Implementation Attacks on Post-Quantum Cryptographic Schemes

Mostafa Taha and Thomas Eisenbarth

Worcester Polytechnic Institute,
Worcester, MA 01609, USA
Email: mtaha@aun.edu.eg, teisenbarth@wpi.edu

Abstract. Post-quantum cryptographic schemes have been developed in the last decade in response to the rise of quantum computers. Fortunately, several schemes have been developed with quantum resistance. However, there is very little effort in evaluating and comparing these schemes in the embedded settings. Low cost embedded devices represents a highly-constraint environment that challenges all post-quantum cryptographic schemes. Moreover, there are even fewer efforts in evaluating the security of these schemes against implementation attacks including side-channel and fault attacks. It is commonly accepted that, any embedded cryptographic module that is built without a sound countermeasure, can be easily broken. Therefore, we investigate the question: *Are we ready to implement post-quantum cryptographic schemes on embedded systems?* We present an exhaustive survey of research efforts in designing embedded modules of post-quantum cryptographic schemes and the efforts in securing these modules against implementation attacks. Unfortunately, the study shows that: we are not ready yet to implement any post-quantum cryptographic scheme in practical embedded systems. There is still a considerable amount of research that needs to be conducted before reaching a satisfactory level of security.

1 Introduction

Quantum computers are a special class of computing machines that can harness properties from quantum mechanics into computing power. The quantum properties of concern include for example, superposition. Superposition accepts the state of a bit in quantum computers (aka qbit) not only as a 0 or 1 (as classical computers) but also as any superposition of them. Superposition was first proposed as a tool to explain the wave-particle duality of electrons. Quantum computing is a new computing paradigm that is *not intended* to replace classical computing, but it can achieve a drastic speed-up for some applications beyond what is possible with classical computing. For example, quantum computers are inherently parallel and can achieve quadratic speed-up in search algorithms [22]. Although a full-fledged quantum computer has not been designed yet, there are many on-going projects that are competing to achieve this goal (e.g. [1]).

This new computing paradigm will enable novel algorithms that were never possible in classical computing systems. One of these algorithms is the Shor's algorithm, which can solve the integer factorization problem in polynomial time, i.e. it can break most of the current public-key cryptographic schemes, jeopardizing the public key infrastructure that the web infrastructure relies on. In other words, once a reasonably-sized quantum computer is demonstrated, all security protocols must be updated to phase-out RSA, DSA, ECC and more. In response, the cryptographic community developed new public-key cryptographic schemes with quantum resistance. The new schemes depend on other hard problems (as discussed in the remaining sections) that are believed to remain difficult even in the presence of quantum computers. Although the inherent parallelism of quantum computers will reduce the computation-time of brute force attacks, this threat can be alleviated by increasing the security level of the cryptographic scheme (e.g. from 128 bits to 256 bits).

However, these new cryptographic schemes cannot be implemented in embedded systems unless they are protected against implementation attacks. Implementation attacks refers to the ability of an attacker to break a cryptographic module by exploiting vulnerabilities in the underlying implementation (e.g. side-channel attacks), rather than the mathematical structure. For example, side-channel attacks harvest information leaked through power, electromagnetic emanation, or execution time to break the cryptographic scheme [35]. Also, the attacker can maliciously inject faults into the cryptographic module and study the faulty output, which can reveal some information about the secret key (aka fault attacks) [7]. It is accepted in the cryptographic community that, any implementation that is not protected against implementation attacks can easily be broken. Hence, the current critical question is: *Are we ready to implement post-quantum cryptographic schemes on embedded systems?*

In this paper, we try to answer this question. In Section 2, we start with a brief introduction to side-channel analysis and fault attacks. Then, we study the different categories of post-quantum cryptography namely, code-based, hash-based , lattice-based, and Multivariate cryptography. In each category, we briefly review the cryptographic hard problem used and a sample scenario for its usage. Also, we review the stable cryptographic algorithms in each category. Then, we study the current implementation attacks and the efforts in designing secure implementation. We conclude each section with the remaining open research problems. Finally, Section 7 gives the overall conclusion. Unfortunately, the study shows that: we are not ready yet to implement any post-quantum cryptographic schemes in practical embedded systems. There is still a considerable amount of research that needs to be conduct before reaching a satisfactory level of security.

2 Background

Classical cryptography protects the secret key from adversaries with access to the input (plaintext) and/or the output (ciphertext) of an encryption process. On the other hand, implementation attacks enable the adversary to acquire extra knowledge about the secret key by harvesting information leaked from the cryptographic module itself during operation. In this section, we will review the most practical implementation attacks, side-channel analysis and fault attacks.

2.1 Side-Channel Analysis

Side-channel analysis (SCA) is a passive attack where the adversary can harvest information from power consumption, electromagnetic radiation, execution time, and more. The instantaneous value of each of these sources depends on the data being processed within the module, which in-turn depends on the secret key. SCA is a major security threat because power leakage (for example) gives localized information about every single operation within the cryptographic algorithm, hence the attacker can target the first or last operation to focus exclusively on small parts of the secret key (sub-key). Therefore, increasing the security level (size of the key) of a cryptographic scheme does not prevent SCA because each sub-key can be recovered individually.

SCA can be mounted using only one leakage trace (or the average of many leakage traces collected with the *same* inputs), which is known as Simple Power Analysis (SPA). Protection against SPA attacks can be done by preventing data-dependent branches and increasing the noise level. SCA can also be mounted by combining information from many leakage traces collected at *different* inputs, which is known as Differential Power Analysis (DPA). DPA, which is the more serious threat, works in the following steps:

1. Select a sensitive variable within the algorithm. A sensitive variable is an intermediate variable within the cryptographic algorithm that depends on both the key (typically only 8 bits or less) and known inputs or outputs (plaintext or ciphertext).
2. Collect the leakage observation (power consumption) of the cryptographic module while processing many different inputs.
3. At the same inputs, compute values of the sensitive variable assuming all key hypotheses.
4. Convert values into hypothetical power consumption using a leakage model.
5. Compare the hypothetical power consumption to the actual one searching for the key that leads to the best match.

The leakage (power) model can be approximated by simplifying models such as Hamming weight of the value, which is suitable for processors, or the Hamming distance between two consecutive register states, which is suitable for hardware modules. *Profiling attacks* improve over this approach by accurately estimating the leakage model from a cloned device. Also, the power model can be extracted on-the-fly from the collected traces themselves, as done e.g. in collision attacks [41]. The comparison done in the final step can be done with correlation coefficient, mutual information, or comparison of statistical moments. A cryptographic module is vulnerable to DPA attacks if the following conditions are met:

1. Attacker can predict the value of the sensitive variable.

2. The value of the sensitive variable affects the leakage in a somewhat predictable way.
3. Information from several observations can be combined.

Countermeasures against DPA attacks fall into three categories. Each category breaks one of the aforementioned conditions as follows:

1. *Masking:* A random variable can be generated on-board to randomize the input and all the computations. Randomness should be removed only at the end of computation. This should prevent the attacker from correctly predicting the intermediate variables.
2. *Hiding:* A redundant module can be designed on-board to process the complement of the input, so that the system power consumption is kept constant. Also, the order of program execution can be randomized where there is no data dependency (shuffling). A third method is to design an on-board noise generator. The goal of these tools is to minimize the signal-to-noise ratio within the trace, which limits the effect of intermediate variables on power leakage.
3. *Leakage Resiliency:* A key-updating mechanism can be used to limit the life-time of any key to only few cryptographic operations.

2.2 Fault Analysis

Fault analysis is an active implementation attack. In case of a computation fault, the output will be wrong and in some way, it should also be random. Unfortunately, faults can be maliciously injected into specific operations so that, the faulty output carries some information about the secret key. Faults injection can be targeted in the spatial dimension (a selected location) using a laser or a focused ion beam. Also, faults can be targeted in the temporal dimension (clock-cycle) by reducing the supply voltage and/or increasing the operating frequency.

There are two categories of fault analysis attacks. First, the faulty output(s) can be analyzed against the correct one with mathematical techniques for cryptanalysis. Typically, there is only one key that can change the output to follow the monitored behavior. Conditions for a successful fault attack within this category depend on the cryptographic algorithm itself. The second category of fault attacks is*fault sensitivity analysis* [34]. Here, the operating frequency (for example) can be gradually increased to the point where fault occurs. The exact point where fault occurs depend on the data-being processed. Hence, the techniques of this category are very similar to DPA attacks. Also, fault sensitivity analysis does not require the exact value of the faulty output. Knowledge that a fault occurred is sufficient to mount such attacks.

Protection against fault attacks follows two methodologies: prevention and detection. A cryptographic module can be designed with inherent fault prevention techniques similar to embedded systems in highly-faulty environments (e.g. space shuttles). Also, cryptographic module can be enforced with fault detection mechanisms, e.g. computing the same result twice and compare them. Once a fault is detected, the module can deliver a random output that is not related to the data being processed, or protect the key by other means.

The remaining sections summarize the research efforts in analyzing post-quantum cryptographic schemes against implementation attacks. We also provide an overview of all the cryptographic modules that are designed with protection against these attacks.

3 Code-based Cryptography

Code-based cryptographic systems depend on the hardness of correcting a codeword following a random error-correcting code. The concept is that, the secret-key owner (Alice) selects an error-correcting code and publishes a random-looking version of that code as her public-key. In one realization, messages are treated as codewords, multiplied by the public-key, and augmented by random noise, so that only the secret key holder knowing the correct code can recover the correct codeword. This concept is used by The McEliece encryption scheme [36]. Alternatively, the codewords can be chosen at random, and the message can be added as noise, as done in the Niederreiter PKC [42].

The first code-based public-key cryptosystem was proposed my McEliece [36] using binary Goppa code. Thereafter, almost every error-correcting code has been tested in this cryptographic setting and, unfortunately, they all failed to achieve sound cryptographic properties except the original proposal of Goppa codes and the Moderate-Density Parity-Check codes (MDPC) [39]. The core idea of using error-correcting codes as a trapdoor one-way function has been further developed into other public key encryption schemes [42], signature schemes [12], identification schemes [50, 58], random number generators [19], as well as stream ciphers [20] and hash functions [3].

Code-based cryptosystems are not currently popular due to the large size of their public keys, e.g. 32,768 bits using Goppa codes [38] or 9,857 bits using MDPC [39], compared to 3072 bits for RSA or 256 bits for ECC, all at 128-bits of security. However, once the cryptographic structure of RSA/ECC breaks due to the rise of quantum computers, Code-based cryptosystems will be an interesting alternative.

3.1 Current Attacks

Side-channel analysis can be used to recover the secret key by monitoring the decryption (aka decoding) process done by Alice, as it involves the use of the original, un-randomized version of the code. In this section, we review available implementation attacks against McEliece cryptosystem.

SCA of decoding of Goppa codes The most widely used decoding scheme for Goppa codes is the Patterson algorithm [44]. The first timing attack against the PC implementation of Patterson algorithm was reported in [54]. This attack targeted computing the error locator polynomial as part of the Patterson decoding algorithm hence, they could successfully reveal the encrypted message but not the secret key. The attack was improved in [4, 47] and tested against FPGA platforms. Also, the attack was extended to target the homomorphic properties of other cryptosystems that has similar structure to McEliece [52]. Further analysis of the Patterson decoding algorithm lead to more serious attacks that can recover the secret key of a PC implementation [51, 53].

The first power attack against McEliece cryptosystems was proposed in [24]. Power attacks could only be investigated after implementing McEliece on a low-cost microcontroller as the one reported in [16]. This attack could completely recover the secret key. As a countermeasure, they suggested to use a mix of shuffling and boolean masking as a countermeasure, however they did not propose a complete solution. Later, another power attack was proposed in [40] with fewer assumptions on the underlying implementation. However, the attack can only reveal the encrypted message but not the secret key. This attack was successfully tested against an FPGA implementation [48].

In fact, demonstrating a successful power attack against the Patterson decoding algorithm shows that, all the previous modules that are protected against time attacks are still vulnerable to power attack. The reason is that, the information leakage through time and power are orthogonal to each other. Protecting against one attack does not guarantee protection against the other because, even if the execution time is fixed, the power consumption may still depend on sensitive information.

SCA of the decoding of MDPC codes Very recently, MDPC codes was proposed as a promising candidate to replace Goppa codes for using public-key of a smaller size (especially in the quasi-cyclic form) [39]. Decoding of MDPC codes is typically done by variants of the bit-flipping algorithms [21]. A lightweight implementation of McEliece with MDPC codes on a low-cost microcontroller was proposed in [25]. This implementation was targeted by simple power and timing attacks, and hence a better implementation was proposed with equalized execution time [61]. Then, an FPGA implementation with improved performance was proposed in [60]. Again, this realization was attacked with differential power analysis by harvesting information from many different traces against the same secret key in [11]. In fact, this attack was the first differential analysis to be reported against any McEliece implementation.

Fault attacks In general, code-based cryptographic cryptosystems are heavily resistant against fault attacks due to the inherent error correction capability. It was shown that, decoding may lead to a wrong message, but generally, no secret information is revealed [10]. However, this does not ensure that faulting the control part of an implementation might not result in information loss. To date, code-based cryptographic cryptosystems have not been tested against fault sensitivity attacks.

3.2 Open Research Problems

Table 1 shows an exhaustive summary, to the best of our knowledge, of the current studies. Although a considerable amount of research has investigated side-channel properties of the McEliece cryptosystems, the research in this direction is still in its infancy. Please note that, any embedded product needs to be protected against timing, power *and* fault attacks before being deployed.

Table 1. Available studies in the implementation security of McEliece

Code	**Time**	**Power**	**Faults**
Goppa	PC [4, 51, 52, 53, 54]	μ C [24]	[10]
	FPGA [47]	FPGA [40]	
MDPC	μ C [61]	FPGA [11]	

4 Hash-based Cryptography

Hash-based digital signature schemes have been studied since 1979. The original proposal was for hash-based one-time signatures by Lamport [33], and the subsequent extension to many-time signatures known as Merkle Hash-trees [37]. These schemes have security proofs that usually rely on the collision resistance of the underlying hash function. Hence, by choosing an appropriate hash function, Hash-based signature schemes become a natural candidate for a world with quantum computers.

Hash-based signatures can perform well in terms of computation time. Implementations of hash-based signatures have been proposed in detail, for PCs as well as embedded processors [5, 17, 28]. All designs rely heavily on the performance of the underlying hash function.

4.1 Current Attacks and Open Research Problems

The probably biggest drawback of hash-based signatures is the rather high signature length. A common optimization to reduce signature length is the Winternitz optimization [15], with further improvements in [27]. Other drawbacks, such as limited number of signatures have been addressed by several works [9,28], as has the stateful design, just recently, in [5].

There is only very limited work with respect to side-channel analysis. One exception is the study in [17], which claims strong leakage resiliency to DPA-like attacks, since individual one-time signature keys are ideally used only once. In practice, they are used slightly more frequently, the number is nevertheless limited, giving these signatures a huge advantage over prevailing signature schemes. No work on fault attacks or SPA has been performed that is specific to Hash-based schemes. However, since the security of the schemes always depends of that of the underlying hash function, any implementation attack on such an implementation might have impact of the signature scheme invoking it. Many hashing functions have been studied against implementation attack. For instance, the next standard for secure hashing functions SHA-3 has been investigated in both embedded software [55], and hardware [45]. Embedded implementation of SHA-3 supported with SCA-countermeasure following the masking principle was proposed in [6]. Another protected implementation that follows the leakage resiliency princible was proposed in [56].

To date, there are no hash-based public key encryption schemes. This has been an open research problem for a long time.

5 Lattice-based Cryptography

Lattice, as used in cryptography, is the group of points that can be expressed as integer coefficients of a set of basis vectors. A 2-D lattice can be geometrically expressed as a tile of any regular shape. Although a lattice is fully defined by its basis vectors, there are an infinite number of different basis vectors that can express the same lattice. Hence, given the basis vectors of a lattice, it is hard for an attacker to find the shortest (non-zero) vector in that lattice (i.e. the closest point to the origin of the lattice). This hard problem is known as the Shortest Vector Problem (SVP). There are many other related hard problems in lattices including the Closest Vector Problem (CVP), the Short Integer Solution (SIS) Problem and the Learning With Errors (LWE) Problem. Lattices have been developed extensively in the last decade with complete proposals in public-key encryption, digital signatures, and fully homomorphic encryption. The most promising samples of lattice-based cryptography are NTRU [26] and variants of the LWE [46].

5.1 Current Attacks and Open Research Problems

Table 2 shows an exhaustive summary, to the best of our knowledge, of the current research efforts in the implementation security of lattice-based cryptographic schemes. Power attacks against the NTRU cryptosystems have been demonstrated in [2] and [64]. Similar study in the context of body-area networks was proposed in [62]. Some ideas for designing a protected module were discussed in [62,64] however, with very limited details. Also, timing attacks against NTRU was proposed in [49] and [59]. Fault attacks against the NTRU cryptosystems working in encryption and digital signatures were proposed in [31] and [29]. Countermeasures against fault attacks were proposed in [30].

NTRU is the only lattice-based cryptographic scheme that has been studied against implementation attacks. Still, to the best of our knowledge, there is no NTRU embedded design that has been proposed with adequate protection against differential power attacks.

Table 2. Available studies in the implementation security of NTRU

Code	**Time**	**Power**	**Faults**
NTRU	[49, 59]	[2, 62, 64]	[29, 30, 31]

6 Multivariate Cryptography

Multivariate cryptosystems depend on the hardness of solving a system of non-linear (typically, quadratic) equations over a finite field. The concept is that, the secret-key owner (Alice) selects a quadratic map $\mathcal{F}$, which can be easily and computationally efficient inverted. Then, she publishes a random-looking version of this map after multiplying with two invertible affine transformations $\mathcal{T}$ and $\mathcal{S}$, so that the public key is $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{S}$.

As a public-key scheme, whoever wants to send Alice a message (say Bob), he represent the message as a vector x and computes the ciphertext $y = \mathcal{P}(x)$. To decrypt the message, Alice needs to solve the system of non-linear equations:

$$\mathcal{P}(x') = y$$

to find the message x'. Alice can easily solve this system for owning the three ingredient of $\mathcal{P}$, where each of the ingredients is invertible. This can be done by computing $y_1 = \mathcal{T}(y)$, $y_2 = \mathcal{F}(y_1)$ then $x' = \mathcal{S}(y_2)$. However it is computationally hard for anyone else (Eve) to solve the system of equations as there is no efficient code to invert $\mathcal{P}$.

Unfortunately, it turns out that it is difficult to hide the private ingredients while publishing the public key. Indeed, most public-key schemes built with multivariate quadratic polynomials are currently broken (e.g. [8, 18]). However, Multivariate cryptosystems are still an important class of post-quantum cryptography as it can work very efficiently in signature schemes. To generate a signature for message y, Alice, the secret-key owner, needs to find any solution x that satisfies the equation ($\mathcal{P}(x) = y$). Whoever needs to verify the signature, Bob, he computes $\mathcal{P}(x)$ and checks if the result matches y or not. Here, the mapping between x and y is *not necessarily* bijective (one to one mapping) as the encryption case. The same message y can have many valid signatures (or solutions to the quadratic equations), hence we can drop or randomize some coefficients of the map $\mathcal{P}$ without affecting functionality. This added feature allowed the design of several secure multivariate signature schemes (e.g. Unbalanced Oil and Vinegar (UOV) [32], Rainbow [14] and Tame Transformation Signatures (TTS) [63]). These signature scheme were practically implemented on a low-cost 8-bit microcontroller [13, 57].

6.1 Current Attacks and Open Research Problems

It is fairly reasonable to say that, the research on the implementation security of multivariate cryptographic schemes has not started yet. The reason is that, SCA evaluates only practical instances, which are built only after gaining some confidence in the mathematical security of a given scheme, which did not happen for most multivariate schemes. To the best of our knowledge, there are only two studies in this direction. First, Okeya *et al.* proposed side-channel attack on Sash [43], an encryption scheme that is currently broken. Also, Hashimoto *et al.* proposed fault attacks against several signature schemes [23]. To date, there is no study that address the power analysis of any multivariate cryptosystem.

As a future research direction, practical multivariate cryptosystem (e.g. [13, 57]) can be investigated against simple and differential power attacks. Also, secure instantiations need to be designed before using multivariate cryptosystem in any embedded product.

7 Conclusion

This work presents an exhaustive survey of the research efforts in designing post-quantum cryptographic schemes on embedded systems with resistance against implementation attacks. Our study shows that, there are only very limited explorations of the implementation space, especially for embedded systems in software and hardware, for most of the post-quantum schemes. Consequently, the impact of countermeasures to such implementations is even less studied.

With respect to implementation attacks, Code based cryptography has received the majority of analysis so far, followed by lattice-based cryptography. Hash-based cryptography, while available and stable for a long time, has seen little effort with respect to implementation security, as has Multivariate cryptography. In many cases, these studies have shown that post-quantum schemes behave quite differently to the well studied and currently used public key schemes. Hence, we believe that the state of the art motivates deeper efforts, both in terms of further vulnerability analysis as well as deeper study of countermeasures.

References

1. NASA's quantum artificial intelligence laboratory.
2. A. Atici, L. Batina, B. Gierlichs, and I. Verbauwhede. Power analysis on NTRU implementations for RFIDs: First results. 2008.
3. D. Augot, M. Finiasz, and N. Sendrier. A family of fast syndrome based cryptographic hash functions. In *Progress in Cryptology Mycrypt 2005*, volume 3715 of *Springer LNCS*, pages 64–83. 2005.

4. R. Avanzi, S. Hoerder, D. Page, and M. Tunstall. Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems. *Journal of Cryptographic Engineering*, 1(4):271–281, 2011.
5. D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, P. Schwabe, and Z. Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. Tech Report, October 2014.
6. B. Bilgin, J. Daemen, V. Nikov, S. Nikova, V. Rijmen, and G. Van Assche. Efficient and first-order DPA resistant implementations of keccak. In *CARDIS 2013*, 2013.
7. D. Boneh, R. DeMillo, and R. Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology EUROCRYPT 97*, volume 1233 of *Springer LNCS*, pages 37–51. 1997.
8. C. Bouillaguet, P.-A. Fouque, and G. Macario-Rat. Practical key-recovery for all possible parameters of SFLASH. In *Advances in Cryptology ASIACRYPT 2011*, volume 7073 of *Springer LNCS*, pages 667–685. 2011.
9. J. Buchmann, E. Dahmen, E. Klintsevich, K. Okeya, and C. Vuillaume. Merkle signatures with virtually unlimited signature capacity. In *Applied Cryptography and Network Security*, volume 4521 of *Springer LNCS*, pages 31–45. 2007.
10. P. Cayrel and P. Dusart. McEliece/Niederreiter PKC: Sensitivity to fault injection. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pages 1–6, May 2010.
11. C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt. Differential power analysis of a mceliece cryptosystem. Cryptology ePrint Archive, Report 2014/534, 2014. `http://eprint.iacr.org/`.
12. N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology ASIACRYPT 2001*, volume 2248 of *Springer LNCS*, pages 157–174. 2001.
13. P. Czypek, S. Heyse, and E. Thomae. Efficient implementations of MQPKS on constrained devices. In *Cryptographic Hardware and Embedded Systems, CHES 2012*, volume 7428 of *Springer LNCS*, pages 374–389. 2012.
14. J. Ding and D. Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *Applied Cryptography and Network Security*, volume 3531 of *Springer LNCS*, pages 164–175. 2005.
15. C. Dods, N. P. Smart, and M. Stam. Hash Based Digital Signature Schemes. In *Cryptography and Coding*, pages 96–115, 2005.
16. T. Eisenbarth, T. Güneysu, S. Heyse, and C. Paar. MicroEliece: McEliece for embedded devices. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Springer LNCS*, pages 49–64. 2009.
17. T. Eisenbarth, I. von Maurich, and X. Ye. Faster hash-based signatures with bounded leakage. In *Selected Areas in Cryptography – SAC 2013*, Springer LNCS, pages 223–243. 2014.
18. J.-C. Faugre and A. Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using grbner bases. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Springer LNCS*, pages 44–60. 2003.
19. J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Advances in Cryptology, EUROCRYPT 96*, volume 1070 of *Springer LNCS*, pages 245–255. 1996.
20. P. Gaborit, C. Lauradoux, and N. Sendrier. Synd: a fast code-based stream cipher with a security reduction. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 186–190, June 2007.
21. R. Gallager. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1):21–28, January 1962.

22. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.
23. Y. Hashimoto, T. Takagi, and K. Sakurai. General fault attacks on multivariate public key cryptosystems. In *Post-Quantum Cryptography*, volume 7071 of *Springer LNCS*, pages 1–18. 2011.
24. S. Heyse, A. Moradi, and C. Paar. Practical power analysis attacks on software implementations of McEliece. In *Post-Quantum Cryptography*, volume 6061 of *Springer LNCS*, pages 108–125. 2010.
25. S. Heyse, I. von Maurich, and T. Güneysu. Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices. In *Cryptographic Hardware and Embedded Systems - CHES 2013*, volume 8086 of *Springer LNCS*, pages 273–292. 2013.
26. J. Hoffstein, J. Pipher, and J. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*, volume 1423 of *Springer LNCS*, pages 267–288. 1998.
27. A. Hülsing. W-OTS+ - Shorter Signatures for Hash-Based Signature Schemes. In *AFRICACRYPT*, pages 173–188, 2013.
28. A. Hülsing, L. Rausch, and J. Buchmann. Optimal parameters for xmss mt. In *Security Engineering and Intelligence Informatics*, volume 8128, pages 194–208. Springer LNCS, 2013.
29. A. Kamal and A. Youssef. Fault analysis of the NTRUSign digital signature scheme. *Cryptography and Communications*, 4(2):131–144, 2012.
30. A. Kamal and A. Youssef. Strengthening hardware implementations of NTRUEncrypt against fault analysis attacks. *Journal of Cryptographic Engineering*, 3(4):227–240, 2013.
31. A. A. Kamal and A. Youssef. Fault analysis of the NTRUEncrypt cryptosystem. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 94(4):1156–1158, 2011.
32. A. Kipnis, J. Patarin, and L. Goubin. Unbalanced oil and vinegar signature schemes. In *Advances in Cryptology EUROCRYPT 99*, volume 1592 of *Springer LNCS*, pages 206–222. 1999.
33. L. Lamport. Constructing Digital Signatures from a One-Way Function. Technical report, CSL-98, SRI International, 1979.
34. Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta. Fault sensitivity analysis. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, volume 6225 of *Springer LNCS*, pages 320–334. 2010.
35. S. Mangard, E. Oswald, and T. Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer, 2008.
36. R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978.
37. R. C. Merkle. A Certified Digital Signature. In *CRYPTO*, pages 218–238, 1989.
38. R. Misoczki and P. Barreto. Compact McEliece keys from Goppa codes. In *Selected Areas in Cryptography*, volume 5867 of *Springer LNCS*, pages 376–392. 2009.
39. R. Misoczki, J.-P. Tillich, N. Sendrier, and P. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2069–2073, 2013.
40. H. Molter, M. Stöttinger, A. Shoufan, and F. Strenzke. A simple power analysis attack on a McEliece cryptoprocessor. *Journal of Cryptographic Engineering*, 1(1):29–36, 2011.

41. A. Moradi, O. Mischke, and T. Eisenbarth. Correlation-Enhanced Power Analysis Collision Attack. In *Cryptographic Hardware and Embedded Systems – CHES 2010*, volume 6225 of *Springer LNCS*, pages 125–139. 2010.
42. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.
43. K. Okeya, T. Takagi, and C. Vuillaume. On the importance of protecting delta; in SFLASH against side channel attacks. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 560–568 Vol.2, 2004.
44. N. Patterson. The algebraic decoding of Goppa codes. *Information Theory, IEEE Transactions on*, 21(2):203–207, 1975.
45. X. F. A. A. D. M. L. Pei Luo, Yunsi Fei and D. R. Kaeli. Power analysis attack on hardware implementation of mac-keccak on fpgas. Cryptology ePrint Archive, Report 2014/854, 2014.
46. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
47. A. Shoufan, F. Strenzke, H. Molter, and M. Stöttinger. A timing attack against patterson algorithm in the McEliece PKC. In *Information, Security and Cryptology, ICISC 2009*, volume 5984 of *Springer LNCS*, pages 161–175. 2010.
48. A. Shoufan, T. Wink, G. Molter, S. Huss, and F. Strentzke. A novel processor architecture for McEliece cryptosystem and fpga platforms. In *Application-specific Systems, Architectures and Processors, 2009. ASAP 2009. 20th IEEE International Conference on*, pages 98–105, 2009.
49. J. Silverman and W. Whyte. Timing attacks on NTRUEncrypt via variation in the number of hash calls. In *Topics in Cryptology, CT-RSA 2007*, volume 4377 of *Springer LNCS*, pages 208–224. 2006.
50. J. Stern. A new identification scheme based on syndrome decoding. In *Advances in Cryptology CRYPTO 93*, volume 773 of *Springer LNCS*, pages 13–21. 1994.
51. F. Strenzke. A timing attack against the secret permutation in the McEliece PKC. In *Post-Quantum Cryptography*, volume 6061 of *Springer LNCS*, pages 95–107. 2010.
52. F. Strenzke. Message-aimed side channel and fault attacks against public key cryptosystems with homomorphic properties. *Journal of Cryptographic Engineering*, 1(4):283–292, 2011.
53. F. Strenzke. Timing attacks against the syndrome inversion in code-based cryptosystems. In *Post-Quantum Cryptography*, volume 7932 of *Springer LNCS*, pages 217–230. 2013.
54. F. Strenzke, E. Tews, H. Molter, R. Overbeck, and A. Shoufan. Side channels in the McEliece PKC. In *Post-Quantum Cryptography*, volume 5299 of *Springer LNCS*, pages 216–229. 2008.
55. M. Taha and P. Schaumont. Differential power analysis of mac-keccak at any key-length. In *Advances in Information and Computer Security*, volume 8231 of *Springer LNCS*, pages 68–82. 2013.
56. M. Taha and P. Schaumont. Side-channel countermeasure for SHA-3 at almost-zero area overhead. In *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*, pages 93–96, May 2014.
57. S. Tang, H. Yi, J. Ding, H. Chen, and G. Chen. High-speed hardware implementation of rainbow signature on fpgas. In *Post-Quantum Cryptography*, volume 7071 of *Springer LNCS*, pages 228–243. 2011.
58. P. Véron. Improved identification schemes based on error-correcting codes. *Applicable Algebra in Engineering, Communication and Computing*, 8(1):57–69, 1997.

59. N. V. Vizev. *Side Channel Attacks on NTRUEncrypt*. PhD thesis, Bachelors thesis, Technical University of Darmstadt, Germany, 2007, 2007.

60. I. von Maurich and T. Güneysu. Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–6, March 2014.

61. I. von Maurich and T. Güneysu. Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices. In *Post-Quantum Cryptography*, volume 8772 of *Springer LNCS*, pages 266–282. 2014.

62. A. Wang, X. Zheng, and Z. Wang. Power analysis attacks and countermeasures on NTRU-based wireless body area networks. *KSII Transactions on Internet and Information Systems (TIIS)*, 7(5):1094–1107, 2013.

63. B.-Y. Yang and J.-M. Chen. Building secure tame-like multivariate public-key cryptosystems: The new tts. In *Information Security and Privacy*, volume 3574 of *Springer LNCS*, pages 518–531. 2005.

64. X. Zheng, A. Wang, and W. Wei. First-order collision attack on protected NTRU cryptosystem. *Microprocessors and Microsystems*, 37(67):601 – 609, 2013.