On Secure and Side-Channel Resistant Hardware Implementations of Post-Quantum Cryptography

Petr Jedlicka
@vut.cz
Brno University of Technology
Czech Republic

Lukas Malina
malina@vut.cz
Brno University of Technology
Czech Republic

Petr Socha
petr.socha@fit.cvut.cz
Czech Technical University in Prague
Czech Republic

Tomas Gerlich
@vut.cz
Brno University of Technology
Czech Republic

Zdenek Martinasek
martinasek@vut.cz
Brno University of Technology
Czech Republic

Jan Hajny
hajny@vut.cz
Brno University of Technology
Czech Republic

ABSTRACT

Currently, many post-quantum cryptography schemes have been implemented on various hardware platforms in order to provide efficient solutions in cybersecurity services. As researchers and hardware developers focus primarily on designs providing small latency and requiring fewer hardware resources, their implementations could seldom omit protection techniques against various physical attacks. This paper studies potential attacks on the cryptography implementations that run on Field-Programmable Gate Array (FPGA) platforms. We mainly analyze how Post-Quantum Cryptography (PQC) implementations could be vulnerable on various platforms. Further, we aim at the FPGA-based implementations of National Institute of Standards and Technology (NIST)'s PQC competition finalists. Our study should present to developers the current overview of attacks and countermeasures that can be implemented on specific PQC schemes on FPGA platforms. Moreover, we present novel implementation of one universal countermeasure component and reveal additional resources that are needed.

CCS CONCEPTS

• **Security and privacy → Side-channel analysis and countermeasures**; **Cryptanalysis and other attacks**.

KEYWORDS

Applied Cryptography, FPGA, Hardware Implementation, Post-Quantum Cryptography, Secure Implementation, Side Channel Attacks

ACM Reference Format:
Petr Jedlicka, Lukas Malina, Petr Socha, Tomas Gerlich, Zdenek Martinasek, and Jan Hajny. 2022. On Secure and Side-Channel Resistant Hardware Implementations of Post-Quantum Cryptography . In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3538969.3544423

ARES 2022, August 23–26, 2022, Vienna, Austria

ACM ISBN 978-1-4503-9670-7/22/08…$15.00
https://doi.org/10.1145/3538969.3544423

1 INTRODUCTION

Secure and efficient hardware-based implementations of cryptographic schemes are essential for reliable ICT services. Designers and developers often aim at efficiency but may overlook potential risks and threats that can be caused by advanced physical attacks such as timing, power, electromagnetic side-channels, faults injections, and hardware trojans. Fortunately, numerous countermeasures can be deployed into hardware-based implementations of cryptography schemes. In this paper, we pay attention to post-quantum cryptography as the National Institute of Standards and Technology (NIST)'s Post-Quantum Cryptography (PQC) competition approaching to its final and new standards will be released. On the one hand, hardware-based implementations can mitigate the efficiency disadvantages of robust post-quantum cryptographic schemes and accelerate these operations. On the other hand, designing the secure hardware implementations of PQC that are resisted to physical attacks usually adds a certain number of cycles and hardware resources. Then, it can sometimes be challenging to upload such implementations to FPGA boards with small hardware resources represented by the number of Logic Cells, Look-Up Tables (LUTs), Flip-Flops (FFs), Digital Signal Processor (DSP) slices, and Block Random Access Memories (BRAMs). Currently, Xilinx offers numerous families of FPGA platforms having various sizes, and small FPGA platforms typically offer less than 100k logic cells (ca 63k LUTs/126k FFs) and tens of BRAMs. HW-based PQC accelerators that combine security (attack resistance), efficiency (small latency; the high number of operations per second), and having minimal hardware resources can be reasonable options for emerging intelligent infrastructures, IoT, and smart cities applications that run at the end-points open to various physical attacks. Therefore, FPGA-based security applications have to find a balance between all the above aspects.

1.1 Related work

In the last two decades, there are many works, e.g., [5, 23, 30, 32, 35, 40–42, 44] that study various physical attacks, threats, and countermeasures aimed at the hardware-implementations of symmetric and asymmetric cryptography at FPGA platforms. For instance,

Wollinger *et al.* [44] discussed security issues on FPGAs and summarized the security of public and symmetric-key algorithm implementations on FPGAs in 2004. In 2006, Standaert *et al.* [41] overviewed power analysis attacks against FPGA, discussed the protection techniques, and compared SW and HW based implementations. Recently, general security vulnerabilities caused by FPGA design and specific run-time physical properties such as power consumption, temperature, electromagnetic emission, and long-wire crosstalk coupling have been discussed in [30]. Then, J. Zhang and G. Qu in [45] have reviewed the security and trust issues related to FPGA-based systems from the market perspective.

As the NIST PQC standardization approached the final (3^{rd}) round, few research works also focused on attacks aimed directly at PQC at FPGA, such as [1, 14, 15, 18, 20, 21]. General challenges and issues of post-quantum cryptography in hardware were discussed by Gaj in [15]. Other works often focused on some concrete scheme and its improvement against physical attacks at FPGA. For instance, Howe *et al.* [18] discussed the countermeasures e.g. error samplers for lattice-based cryptography implemented at FPGA. Abdulgadir *et al.* [1] presented a hardware implementation of Saber key encapsulation mechanism resistant against side-channel attacks. Jati *et al.* [20] dealt with configurable Crystals-Kyber hardware implementation with the side-channel protection that consumes only additional 5% of HW resources. More complex view on power-based side channel attack analysis on PQC provided the work [21] but this study is mainly focused on the development of a multi-target and multi-tool platform to conduct test vector leakage assessment.

In this work, we provide a novel and updated overview of current threats and attacks related to hardware-based implementations of post-quantum cryptography and discuss also efficiency of existed countermeasures.

1.2 Contributions and Paper Organization

This work focuses mainly on these research questions: *1) Which post-quantum cryptography schemes are most jeopardized by attacks at their hardware-based implementations at FPGA platforms? 2) How secure and efficient are current countermeasures applied to hardware implementations of PQC at FPGA?*

Questions 1) and 2) are studied in Sections 2 and 3 where we map existing general threats and also attacks aimed at the hardware implementations of PQC and discuss the current security countermeasures and protections used in hardware implementations. Furthermore, Section 4 presents our experimental implementation of a universal countermeasure that is designed to be easily applicable to all cryptosystems. We conclude this paper with a future research discussion in Section 5.

2 CURRENT THREATS AND ATTACKS AT HARDWARE PLATFORMS

This section presents a basic overview of general attacks that aim at cryptography implementations. Then, we briefly introduce our experimental testbed and equipment that serve for our testing. Further, we study current attacks aimed at hardware-based implementations of PQC schemes.

2.1 General Attacks Focused on Implementations

The implementations of cryptography schemes may suffer numerous imperfections and flaws that can be targeted by the following attacks.

2.1.1 Glitch Attack. This attack targets the outputs of the AES cipher SBox [27]. It exploits glitches that occur during transitions between CMOS states. The glitches are dependent on the inputs of the SBox, thus leaking sensitive information. Using the correct model that can be obtained by simulating the SBox, the attack was successful even on the masked scheme.

2.1.2 Differential Side-Channel Attack. Differential side channel attack (DPA) uses statistics to extract secret information from measured traces [25]. DPA works on the following principle. The cryptographic device processes the input data sequentially using a static key. While the input data is being encrypted, the measurement of the selected side channel is recorded and stored. An attacker then selects intermediate values of the algorithm which depend on both the secret key values and the input data. With a key estimation and the input data the attacker computes the intermediate result and converts it to the predicted leakage model. If the key is estimated correctly, a correlation between the measurement and the leakage model will be found at some point in time. This correlation can be revealed using a different statistical approaches, a difference of means test, Pearson correlation or Spearman's rank correlation [12]. A countermeasure for DPA is randomizing the intermediate data then the leakage model will be unusable.

2.1.3 Template Attack. A template attack is a very efficient side-channel attack (SCA) [6]. This attack can break implementations with countermeasures that limit an adversary to obtain only a limited number of side-channel (SC) samples. The first requirement for this type of SCA is how each access to the identical device that can be programmed. The second requirement is a creation of large number of templates in an adaptive manner. The adversary's identical device categorizes small parts of SC samples. Due to same time the adversary builds templates corresponding to different value of unknown keys. The templates are used to classify portions of SC samples to limit possible key bits. Countermeasures for this attack are using randomness in computation applied at different devices such as data scrambling and masking in order to confuse the adversary who cannot obtain same results from devices.

2.1.4 Fault Injection. These techniques are designed to change the behavior of a computing device [3]. The Fault Injection (FI) attacks include these malicious and unwanted changes, such as, changing the supply voltage, changing the clock frequency, exposing the device to electromagnetic radiation, heating the device, or exposing it to strong light radiation. With low voltage, errors occur uniformly throughout the computation and then, this type does not usually need much power. This cause that an attacker must recognize results that are unsuitable for a successful attack. The attacker must be equipped with own voltage source and have access to a target device. No implementation knowledge or advanced skills are required to perform this attack. Once the attack is completed, there will be no evidence of tampering with the device. Another technique is

to use a focused light beam to change the state of one or more logic gates. By irradiating the transistors, a conductive channel is created and the state is changed in a selected area of the circuit. These techniques can also be used by circuit designers to detect potential threats by implementing them according to the attacker's capabilities.

2.2 SCA Experimental Testbed

In order to perform own research experiments with power and electromagnetic side-channels, we set an experimental testbed, depicted in Figure 1. The main devices is the FPGA board SAKURA-G that is designed for testing hardware security of various implementations. Its two programmable logical circuits can be configured to users' desire by VHDL or Verilog code. The second depicted device is high sampling frequency oscilloscope Agilent MSO3103B. Its function is the measurement of power consumption or electromagnetic emission, depend on a type of a probe connected to the oscilloscope. Each measurement is then transferred to a personal computer. PC and its software have an important role in tests. The software part controls feeding data to the SAKURA-G board and saves measurements from the oscilloscope for the next processing depending on the type of attacks.

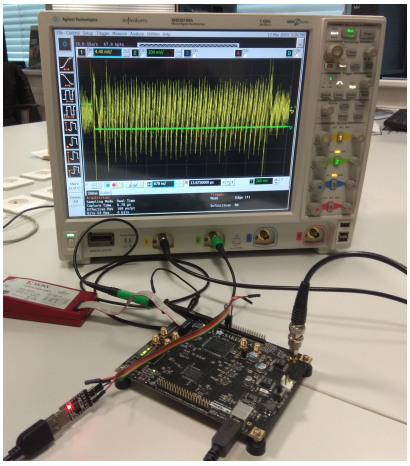

Figure 1: Experimental testbed with SAKURA G.

2.3 Attacks Aimed at Hardware-based Post-Quantum Cryptography

In this section, we mention attacks related to the post-quantum cryptosystems that are among the finalists of the currently ongoing NIST post-quantum standardization. Some attacks are applicable to all algorithms of a particular post-quantum cryptography family, and some attacks are specific to concrete schemes.

2.3.1 Attacks on Error Samplers: So-called error samplers are used to generate additive noise, which is needed for hiding the secret in lattice-based schemes whose hardness is based on variants of the LWE problem. Security attacks could target this crucial component. The error samplers could be attacked by extracting the noise via side channels, but there is currently no known attack on FPGA. Another way is to disrupt the proper function of the sampler and set its output to a constant or any other values easily predictable for an adversary. This fault attack was presented by [11] who injected the faults to an FPGA through clock glitches.

2.3.2 Cold Boot Attacks: Cold boot attack aims to minimize the bit-flip rate of data saved in a volatile memory after it is powered down. The memory is frozen, removed from the device comprising the cryptographic scheme, and subsequently analyzed by an adversary. The lower the temperature at which the memory is frozen, the lower the bit decay rate. According to the first cold boot attack experiment published in [16], to achieve about only 0.1 % of bits decay within one minute, it is necessary to freeze the memory at -50 °C. To extend the time to one hour, a temperature of approximately -196 °C is needed. In general, this attack is applicable to various cryptographic schemes, but its success rate depends on the format of saved keys and on the acceptable noise level added to the keys to have the successful attack [16]. An attack on secret keys of Kyber was demonstrated in [2]. The attack was able to find between 60 % and 90 % of the secret coefficients. The experiment shows that saving the secret coefficients in the NTT (*Number Theoretic Transform*) domain decreases the level of the acceptable additive noise caused by the bit decay. In other words, saving the secret key in the NTT domain makes the implementation more secure. Although the attack in [2] is targeted at PC DRAM memories, the attack would also be applicable to hardware implementations because they use memories based on the same principles. Possible cold boot attacks on post-quantum schemes, especially on NTRU, were also analyzed in [34] but they do not show any concrete experimental results.

2.3.3 Side-Channel Analysis of McEliece: McEliece is the only post-quantum cryptosystem based on error-correcting codes that has advanced to the current third round of the NIST post-quantum standardization.

Several works dealt with the differential leakage analysis of McEliece. For example, Chen *et al.* [7] presented horizontal and vertical side-channel analysis techniques on an FPGA-based implementation presented in [43]. Concretely, vertical differential power analysis of the syndrome computation and horizontal differential power analysis of the key rotation were applied to the implementation. Using this combination of attacks, a complete secret key was recovered after a few analyzed decryptions. Given the nature of error-correction codes, which are designed to work properly even with some bits of an incorrect value, it was possible to recover the key even in the case of the knowledge of not all bits [7]. Differential power analysis of McEliece was also presented in [36], but it was applied to an ARM-based implementation.

2.3.4 Fault Injection Attacks on McEliece: In the paper [4], the authors dealt with the sensitivity to fault injections. After a series of attempts using various approaches, they came to the conclusion that McEliece is resistant to this type of the attack if a code with sufficient capacity is used.

2.3.5 Side-Channel Analysis of Rainbow: Rainbow [10] is a multivariate quadratic signature scheme based on the Unbalanced Oil and Vinegar [24] scheme. The signature is derived from a solution of a quadratic equation system obtained from a private-key quadratic map F (a set of polynomials), which is structured so that a linear

solver can solve the system during the signing process. For public use, the polynomial structure is hidden using two private-key linear maps S, T, resulting in a public quadratic map $P = S \circ F \circ T$. Then the private signing key is a tuple (S^{-1}, F, T^{-1}) of the two linear maps and the quadratic map, and the public key is the quadratic map (P). Note that with knowledge of the S and T maps, the structured quadratic map F can be feasibly obtained from the public quadratic map P.

A correlation power analysis attack on a software Rainbow implementation in an 8-bit microcontroller is described in [33]. The authors present a way to extract the S and T linear maps by targeting the matrix multiplication (maps application) during the signing process, effectively compromising the signature scheme. An extended version of the attack is presented in [37], where the authors attack an implementation in a 32-bit microcontroller. Both mentioned attacks assume a Hamming weight leakage model. However, the latter attack predicts two consecutive subkeys to obtain the power predictions; such a prediction is also suitable in a Hamming distance leakage model, which applies to hardware implementations in general. Furthermore, since the correlation power analysis attack is possible, a profiled attack on the matrix multiplication is almost certainly applicable, although to the best of our knowledge, it has not been published yet. Considering the profiling scenario, an attack on the central quadratic map should also be possible but more costly.

2.3.6 Fault Attacks on Rainbow: Fault attacks on Rainbow were proposed and examined in [17, 26]. These attacks target either the quadratic map coefficients or used random numbers. While the authors show the attacks both have a high success probability, neither attack allows for a full key recovery. The authors in [26] conclude a low overall susceptibility of the multivariate schemes to fault attacks.

2.3.7 Correlation Power Analysis on FALCON:. A non-profiled CPA attack on FALCON is presented in [22]. Unlike the other PQC candidates, the FALCON uses the traditional Fast Fourier Transformation (FFT) over floating points instead of the discrete Number Theoretic Transform (NTT). The authors target their attack on the floating-point results within the FFT and evaluate their attack on an ARM microcontroller. They demonstrate a successful key recovery using approximately 10,000 measurements.

2.3.8 Side-Channel Analysis of CRYSTALS-Dilithium: An efficient non-profiled attack on the Dilithium signature scheme is described in [8]. The authors target the underlying polynomial multiplication arithmetic, and they successfully recover the private key from the reference implementation running on ARM using only 157 power traces.

Furthermore, a profiled attack on the Dilithium is presented in [29], where the authors target the bit unpacking procedure during the signature algorithm. Using a combination of the machine learning and traditional statistical techniques, the authors are able to extract enough information to forge a signature on any message. Their proof of concept is once again evaluated on an ARM micro-controller.

2.3.9 Fault Injection Attack on CRYSTALS-Kyber: Based on a decryption error, an artificial error is inserted into the first ciphertext component, leaving the other components unchanged [9]. Depending on the value of the secret key, a decryption error can be achieved if the re-encrypted message and the original one differ by a single bit. Since a random value derived from the message is used in the re-encryption, a situation may arise where a decryption error does not occur. This event can be intercepted by a side channel and used to obtain information about the secret key.

2.4 Summary

Table 1 maps current attacks aimed at the hardware implementations of PQC schemes. Each cryptosystem has some vulnerabilities that could be exploited by at least one attack. Some attacks are applicable to more cryptosystems. For example, the fault injection attack on the error sampler described in [11] could be applied to all lattice-based algorithms. On the other hand, there are also attacks that can be used for only one specific cryptographic scheme, e.g., attacks on Rainbow.

3 COUNTERMEASURES IN HARDWARE IMPLEMENTATIONS

The section provides an overview of general and specific countermeasures against attacks on the implementations of post-quantum schemes.

3.1 General Countermeasures for Hardware-based Implementations

3.1.1 Countermeasures for Sbox Glitch: The countermeasure focuses on the information leakage that occurs in the non-linear part at the output of the SBox [28]. The multipliers that are the source of information leakage are part of the SBox. The proposed countermeasures against data leakage are using artificial delay or using enable signal. The additional delay may not be appropriate for all types of implementations. The enable signal will increase implementation complexity and increase resource utilization.

3.1.2 Countermeasures for Elliptic Curve Fault Attack : Point Validation is used to check if a point lies on the selected curve [13]. The validation should be evaluated before and after scalar multiplication. If the point or result does not lie on the curve, the result will not be displayed. usable against DFA attacks

Curve Integrity Check is used to find fault curve parameters [13]. The parameters are checked in memory before scalar multiplication. Then, it is possible to detect errors during scalar multiplication.

3.2 Countermeasures for Hardware-based Post-Quantum Cryptography

Several countermeasures and protection techniques for hardware PQC implementations have been introduced in recent studies. Some countermeasures are proposed for a certain post-quantum cryptography family or for a concrete scheme. Such techniques can hardly be adjusted to different schemes.

3.2.1 Countermeasures for Error Samplers: Two types of countermeasures to protect error samplers against attacks described in the section 2.3.1 were published. The first method discussed by [39] is based on shuffling the output coefficients of the error sampler.

Table 1: Hardware-based Attacks on Post-Quantum Cryptography NIST Finalists Implementation on FPGA.

Scheme	SCA	Fault Injection
Encryption/KEM NIST PQC Finalists		
Kyber	Cold boot attack (2018) [2]	Attack on the Fujisaki-Okamoto transform (2021) [9] Attack on error samplers (2018) [11]
McEliece	Differential power analysis (2016) [7] [36]	✗
NTRU	✗	Attack on error samplers (2018) [11]
SABER	✗	Attack on the Fujisaki-Okamoto transform (2021) [9] Attack on error samplers (2018) [11]
Signature NIST PQC Finalists		
Dilithium	Correlation power analysis (2022) [8] [29]	Attack on error samplers (2018) [11]
FALCON	Correlation power analysis (2021) [22]	Attack on error samplers (2018) [11]
Rainbow	Correlation power analysis (2021) [33] [37]	Attack on the quadratic map (2021) [17] [26]

Note: ✗– no attack could be found.

Shuffling works as a protection against side-channel attacks to provide the adversary with improperly ordered samples of the additive noise. Another method described in [19] computes statistical parameters over error sampler output and checks its correspondence to a given distribution.

3.2.2 Countermeasures for Cold Boot Attacks: To the best of the author's knowledge, any countermeasure adoptable to the hardware implementations of post-quantum cryptography has not been published yet. However, shuffling mentioned in the previous paragraph could be one of the methods how to make the secret data extraction more complicated. Spreading the data set among different areas in a memory or the application of unknown encoding before storing could also be considered. After all, finding the right position in the memory is not a trivial task itself.

3.2.3 Countermeasure for McEliece. As it has been mentioned above, an attack using the differential power analysis was discussed in the papers [7] and [36]. The same author also mentioned some countermeasure possibilities. In paper [7], the authors suggested massive parallelization and shuffling as a method how to prevent the attack. Another countermeasure was described in [36] where the authors mask the cryptosystem by adding Goppa codewords to a ciphertext during the permutation process.

3.2.4 Countermeasures for Rainbow: The known attacks (as described in subsection 2.3.5) aim at the matrix multiplication, which implements the linear maps S and T. The matrix describing S (or T) can be trivially split in two (or more) matrices S_1, S_2 such that S_1 is randomly chosen and $S_2 = S - S_1$. Then by the distributive property, it holds that $x(S_1 + S_2) = xS_1 + xS_2$. This approach allows for a secure matrix multiplication implementation [38] regardless of the implementation platform. However, this does not secure other parts of the implementation where unknown attack vectors may still exist. Another approach to secure the Rainbow implementation is a data blinding, where the signed digest is multiplied by a scalar mask [33], which can be used during the whole signing process [37]. However, the multiplicative masking is inherently unable to mask a zero value. Furthermore, in the case of Rainbow, a scalar mask results in a low masking entropy. Besides algorithm-level countermeasures, the Rainbow implementation is well suitable for execution flow randomization, e.g., indexes during the matrix multiplications may be randomly permuted.

Several countermeasures against fault attacks are proposed in [26]. These include checking for a faulty map, checking the randomness, or increasing the chances for vinegar variables to be zero. A fault analysis-resistant FPGA implementation of Rainbow is presented in [31], where the private keys are being checked for faults.

3.2.5 KYBER Fujisaki-Okamoto Transform. A transformation is based on the principle that no sensitive information should reach the attacker in case a corrupted ciphertext is entered [9]. The transformation does not protect against attacks to obtain intermediate results during the execution of the algorithm. After the ciphertext is entered, the ciphertext is decrypted and re-encrypted and compared with the entered ciphertext. In case of deviations, the way in which the messages differ must not be leaked to the attacker.

3.3 Summary

Table 2 maps current countermeasures applied at hardware-based implementations of PQC schemes. The countermeasures cover most of the attacks from Table 1. However, there are still attacks for which any countermeasure has not already been published. These attacks include the correlation power analysis for the lattice-based algorithms and the cold boot attacks.

4 EXPERIMENTAL IMPLEMENTATION OF HIDING IN TIME DOMAIN AND ITS EFFICIENCY

In this section, we describe our experimental hardware implementation of hiding in the time domain. The main idea was to propose a universal solution that would be easily applicable to every cryptosystem with minimal modification of the hardware implementation of the cryptosystem. For this reason a component that randomizes the clock enable (CE) signals of the registers in an FPGA was created. Since the inputs for CE signals are already prepared in the registers, the hardware utilization of the cryptosystem is unchanged. Moreover, the only needed modification of the code of the cryptographic component is adding the CE input to its interface and adding the CE signal to the statement checking the rising edge of the clock signal. A simple block diagram of the countermeasure implementation is in Figure 2.

Table 2: Countermeasures for FPGA-based Implementations of PQC NIST Finalists.

Scheme	Against SCA	Against Fault Injection
Encryption/KEM NIST PQC Finalists		
Kyber	Attack on error samplers (2017) [39]	Attack on the Fujisaki-Okamoto transform (2021) [9] Attack on error samplers (2019) [19]
McEliece	Differential power analysis (2016) [7] [36]	✗
NTRU	Attack on error samplers (2017) [39]	Attack on error samplers (2019) [11]
SABER	Attack on error samplers (2017) [39]	Attack on the Fujisaki-Okamoto transform (2021) [9] Attack on error samplers (2019) [11]
Signature NIST PQC Finalists		
Dilithium	Attack on error samplers (2018) [11]	Attack on error samplers (2019) [11]
FALCON	Attack on error samplers (2018) [11]	Attack on error samplers (2019) [11]
Rainbow	Correlation power analysis (2021) [38] [37]	Attack on the quadratic map (2021) [26]

Note: ✗– no countermeasure published yet

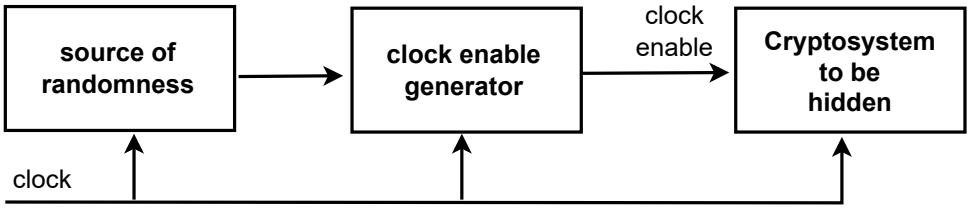

Figure 2: Block Diagram of Countermeasure Implementation.

The implementation needs some source of randomness. We used samples of atmospheric noise stored in block RAM memory for example. The second part receives the random data and computes the randomized CE signal. Three different methods of the randomization of the CE signal were developed, and the hardware utilization of the clock enable generator slightly differs in dependence on the chosen method. These methods are described in Sections 4.1, 4.2 and 4.3 and the hardware utilization of these methods are summarized in Table 3. The results show that the hardware utilization can be minimal for all methods. This makes them suitable for the implementation on various hardware platforms without numerous hardware resources.

Table 3: Hardware Utilization of Countermeasures.

LUT	FF	DSP	BRAM	LUTRAM
Completely random CE signal				
21	36	0	1	0
CE signal with constant duty cycle				
57	83	0	1	0
CE signal with variable duty cycle				
62	98	0	1	0

4.1 Completely Random CE Signal

The generated CE signal corresponds to the bit sequence of the random signal in the BRAM memory. Assuming that logic zeros and logic ones are equally represented in the random sequence, the execution time of the algorithm is approximately doubled to twice the original value. An example of the time course of the clock signal and the CE signal is shown in Figure 3.

4.2 CE Signal with Constant Duty Cycle

At the beginning of the execution of the cryptographic algorithm, a random value is used to calculate the CE signal duty cycle, which is unchanged until the algorithm terminates. The disadvantage of this approach is that the total execution times of the different runs of the algorithm vary widely from each other. Depending on the computed duty cycle, some algorithm runs may take many times longer, and others may be close to the original value. Examples of the time courses of the clock signal and CE signal are shown for different values of the duty cycle in Figures 4 and 5.

4.3 CE Signal with Variable Duty Cycle

The disadvantage of the previous method, in which the execution times of the individual runs of the algorithm varied significantly, was eliminated by using a CE signal with a variable duty cycle during one run of the algorithm. Assuming that logic zeros and logic ones are equally represented in the random sequence, the execution time of the algorithm is approximately doubled to twice the original value. An example of the time course of the clock signal and the CE signal is shown in Figure 6.

5 CONCLUSION

In this paper, we analyzed currently existing side-channel attacks and fault attacks, and countermeasures targeting post-quantum cryptography schemes that advanced to the currently ongoing third round of the standardization process for the post-quantum cryptography under the auspices of NIST. Moreover, we described a

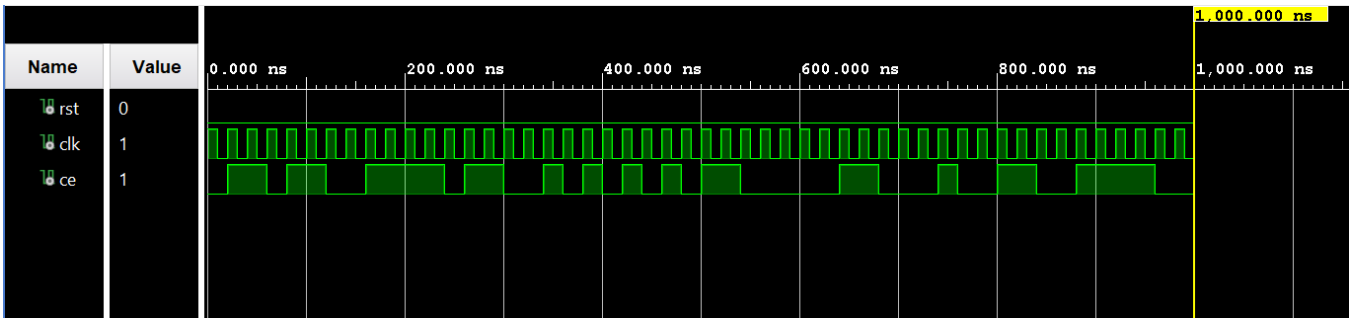

Figure 3: Completely Random CE Signal.

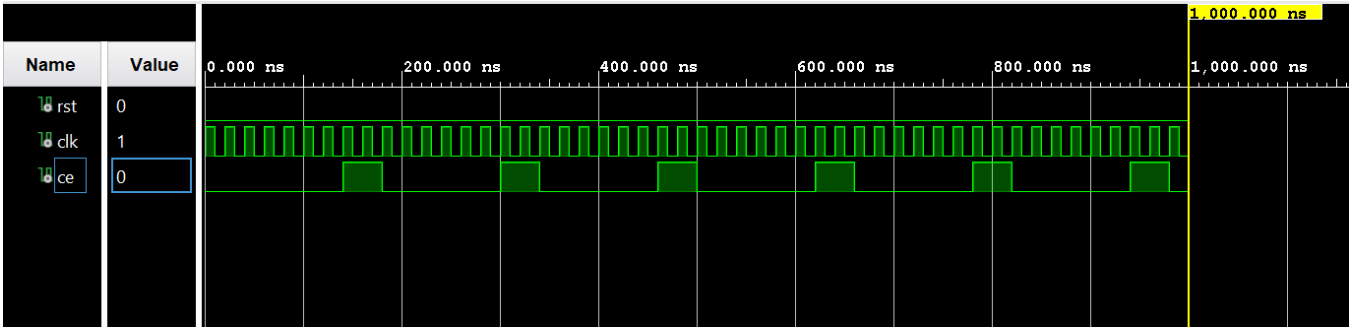

Figure 4: CE Signal with Constant Duty Cycle 1.

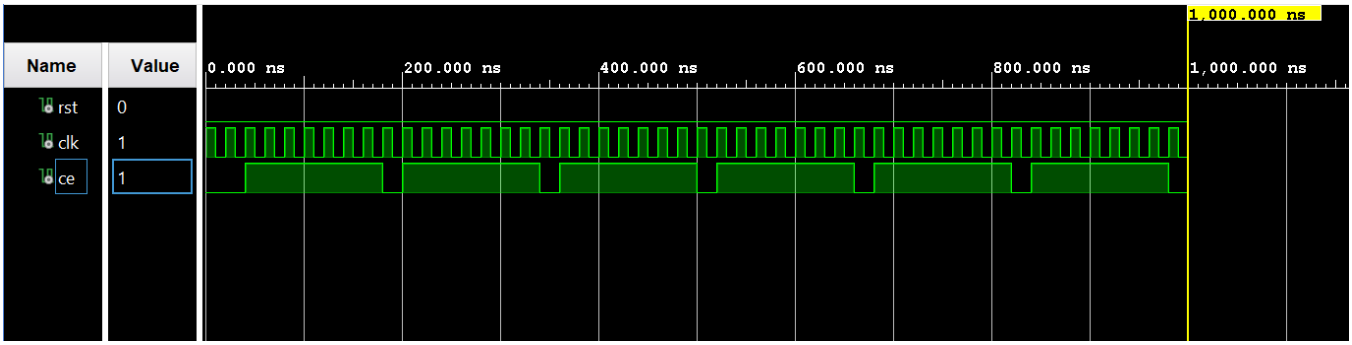

Figure 5: CE Signal with Constant Duty Cycle 2.

novel implementation of hiding in the time domain, which is easily applicable to every design with the clock enable input.

There are four finalists for encryption and key distribution and three finalists for digital signing regarding the NIST standardization, and as it is shown in Section 2, each of these cryptosystems has some vulnerabilities that could be exploited by either side-channel attack or fault attack. Fortunately, there are existing countermeasures for most of these attacks. The countermeasures are briefly discussed in Section 3. However, there are also attacks for which any countermeasure has not already been published, for example, the correlation power analysis of Saber. Thus, the countermeasures to these attacks should be the subject of future work in this field.

The current version of the countermeasure implementation described in Section 4 reduces the performance of the hidden algorithm by 50 % because the clock signal is disabled for 50 % of the time. To optimize this time ratio, the real hiding capabilities of the countermeasure will be examined by applying the discussed side-channel attacks on the hardware implementations of the post-quantum cryptosystems in future work.

ACKNOWLEDGMENTS

This work is supported by Ministry of the Interior of the Czech Republic under Grant VJ02010010.

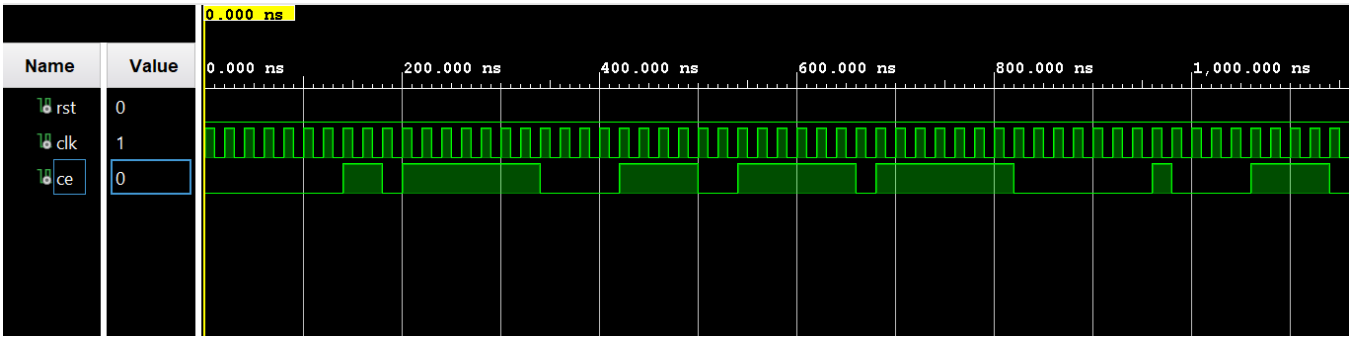

Figure 6: CE Signal with Variable Duty Cycle.

REFERENCES

[1] Abubakr Abdulgadir, Kamyar Mohajerani, Viet Ba Dang, Jens-Peter Kaps, and Kris Gaj. 2021. A Lightweight Implementation of Saber Resistant Against Side-Channel Attacks. In *International Conference on Cryptology in India*. Springer, 224–245.

[2] Martin R. Albrecht, Amit Deo, and Kenneth G. Paterson. 2018. Cold Boot Attacks on Ring and Module LWE Keys Under the NTT. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018, 3 (Aug. 2018), 173–213.

[3] Alessandro Barenghi, Luca Breveglieri, Israel Koren, and David Naccache. 2012. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proc. IEEE* 100, 11 (2012), 3056–3076.

[4] Pierre-Louis Cayrel and Pierre Dusart. 2010. McEliece/Niederreiter PKC: Sensitivity to Fault Injection. In *2010 5th International Conference on Future Information Technology*. 1–6.

[5] Rajat Subhra Chakraborty, Indrasish Saha, Ayan Palchaudhuri, and Gowtham Kumar Naik. 2013. Hardware Trojan insertion by direct modification of FPGA configuration bitstream. *IEEE Design & Test* 30, 2 (2013), 45–54.

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

[7] Cong Chen, Thomas Eisenbarth, Ingo von Maurich, and Rainer Steinwandt. 2016. Horizontal and Vertical Side Channel Analysis of a McEliece Cryptosystem. *IEEE Transactions on Information Forensics and Security* 11, 6 (2016), 1093–1105.

[8] Zhaohui Chen, Emre Karabulut, Aydin Aysu, Yuan Ma, and Jiwu Jing. 2021. An Efficient Non-Profiled Side-Channel Attack on the CRYSTALS-Dilithium Post-Quantum Signature. In *2021 IEEE 39th International Conference on Computer Design (ICCD)*. IEEE, 583–590.

[9] Jan-Pieter D'Anvers, Daniel Heinz, Peter Pessl, Michiel van Beirendonck, and Ingrid Verbauwhede. 2021. Higher-Order Masked Ciphertext Comparison for Lattice-Based Cryptography. *Cryptology ePrint Archive* (2021).

[10] Jintai Ding and Dieter Schmidt. 2005. Rainbow, a new multivariable polynomial signature scheme. In *International conference on applied cryptography and network security*. Springer, 164–175.

[11] Thomas Espitau, Pierre-Alain Fouque, Benoit Gerard, and Tibouchi. 2018. Loop-abort faults on lattice-based signature schemes and key exchange protocols. In *IEEE Transactions on Computers*. IEEE, 1535–1549.

[12] Junfeng Fan, Xu Guo, Elke De Mulder, Patrick Schaumont, Bart Preneel, and Ingrid Verbauwhede. 2010. State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. 76–87. https://doi.org/10.1109/HST.2010.5513110

[13] Junfeng Fan and Ingrid Verbauwhede. 2012. An updated survey on secure ECC implementations: Attacks, countermeasures and cost. In *Cryptography and Security: From Theory to Applications*. Springer, 265–282.

[14] Tim Fritzmann, Michiel Van Beirendonck, Debapriya Basu Roy, Patrick Karl, Thomas Schamberger, Ingrid Verbauwhede, and Georg Sigl. 2021. Masked accelerators and instruction set extensions for post-quantum cryptography. *Cryptology ePrint Archive* (2021).

[15] Kris Gaj. 2018. Challenges and rewards of implementing and benchmarking post-quantum cryptography in hardware. In *Proceedings of the 2018 on Great Lakes Symposium on VLSI*. 359–364.

[16] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. 2009. Lest we remember: cold-boot attacks on encryption. In *Commun. ACM*, Vol. 52. 91–98.

[17] Yasufumi Hashimoto, Tsuyoshi Takagi, and Kouichi Sakurai. 2011. General fault attacks on multivariate public key cryptosystems. In *International Workshop on Post-Quantum Cryptography*. Springer, 1–18.

[18] James Howe, Ayesha Khalid, Marco Martinoli, Francesco Regazzoni, and Elisabeth Oswald. 2019. Fault attack countermeasures for error samplers in lattice-based cryptography. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.

[19] James Howe, Ayesha Khalid, Marco Martinoli, Francesco Regazzoni, and Elisabeth Oswald. 2019. Fault Attack Countermeasures for Error Samplers in Lattice-Based Cryptography. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5.

[20] Arpan Jati, Naina Gupta, Anupam Chattopadhyay, and Somitra Kumar Sanadhya. 2021. A Configurable Crystals-Kyber Hardware Implementation with Side-Channel Protection. *Cryptology ePrint Archive* (2021).

[21] Tendayi Kamucheka, Michael Fahr, Tristen Teague, Alexander Nelson, David Andrews, and Miaoqing Huang. 2021. Power-based side channel attack analysis on PQC algorithms. *Cryptology ePrint Archive* (2021).

[22] Emre Karabulut and Aydin Aysu. 2021. Falcon Down: Breaking Falcon Post-Quantum Signature Scheme through Side-Channel Attacks. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 691–696.

[23] ChangKyun Kim, Martin Schläffer, and SangJae Moon. 2008. Differential side channel analysis attacks on FPGA implementations of ARIA. *ETRI journal* 30, 2 (2008), 315–325.

[24] Aviad Kipnis, Jacques Patarin, and Louis Goubin. 1999. Unbalanced oil and vinegar signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 206–222.

[25] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Annual international cryptology conference*. Springer, 388–397.

[26] Juliane Krämer and Mirjam Loiero. 2019. Fault attacks on UOV and rainbow. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 193–214.

[27] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. 2005. Successfully Attacking Masked AES Hardware Implementations. In *Cryptographic Hardware and Embedded Systems – CHES 2005*, Josyula R. Rao and Berk Sunar (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 157–171.

[28] Stefan Mangard and Kai Schramm. 2006. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In *Cryptographic Hardware and Embedded Systems - CHES 2006*, Louis Goubin and Mitsuru Matsui (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 76–90.

[29] Soundes Marzougui, Vincent Ulitzsch, Mehdi Tibouchi, and Jean-Pierre Seifert. 2022. Profiling Side-Channel Attacks on Dilithium: A Small Bit-Fiddling Leak Breaks It All. *Cryptology ePrint Archive* (2022).

[30] Seyedeh Sharareh Mirzargar and Mirjana Stojilović. 2019. Physical side-channel attacks and covert communication on FPGAs: A survey. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 202–210.

[31] Mouna Nakkar, Moustafa Mahmoud, and Amr Youssef. 2017. Fault analysis-resistant implementation of Rainbow Signature scheme. In *2017 29th International Conference on Microelectronics (ICM)*. IEEE, 1–5.

[32] Sıddıka Berna Örs, Elisabeth Oswald, and Bart Preneel. 2003. Power-analysis attacks on an FPGA–first experimental results. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 35–50.

[33] Aesun Park, Kyung-Ah Shim, Namhun Koo, and Dong-Guk Han. 2018. Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations:-rainbow and uov. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2018), 500–523.

[34] Kenneth G. Paterson and Ricardo Villanueva-Polanco. 2017. Cold Boot Attacks on NTRU. In *Progress in Cryptology – INDOCRYPT 2017*, Arpita Patra and Nigel P. Smart (Eds.). Springer International Publishing, 107–125.

[35] Eric Peeters, François-Xavier Standaert, Nicolas Donckers, and Jean-Jacques Quisquater. 2005. Improved higher-order side-channel attacks with FPGA experiments. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 309–323.

[36] Martin Petrvalsky, Tania Richmond, Milos Drutarovsky, Pierre-Louis Cayrel, and Viktor Fischer. 2016. Differential power analysis attack on the secure bit permutation in the McEliece cryptosystem. In *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*. 132–137.

[37] David Pokorný, Petr Socha, and Martin Novotný. 2021. Side-channel attack on Rainbow post-quantum signature. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 565–568.

[38] David Pokorný. 2021. *Analýza postranních kanálů postkvantového podpisu Rainbow*. Master's thesis. České vysoké učení technické v Praze.

[39] Markku-Juhani Saarinen. 2017. Arithmetic coding and blinding countermeasures for lattice signatures. *Journal of Cryptographic Engineering* (2017), 1–14.

[40] François-Xavier Standaert, Loïc van Oldeneel tot Oldenzeel, David Samyde, and Jean-Jacques Quisquater. 2003. Power analysis of FPGAs: How practical is the attack?. In *International Conference on Field Programmable Logic and Applications*. Springer, 701–710.

[41] O-X Standaert, Eric Peeters, Gaël Rouvroy, and J-J Quisquater. 2006. An overview of power analysis attacks against field programmable gate arrays. *Proc. IEEE* 94, 2 (2006), 383–394.

[42] Pawel Swierczynski, Marc Fyrbiak, Philipp Koppe, and Christof Paar. 2015. FPGA Trojans through detecting and weakening of cryptographic primitives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 8 (2015), 1236–1249.

[43] Ingo von Maurich and Tim Güneysu. 2014. Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*. 1–6.

[44] Thomas Wollinger, Jorge Guajardo, and Christof Paar. 2004. Security on FPGAs: State-of-the-art implementations and attacks. *ACM Transactions on Embedded Computing Systems (TECS)* 3, 3 (2004), 534–574.

[45] Jiliang Zhang and Gang Qu. 2019. Recent attacks and defenses on FPGA-based systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 12, 3 (2019), 1–24.