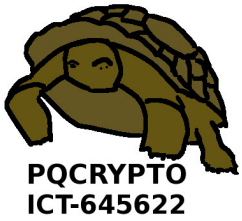

PQCRYPTO

Post-Quantum Cryptography for Long-Term Security

Project number: Horizon 2020 ICT-645622

Small Devices: D1.5 Intermediate Report on Physical Attacks

Due date of deliverable: 30. September 2016
Actual submission date: May 3, 2017

Start date of project: 1. March 2015 Duration: 3 years

Coordinator:
Technische Universiteit Eindhoven
Email: coordinator@pqcrypto.eu.org
www.pqcrypto.eu.org

Revision 1

Project co-funded by the European Commission within Horizon 2020		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

Small Devices: D1.5 Intermediate Report on Physical Attacks

Tim Güneysu, Tobias Oder

Contributors:

May 3, 2017
Revision 1

The work described in this report has in part been supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO.

Abstract

This document provides the PQCRYPTO project's intermediate report for post-quantum cryptographic algorithms that focus small devices. Algorithms are selected based on a level on confidence and their suitability for the constraints of small embedded devices.

Keywords: Post-quantum cryptography, small devices, hardware devices, microcontrollers, physical attacks, side-channel analysis

Contents

1 Introduction

PQCRYPTO recommends algorithms for encryption, digital signatures, and key exchange that are believed to withstand mathematical attacks even if those are carried out by quantum computers. However, implementations of these algorithms might still by vulnerable to physical attacks. Especially embedded applications are vulnerable to these kind of attacks since the attacker is in possession of the device which executes the algorithm. Such an adversary is free to not only control the input to the device and closely monitor the device, observing its physical properties whilst it performs the cryptographic operations. These physical variables, such as the timings required to perform computations, or the instantaneous power consumed during execution of the algorithm, may be sampled and recorded and used to derive intermediate values of the algorithms. In this report we examine possible attack vectors for implementations of post-quantum algorithms and discuss possible countermeasures to prevent these kind of attacks. We also give an overview of the state of the art in the field of side-channel analysis of post-quantum cryptography.

2 Physical Attacks

In this section, we discuss attacks exploiting physical properties of an implementation to gain knowledge of the secret key used in the executed algorithm. One distinguishes between passive attacks in which the attacker only monitors information, like execution time, power consumption, or electromagnetic radiation, and active attacks in which the attacker is allowed to interfere in the execution of the cipher.

When dealing with active attacks, one distinguishes between different levels of invasiveness. A non-invasive attacker is only allowed to modify the environment like the temperature, the voltage of the power supply, or the duration of clock cycles. These attacks usually aim to generate a faulty result that can be used to reveal the secret key. A semi-invasive attacker removes the package material of the device and introduces faults by shooting at the a specific location at the device with light or electromagnetic radiation. Invasive attackers aim to even alters the device itself and reverse-engineer the implementation.

Implementations are vulnerable to timing attacks if their execution time depends on secret data. Power analysis exploits the fact that in CMOS technology the dynamic power consumption is dominating in comparison to the static power consumption. An attacker executes the algorithm and measures the power consumption during the execution. The most important types of attacks on the power consumption leakage are simple power analysis (SPA) and differential power analysis (DPA).

2.1 Fault attacks

The idea of fault attacks is to induce a fault into a circuit and use the faulty output to get information about the secret key. This can be achieved by high temperature, unsupported supply voltage or current, excessively high overclocking, strong electric or magnetic fields, or even ionizing radiation. Fault attacks are usually non-invasive as the induced fault is only temporary and the device is not permanently damaged. The most prominent fault attack in cryptography was carried out by Boneh et al. [5]. Their attack on RSA-CRT signatures requires only one (arbitrary) faulty output and one correct output to break the scheme.

2.2 Timing attacks

When implementing cryptographic algorithms, the developer has to make sure that the execution time is independent of the secret data that is processed. Otherwise an attacker might be able to exploit the information about the execution time. Such attacks should not only be considered for embedded devices for which the attacker has physical access to, but also remote timing attacks are a threat that must be considered as shown by Brumley and Boneh [7]. Timing information can be leaked by conditional branches, instructions with non-constant execution time, and memory accesses that trigger cache hits or misses [2].

2.3 Simple power analysis

Simple power analysis [19] works similar to timing attacks. However, while timing attacks exploit the timing information of one or many executions of the algorithms, one or a few power traces of the executed algorithms are used to perform a simple power analysis. An attacker uses visual examination to identify leaking instructions whose execution depends on secret data. Thus, this attack is especially effective when the order of the executed instructions differs from run to run. For instance, an RSA implementation with a naive implementation of the square-and-multiply algorithm can easily be broken by SPA as the square operations and multiply operations are usually easily distinguishable in the power trace. Signal-processing techniques, like frequency filters, might improve the result and make the visual inspection easier.

2.4 Differential power analysis

While SPA targets the operation-dependency of the power consumption, DPA exploits its data-dependency. Introduced in 1998 by Kocher et al. [19], DPA (in contrast to SPA) needs many power traces and one analyzes the set of traces with statistical methods. When performing DPA an attacker does not attack the whole key at once, but only a part, e.g. one byte. A DPA is divided in an online phase and an offline phase. During the online phase, the attacker runs a vast amount of executions of the algorithm to be attacked with different inputs and measures the power consumption of the target device during each run. DPA requires a leakage model that is a prediction of the power consumption. Some leakage models are rather simple. For example, the Hamming weight model is based on the observation that the Hamming weight of a value that is stored in a register, influences the power consumption. During the offline phase, the attacker guesses the key byte and computes the intermediate value that he considers suitable to apply the power model to. Depending on the power model and the intermediate value, she assigns the corresponding power trace to one of two sets where one contains power traces with high predicted power consumption and one set contains traces with low prediction power consumption. For all power traces, the attacker stores the difference of the means of the sets. If the attack worked, the correct key guess has a much higher difference of means than the other guesses.

The Hamming weight leakage model is often applied when attacking software implementations. For hardware implementations a promising model is the Hamming distance model. The Hamming distance model assumes that the more bit positions of the input and output values of a circuit differ, the more switching operations happened within a circuit, and the higher is the power consumption. Other models that are more accurate require less traces to

successfully attack a target, but also need a deep knowledge of the implementation that is attacked.

2.4.1 t-test

A commonly used methodology for side-channel analysis is the t-test leakage detection method initially proposed in [15, 11]. For the non-specific *fixed vs. random* t-test one takes two types of measurements, one with fixed input and one with random input. The t-statistic t is computed as

$$t = \frac{\mu_F - \mu_R}{\sqrt{\frac{\sigma_F^2}{n_F} + \frac{\sigma_R^2}{n_R}}}$$

where μ_F, σ_F^2, and n_F (resp. μ_R, σ_R^2, and n_R) denote the mean, variance, and number of measurements set with fixed input (resp. random input). If the value exceeds the threshold $|t| > 4.5$, the test has detected leakage. As this test does not perform an actual attack and does not consider a certain power model it is called *non-specific*. Apart from the *fixed vs. random* t-test it is also possible to perform a *semi-fixed vs. random* t-test. Such a test does not fix the input but some intermediate values, e.g. part of the state of a block cipher to get a more accurate result.

3 Countermeasures

In this section we discuss different approach to prevent side-channel analysis. Note that there is not a single countermeasure that can be applied to fix all vulnerabilities, in practice usually a combination of countermeasures is applied.

3.1 Hiding

Hiding countermeasures are applied to raise the difficulty for an attacker to detect sensitive information in a set of power traces. This can be achieved by introducing additional noise or by trying to equalize the power consumption of all operations.

The first approach can be achieved by other computations that are executed in parallel or by shuffling the order of operations. For hardware implementations one can even instantiate dedicated noise generators to randomize the power consumption. If shuffling is applied an attacker needs to perform an extra alignment step before analyzing the power traces. Otherwise the number of required power traces drastically increases.

The second approach is more suitable for hardware implementations as in microcontrollers the developer has only limited influence on the power consumption of an instruction and only one instruction can be executed in parallel (except the microcontroller features SIMD instructions).

3.2 Masking

The idea behind masking is to split a secret value into several shares. The secret value can only be reconstructed with the knowledge of all shares. The splitting of the secret value can be performed in a Boolean way or in an arithmetic way. Boolean masking means that the XOR-sum of all shares results in the secret value and arithmetic masking means that the arithmetic

sum or difference of the shares results in the secret value. There are conversion approaches to switch between arithmetic and Boolean masking [12]. The major advantage of masking schemes is that they allow to prove the side-channel security of an algorithm. Nevertheless, there are still implementation challenges that have to be taken care of. Otherwise, a provably secure algorithm might still have a side-channel leakage. To achieve higher-order security, it is necessary to split the secret value into more shares.

3.3 Constant-time implementation

To prevent timing attacks and simple power analysis it is crucial to develop an implementation that has a constant (or at least secret-independent) execution time. Some pitfalls that should be avoided are:

- **Comparison of secret strings:** Such a comparison must not stop at the first unequal character.
- **Branches:** Branches must not be dependent on secret data. Ideally the same branches are taken for every run of the implementation.
- **Table look-ups:** On platforms with a cache, table look-ups can have varying access times. Thus the index must not depend on secret data for such platforms.
- **Compiler optimization:** A developer must take care that the compiler does not remove instructions that are critical for the security of the implementation but irrelevant for its functionality.

3.4 Fault countermeasures

The most intuitive way to detect a fault is to utilize redundant computations that are used to check the correctness of the result. Spatial redundancy is a possible countermeasure for hardware implementations and means the same operation is executed twice in parallel. This countermeasures has only a small performance overhead but the area consumption doubles. In contrast to that, temporal redundancy means executing another operation after the original operations has been finished. This can either be an additional decryption after an encryption operation to check whether the result matches the original plaintext or simply another encryption to compare both ciphertexts.

For fault attacks that must induce the fault at a specific point in time, it is also possible to randomize the order of the instructions to make an attack harder. Partial reconfiguration on FPGAs can also be used to randomize the location of the circuit that compute the operation. For linear operations error correcting codes can be used to detect faults.

4 Physical security of quantum-secure schemes

In this section, we review the state-of-the-art of research targeting the physical security of post-quantum cryptography. We are not aware of any work related physical attacks on multivariate quadratics and thus we do refrain from discussing these schemes. Similarly, there has not been done much research on side-channel analysis of hash-based primitives yet. However the underlying hash functions have been analyzed thoroughly in works like [3, 28, 30].

4.1 Code-based Cryptography

The McEliece encryption scheme [21] has been proposed in 1978 and belongs to the family of code-based cryptography. Much effort has been spent on analyzing the side-channel security of this scheme and developing suitable countermeasures. Simple power analysis of the scheme has been performed for FPGA implementations [22] and microcontroller implementations [18, 29]. These works also show that the attack can be made much harder by providing a timing- and instruction-invariant implementation.

Further more Chen et al. attacked McEliece FPGA implementations with DPA [8, 10]. The authors analyze the syndrome computation and are able to recover the complete secret key with an additional algebraic step that exploits the relation between the public and private key. As counteremasures to the presented attacks, Chen et al. also propose a masking scheme for McEliece in [9] and evaluate the side-channel security of their masked FPGA implementation. Their masking scheme is a hybrid of Boolean and arithmetic masking. The masks are generated on-the-fly using a pseudo-random number generator.

4.2 Lattice-based Cryptography

The lattice-based ring learning with errors (R-LWE) encryption scheme [20] has also been analyzed for its resistance against side-channel attacks in several works mainly focusing on DPA. The first approach to secure R-LWE against DPA has been proposed by Reparaz et al. in 2015 [27, 26]. The authors attempt to protect the secret key by splitting it into two shares and perform all operations separately on both shares. However, the last step of the algorithm is a decoding function that is not a linear operation and thus requires the knowledge of both shares. To solve this problem [27] proposed a masked decoder. As this decoder has a number of drawbacks, like being non-deterministic and increasing the failure rate of the scheme, Reparaz et al. [25] proposed another approach in 2016. In [25] not the secret key, but the ciphertext is split into two shares. This approach introduces a heavy computational overhead as it requires another run of the decryption during the encryption. Oder et al. [23] combined the ideas of [27] and [25] to avoid the aforementioned problems and also applied a CCA2-conversion to R-LWE to make it secure against adaptive chosen-ciphertext attackers. Furthermore the masking scheme from [23] has a proof to support its claim. Additionally, [27, 25, 23] all provide results of practical measurements to demonstrate that the masking schemes indeed prevent a leakage. Oder et al. also discuss the fault sensitvity of R-LWE in [23].

Lattice-based signatures schemes, like BLISS [13] and GLP [16], have also been analyzed for their vulnerability to fault attacks in [4] and [14]. Both papers consider instruction-skipping resulting in potential loop aborts and how to exploit such a fault. The work of Bindel et al. [4] furthermore examines the impact of zeroing or randomization of critical values. The proposed countermeasures mainly boil down to redundant computations that are used for correctness checks. Another proposed countermeasure is to prevent instruction-skipping is to deliberately induce a segmentation fault by allocating new memory for every intermediate result.

Bruinderink et al. [6] also found a cache-timing attack on the signature scheme BLISS. More specifically, they attacked the Gaussian sampler that is used to generate noise polynomials in BLISS and are able to extract the secret key with only 3,500 signatures. To prevent timing attacks many implementations of lattice-based schemes provide a constant or secret-independent execution time, like vectorized implementations of the GLP signature scheme and

the New Hope key exchange for Intel CPUs [1, 17]. Furthermore there are also microcontroller implementations of R-LWE that are protected against timing attacks [24, 23].

5 Conclusions

The physical security of post-quantum cryptography is essential for embedded applications of these algorithms. The McEliece cryptosystem has already been well researched, however there are still open research questions, like higher-order attacks. First works on side-channel analysis and countermeasures for lattice-based cryptosystems also exist. Nevertheless, there is a great variety of lattice-based cryptosystems and so far most work has focused on R-LWE and signature schemes only. Hash-based signature schemes, like XMSS and SPHINCS and MQ-based schemes are also interesting to look at from a side-channel perspective.

References

[1] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange – a new hope. In *Proceedings of the 25th USENIX Security Symposium*. USENIX, (to appear). Document ID: 0462d84a3d34b12b75e8f5e4ca032869, http://cryptojedi.org/papers/#newhope.

[2] Daniel J Bernstein. Cache-timing attacks on aes, 2005.

[3] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Building power analysis resistant implementations of keccak. In *Second SHA-3 candidate conference*, volume 142. Citeseer, 2010.

[4] Nina Bindel, Johannes Buchmann, and Juliane Krämer. Lattice-based signature schemes and their sensitivity to fault attacks. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2016 Workshop on*, pages 63–77. IEEE, 2016.

[5] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.

[6] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload–a cache attack on the bliss lattice-based signature scheme. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 323–345. Springer, 2016.

[7] David Brumley and Dan Boneh. Remote timing attacks are practical. In *Proceedings of the 12th USENIX Security Symposium, Washington, D.C., USA, August 4-8, 2003*. USENIX Association, 2003.

[8] Cong Chen, Thomas Eisenbarth, Ingo von Maurich, and Rainer Steinwandt. Differential power analysis of a mceliece cryptosystem. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June*

2-5, 2015, Revised Selected Papers, volume 9092 of *Lecture Notes in Computer Science*, pages 538–556. Springer, 2015.

[9] Cong Chen, Thomas Eisenbarth, Ingo von Maurich, and Rainer Steinwandt. Masking large keys in hardware: A masked implementation of mceliece. In *International Conference on Selected Areas in Cryptography*, pages 293–309. Springer, 2015.

[10] Cong Chen, Thomas Eisenbarth, Ingo von Maurich, and Rainer Steinwandt. Horizontal and vertical side channel analysis of a mceliece cryptosystem. *IEEE Trans. Information Forensics and Security*, 11(6):1093–1105, 2016.

[11] Jeremy Cooper, Elke Demulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test Vector Leakage Assessment (TVLA) Methodology in Practice. International Cryptographic Module Conference, 2013.

[12] Jean-Sébastien Coron, Johann Großschädl, Mehdi Tibouchi, and Praveen Kumar Vadnala. Conversion from arithmetic to boolean masking with logarithmic complexity. In *International Workshop on Fast Software Encryption*, pages 130–149. Springer, 2015.

[13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology–CRYPTO 2013*, pages 40–56. Springer, 2013.

[14] Thomas Espitau, Pierre-Alain Fouque, BenoÃőt GÃľrard, and Mehdi Tibouchi. Loop-abort faults on lattice-based fiatâĂŞshamir and hash-and-sign signatures. Cryptology ePrint Archive, Report 2016/449, 2016. http://eprint.iacr.org/2016/449.

[15] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi. A testing methodology for side channel resistance validation. In *NIST non-invasive attack testing workshop*, 2011.

[16] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[17] Tim Güneysu, Tobias Oder, Thomas Pöppelmann, and Peter Schwabe. Software speed records for lattice-based signatures. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, volume 7932 of *Lecture Notes in Computer Science*, pages 67–82. Springer, 2013.

[18] Stefan Heyse, Amir Moradi, and Christof Paar. Practical power analysis attacks on software implementations of mceliece. In *International Workshop on Post-Quantum Cryptography*, pages 108–125. Springer, 2010.

[19] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1):5–27, 2011.

[20] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.

[21] Robert J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, 1978.

[22] H. Gregor Molter, Marc Stöttinger, Abdulhadi Shoufan, and Falko Strenzke. A simple power analysis attack on a mceliece cryptoprocessor. *Journal of Cryptographic Engineering*, 1(1):29–36, 2011.

[23] Tobias Oder, Tobias Schneider, Thomas Pöppelmann, and Tim Güneysu. Practical cca2-secure and masked ring-lwe implementation. *IACR Cryptology ePrint Archive*, 2016:1109, 2016.

[24] Thomas Pöppelmann, Tobias Oder, and Tim Güneysu. High-performance ideal lattice-based cryptography on 8-bit ATxmega microcontrollers. In Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors, *Progress in Cryptology - LATINCRYPT 2015 - 4th International Conference on Cryptology and Information Security in Latin America, Guadalajara, Mexico, August 23-26, 2015, Proceedings*, volume 9230 of *Lecture Notes in Computer Science*, pages 346–365. Springer, 2015.

[25] Oscar Reparaz, Ruan de Clercq, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. Additively homomorphic ring-lwe masking. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, volume 9606 of *Lecture Notes in Computer Science*, pages 233–244. Springer, 2016.

[26] Oscar Reparaz, Sujoy Sinha Roy, Ruan de Clercq, Frederik Vercauteren, and Ingrid Verbauwhede. Masking ring-lwe. *J. Cryptographic Engineering*, 6(2):139–153, 2016.

[27] Oscar Reparaz, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. A masked ring-lwe implementation. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 683–702. Springer, 2015.

[28] Mostafa Taha and Patrick Schaumont. Side-channel analysis of mac-keccak. In *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*, pages 125–130. IEEE, 2013.

[29] Ingo Von Maurich and Tim Güneysu. Towards side-channel resistant implementations of qc-mdpc mceliece encryption on constrained devices. *PQCrypto*, 2014:266–282, 2014.

[30] Michael Zohner, Michael Kasper, Marc Stöttinger, and Sorin A Huss. Side channel analysis of the sha-3 finalists. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1012–1017. EDA Consortium, 2012.