign as these contribute to the fault injection [44]. Our goal was then to find the (amplitude, delay) configuration that is the most propitious to secret key recovery. We made 1,040,000 attempts in around 4.5 days. The highest success rate is 0.62% for an amplitude of 360 V and a delay of 440 ns, which is as if we were to find a secret every 3 minutes and 10 seconds.

2) Impact on SIKE and Countermeasure: In the threat model presented in Section III-B, we have seen that Ti's attack requires two public key generations using the same secret. There is no reason for it to happen if the KEM is correctly implemented, but it may happen if developers do not respect the KEM API. This vulnerability also appears de facto in a multiparty key exchange like the ones presented in [45]. Indeed, if for instance Bob wants to communicate with Alice and Charlie, he must generate one triplet for Alice and one for Charlie using the same secret. The attacker can then alter only one of the triplets, for instance Alice's, and still have a correct

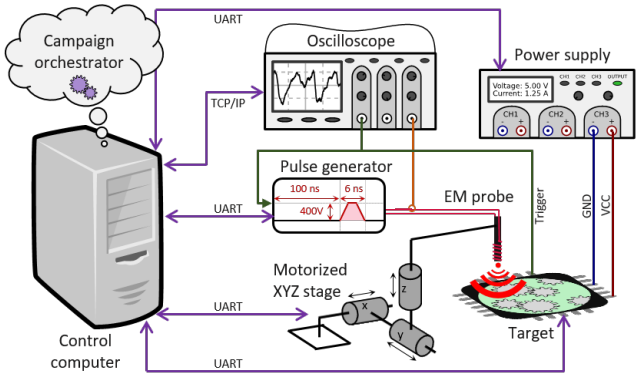

Fig. 3. Campaign setup.

one at his disposal, here Charlie's. Thus he can perform Ti's attack.

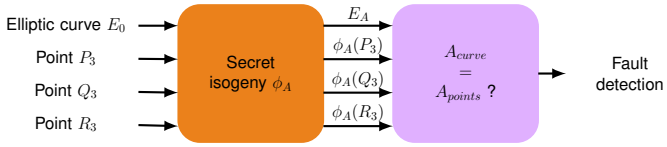

Fig. 4. Countermeasure for Ti's attack.

We propose then a countermeasure as seen on Figure 4. We have seen that the public key curve E_A is computed in the SIKE code but not used as the elliptic curve can be recovered from the public key point triplet. We thus propose to compare the coefficient A of E_A and the one from the curve computed using the triplet. If one of the input points has been altered, the probability that the two curve coefficients are different is high and the overhead is low as we use a redundancy present in SIKE's code.

IV. Leveraging Inexact Computing in Post-Quantum Cryptography

Hard learning problems are important building blocks for the design of various cryptographic functionalities such as authentication protocols and post-quantum public key encryption. The standard implementations of such schemes add some controlled errors to simple computations (e.g., inner products) involving a public challenge and a secret key. In parallel, the move towards nanoscale devices renders modern implementations increasingly prone to various types of errors. As a result, inexact computing has emerged as a new paradigm to efficiently deal with the challenges raised by such erroneous computations, and mitigate the cost and power consumption overheads they cause. In this paper, we show that these cryptographic and electronic challenges can actually be turned into new opportunities, and provide an elegant solution one to the other. That is, we show that inexact implementations of inner product computations lead to a natural way to define new Learning with Physical Noise or Error assumptions, paving the way to more efficient and physically secure implementations,

with potential interest for securing PQC implementations targeting lightweight applications.

A first step in this direction was proposed in [3], where the Learning Parity with Noise (LPN) problem has been re-formalized into the Learning Parity with Physical Noise (LPPN) problem, taking full advantage of inexact computation of an inner product operation to introduce errors. The approach clearly has some advantages. In classical LPN (and in many other Learning With Noise and Errors), the generation of the error is usually demanded to a standard Random Number Generator (RNG), that may require an high cost in terms of resources and may be vulnerable to many physical attacks. In [3] Kamel et al. discuss on the possibility to leverage on intrinsic noisy behavior of physical circuits to generate errors within the implementation itself, thus removing the need of an RNG. Simulated experiments have shown that the adopting frequency and voltage over-scaling, it would be possible to generate error according to some given distribution in a controlled manner. In conventional and synchronous CMOS circuits, registers are used to guarantee timing and correctness of processed data. They sample the value at their data input with a specific timing behavior (e.g., rising edge of the clock). Usually, data at their input after some computation (e.g., inner product) is not directly stable, and a number of transient oscillations, usually called glitches, take place before reaching a final and stable value. Such glitches are exploited to generate errors in LPPN, and, more interestingly, they can be controlled. First evidences of the concrete feasibility to implement such construction on a running prototype have been presented in [46], where a 28nm ASIC prototype implementing an LPPN primitive has been proven to be fully functional also in a broad range of working conditions, providing a low-energy and low-voltage solution for LPN-based authentication protocols. In [47], Kamel et al. show that the unprotected LPPN inherently provides levels of side-channel resistance such that masking will be effective. It has to be noted that given the key-homomorphic structure of the LPPN, a Boolean masked implementation of such primitive would have a quadratic cost in the number shares rather than quadratic, as it is common for block ciphers.

Clearly, leveraging on physical glitches to generate errors (thus, providing the security guarantees) for an LPPN construction, may rise concerns about data dependencies of errors themselves, hence opening for new challenges. In [48], a study of physical imperfections and data dependencies that can affect the security of the LPPN problem and implementations was presented, along with a fully digital prototype running on a Xilinx FPGA with a fault detection mechanism. Among data dependencies, output dependency of the error distribution generated within a LPPN processor has been identified as the most relevant. From a design perspective, different solutions may be applied to mitigate such dependency. Considering ASIC implementations, it has been noted that a strong reduction can be achieved if additional (and data-independent) jitter on the sampling clock is used. Authors also observed that balancing $0 \rightarrow 1$ and $1 \rightarrow 0$ propagation times in combinational gates (e.g., power gating cells on the path to ground and/or gate sizing) used for inner product computation helps in mitigating output dependency. Such solutions cannot be deployed on FPGAs, but simply adding output-invariant dummy operations contributes in reducing output dependency of the LPPN generated error distribution. It has to be remarked that the feasibility to implement inexact computing on FPGA platforms open to new interesting challenges and opportunities, as they are adopted in a plethora of applications due to their cost and reconfigurable nature. From a security perspective, it has been demonstrated that the security provided by LPPN's responses does not fundamentally differ from the security of LPN's ones. Hence, a new family of LPN problems have been proposed, covering this non-ideal behavior of LPPN implementations, denoted as LPN with Output Dependencies (LPN-OD).

These results naturally suggest the study of the Learning With Physical Noise (LWPN) problem from a design and security perspective, and its application to the secure and efficient implementation of PQC as a challenging next step. Clearly, new research questions arise from different directions to further extend this elegant paradigm to more general constructions. From the hardware viewpoint, it is natural to investigate about the feasibility of generating and controlling physical errors with complex distributions (e.g., as needed for LWE), always considering both ASICs and FPGAs. On the other hand, the other natural direction is in the regards of the possibility to find reductions of imperfect implementations to known hard learning problems, that would support all security constraints that such primitives would have to fulfill.

V. Conclusions

Post-Quantum Cryptography is a challenging application that is putting a lot of expectations on the hardware for achieving the desired security, performance and energy efficiency objectives. In this paper, we have given an overview of the implementation challenges involving some PQC algorithmic families. In addition, we have also shown how the usage of inexact computing can be interestingly leveraged in order to achieve highly efficient PQC implementations.

References

[1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.

[2] H. Nejatollahi, N. Dutt, and R. Cammarota, "Special session: trends, challenges and needs for lattice-based cryptography implementations," in *2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp. 1–3, 2017.

[3] D. Kamel, F. Standaert, A. Duc, D. Flandre, and F. Berti, "Learning with physical noise or errors," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 5, pp. 957–971, 2020.

[4] H. Nejatollahi, S. Gupta, M. Imani, T. S. Rosing, R. Cammarota, and N. Dutt, "Cryptopim: In-memory acceleration for lattice-based cryptographic hardware," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2020.

[5] 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*, pp. 238–268, 2018.

[6] "Falcon: Fast-fourier lattice-based compact signatures over ntru." https://falcon-sign.info/falcon.pdf.

[7] C. P. Rentería-Mejía and J. Velasco-Medina, "Hardware design of an ntt-based polynomial multiplier," in *2014 IX Southern Conference on Programmable Logic (SPL)*, pp. 1–5, 2014.

[8] A. C. Mert, E. Öztürk, and E. Savaş, "Design and implementation of encryption/decryption architectures for bfv homomorphic encryption scheme," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 353–362, 2020.

[9] S. Sinha Roy, F. Turan, K. Jarvinen, F. Vercauteren, and I. Verbauwhede, "Fpga-based high-performance parallel architecture for homomorphic computing on encrypted data," in *2019 IEEE International Symp. on High Performance Computer Architecture (HPCA)*, pp. 387–398, 2019.

[10] T. Pöppelmann and T. Güneysu, "Towards efficient arithmetic for lattice-based cryptography on reconfigurable hardware," in *Progress in Cryptology – LATINCRYPT*, pp. 139–158, 2012.

[11] A. Aysu, C. Patterson, and P. Schaumont, "Low-cost and area-efficient fpga implementations of lattice-based cryptography," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 81–86, 2013.

[12] T. Fritzmann and J. Sepúlveda, "Efficient and flexible low-power ntt for lattice-based cryptography," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 141–150, 2019.

[13] C. Li, W. zhu, and L. Liu, "A high speed ntt accelerator for lattice-based cryptography," in *International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2021.

[14] S. S. Roy, F. Vercauteren, N. Mentens, D. D. Chen, and I. Verbauwhede, "Compact ring-lwe cryptoprocessor," in *Cryptographic Hardware and Embedded Systems – CHES 2014* (L. Batina and M. Robshaw, eds.), (Berlin, Heidelberg), pp. 371–391, Springer Berlin Heidelberg, 2014.

[15] G. Zhang, P. Leong, D.-U. Lee, J. Villasenor, R. Cheung, and W. Luk, "Ziggurat-based hardware gaussian random number generator," in *International Conference on Field Programmable Logic and Applications, 2005.*, pp. 275–280, 2005.

[16] H. Edrees, B. Cheung, M. Sandora, D. B. Nummey, and D. Stefan, "Hardware-optimized ziggurat algorithm for high-speed gaussian random number generators," in *Proceedings of the 2009 International Conference on Engineering of Reconfigurable Systems & Algorithms, ERSA July 13-16, 2009, Las Vegas Nevada, USA* (T. P. Plaks, ed.), pp. 254–260, CSREA Press, 2009.

[17] R. Agrawal, L. Bu, and M. A. Kinsy, "A post-quantum secure discrete gaussian noise sampler," in *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 295–304, 2020.

[18] J. H. Silverman and W. Whyte, "Timing attacks on ntruencrypt via variation in the number of hash calls," in *Cryptographers' Track at the RSA Conference*, pp. 208–224, Springer, 2007.

[19] N. V. Vizev, *Side Channel Attacks on NTRUEncrypt*. PhD thesis, Bachelor's thesis, Technical University of Darmstadt, Germany, 2007. Available on http://www. cdc. informatik. tu-darmstadt. de/reports/reports/Nikolay_Vizev. bachelor. pdf, 2007.

[20] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, gauss, and reload–a cache attack on the bliss lattice-based signature scheme," in *International Conference on Cryptographic Hardware and Embedded Systems*, pp. 323–345, Springer, 2016.

[21] N. Bindel, J. Buchmann, J. Krämer, H. Mantel, J. Schickel, and A. Weber, "Bounding the cache-side-channel leakage of lattice-based signature schemes using program semantics." Cryptology ePrint Archive, Report 2017/951, 2017.

[22] A. Park and D.-G. Han, "Chosen ciphertext simple power analysis on software 8-bit implementation of ring-lwe encryption," in *IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*, pp. 1–6, IEEE, 2016.

[23] A. Atici, L. Batina, B. Gierlichs, and I. Verbauwhede, "Power analysis on ntru implementations for rfids: First results," in *Workshop on RFID Security*, Sl: sn, 2008.

[24] M.-K. Lee, J. E. Song, D. Choi, and D.-G. Han, "Countermeasures against power analysis attacks for the ntru public key cryptosystem," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 93, no. 1, pp. 153–163, 2010.

[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, Apr. 2018.

[26] P. Pessl, "Analyzing the shuffling side-channel countermeasure for lattice-based signatures," in *Progress in Cryptology–INDOCRYPT 2016: 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings 17*, pp. 153–170, Springer, 2016.

[27] R. Primas, P. Pessl, and S. Mangard, "Single-trace side-channel attacks on masked lattice-based encryption," in *Cryptographic Hardware and Embedded Systems - CHES 2017 Taipei, Taiwan, September 25-28*, pp. 513–533, 2017.

[28] N. Bindel, J. Buchmann, and J. Krämer, "Lattice-based signature schemes and their sensitivity to fault attacks," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), Workshop on*, pp. 63–77, 2016.

[29] F. Valencia, T. Oder, T. Güneysu, and F. Regazzoni, "Exploring the vulnerability of r-lwe encryption to fault attacks," in *Proceedings of the Fifth Workshop on Cryptography and Security in Computing Systems*, CS2 '18, (New York, NY, USA), pp. 7–12, ACM, 2018.

[30] F. Valencia, I. Polian, and F. Regazzoni, "Fault sensitivity analysis of lattice-based post-quantum cryptographic components," in *2019 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, IEEE, July 2019.

[31] P. Ravi, M. P. Jhanwar, J. Howe, A. Chattopadhyay, and S. Bhasin, "Exploiting determinism in lattice-based signatures: Practical fault attacks on pqm4 implementations of NIST candidates," in *AsiaCCS 2019, Auckland, New Zealand, July 09-12*, pp. 427–440, 2019.

[32] S. McCarthy, J. Howe, N. Smyth, S. Brannigan, and M. O'Neill, "BEARZ attack FALCON: implementation attacks with countermeasures on the FALCON signature scheme," *IACR Cryptology ePrint Archive*, vol. 2019, p. 478, 2019.

[33] D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, "Lwe with side information: Attacks and concrete security estimation," in *Advances in Cryptology – CRYPTO 2020* (D. Micciancio and T. Ristenpart, eds.), (Cham), pp. 329–358, Springer International Publishing, 2020.

[34] Y. B. Ti, "Fault attack on supersingular isogeny cryptosystems," in *International Workshop on Post-Quantum Cryptography*, pp. 107–122, Springer, 2017.

[35] É. Tasso, L. De Feo, N. El Mrabet, and S. Pontié, "Resistance of isogeny-based cryptographic implementations to a fault attack." Cryptology ePrint Archive, Report 2021/850, 2021.

[36] J. H. Silverman, *The arithmetic of elliptic curves*, vol. 106. Springer Science & Business Media, 2009.

[37] L. De Feo, "Mathematics of isogeny based cryptography," *CoRR*, vol. abs/1711.04062, 2017.

[38] C. Costello and B. Smith, "Montgomery curves and their arithmetic," *Journal of Cryptographic Engineering*, vol. 8, no. 3, pp. 227–240, 2018.

[39] J. Vélu, "Isogénies entre courbes elliptiques," *CR Acad. Sci. Paris, Séries A*, vol. 273, pp. 305–347, 1971.

[40] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *International Workshop on Post-Quantum Cryptography*, pp. 19–34, Springer, 2011.

[41] A. Gélin and B. Wesolowski, "Loop-abort faults on supersingular isogeny cryptosystems," in *International Workshop on Post-Quantum Cryptography*, pp. 93–106, Springer, 2017.

[42] J. Proy, K. Heydemann, A. Berzati, and A. Cohen, "Compiler-assisted loop hardening against fault attacks," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 14, no. 4, pp. 1–25, 2017.

[43] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Hutchinson, A. Jalali, K. Karabina, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, G. Pereira, J. Renes, V. Soukharev, and D. Urbanik, "SIKE: supersingular isogeny key encapsulation," 2020.

[44] C. Gaine, D. Aboulkassimi, S. Pontié, J.-P. Nikolovski, and J.-M. Dutertre, "Electromagnetic fault injection as a new forensic approach for SoCs," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, 2020.

[45] R. Azarderakhsh, A. Jalali, D. Jao, and V. Soukharev, "Practical supersingular isogeny group key agreement.," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 330, 2019.

[46] D. Kamel, D. Bellizia, F. Standaert, D. Flandre, and D. Bol, "Demonstrating an LPPN processor," in *Workshop on Attacks and Solutions in Hardware Security, ASHES@CCS 2018, Toronto, ON, Canada, October 19, 2018*, pp. 18–23, ACM, 2018.

[47] D. Kamel, D. Bellizia, O. Bronchain, and F. Standaert, "Side-channel analysis of a learning parity with physical noise processor," *J. Cryptogr. Eng.*, vol. 11, no. 2, pp. 171–179, 2021.

[48] D. Bellizia, C. Hoffmann, D. Kamel, H. Liu, P. Méaux, F.-X. Standaert, and Y. Yu, "Learning parity with physical noise: Imperfections, reductions and fpga prototype," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 3 (to appear), 2021.