Investigation on lattices- and code-based post-quantum signature cryptosystems

Alexandra Feldman[1] , **Kaitai Liang**[1] , **Huanhuan Chen** [1]
[1]TU Delft

Abstract

As quantum-resistant cryptosystems will soon be necessary, the NIST has organized a contest aiming to its standardization. The proposed schemes must be evaluated and thoroughly investigated to notably ensure their security and compare their performance. This paper will explore various lattices-based (pqNTRUSign, BLISS, Dilithium, Falcon, qTesla) and code-based (RaCoSS, pqsigRM) digital signature schemes. An efficiency and security-based comparison is conducted among them and their features are discussed.

1 Introduction

Cryptographic systems are a crucial feature of current technology systems to ensure the safety of the communication and keep - among others - bank transfers, transactions and private messages secure. Most cryptosystems currently in-use are based on mathematical principles that are theoretically unbreakable by a classical computer, or at least it would take too much time to break such cryptosystems, ensuring their safety.

A new computation paradigm is arising and will challenge the security of those cryptosystems. Indeed, the quantum computers that are being built are increasingly powerful and it is likely that it is only a matter of time before one will be able to break standardly used cryptosystems.

While the most powerful quantum computers contains around 65 bits for now, the ambitions in this constantly evolving field are unbounded [3] .

One of the prominent company in this field - IBM - is planning to build a quantum computer with around 1000 bits by 2023 [3]. There is a need for quantum-resistant cryptosystems as most current cryptosystems would be easily breakable by such quantum computer. A thorough investigation of potential post-quantum systems is necessary to ensure their resilience to security attacks and evaluate their practical and theoretical efficiency.

The National Institute of Standards and Technology (NIST) has organized a contest to gather quantum-safe cryptosystems proposals [9]. Overall, 69 cryptosystems, consisting of 49 encryption - PKE (Private Key Encryption) or KEMs (Key Encapsuling Mechanism) - and 20 digital signature schemes, were granted an entry to the first round [10]. This paper contributes to the research conducted over the candidates schemes picked for the first round as the third (and last) round is currently taking place.

The cryptosystems can be further categorized as follows: code-based, hash-based lattice-based, multivariate, supersingular isogeny-based. This paper will focus on the digital signature family and most specifically on the lattices-based and code-based cryptosystems. This paper aims to offer a state-of-the-art review of 7 cryptosystems by answering the following sub-questions: what are their individual features, to what extent are they resilient to security attacks, what is their level of efficiency and complexity both in theory and practice. Moreover, this paper aims to explore the common features and potential vulnerabilities specific to the lattices-based and code-based family.

It should be noted that out of the 7 cryptosystems analyzed in the paper, 6 were effectively submitted to the NIST contest and 2 are currently in the third round.

First, an introduction to the background concepts will be provided, followed by the comparison over the lattices-based and code-based schemes. The method will be covered in the subsequent section. To conclude, the obtained results will be discussed.

2 Background

This section will provide an introduction to the background information over the concepts covered in this paper.

2.1 Digital signature

A digital signature scheme allows an entity to sign a document while guaranteeing its integrity and authenticity.

Consider Alice who wants to share to the world her own certified version of a document. As shown in Figure 1, she can use her private key to sign it and any other person will be able to use her public key to decrypt it and by the same occasion verifies the authenticity of the document. Digital signature systems are a standard tool used to protect the communication and integrity of data in various fields.

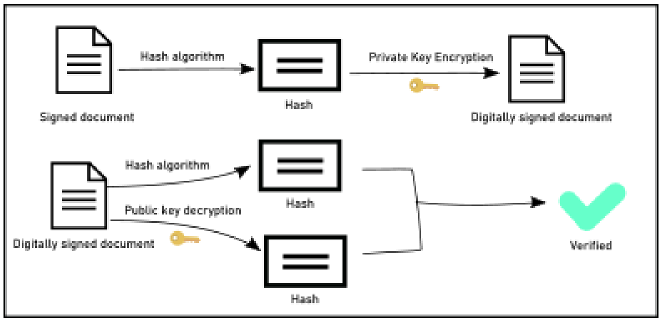

Figure 1: A digital signature scheme Λ

A digital signature scheme generally consists of three elements :

- A key generation algorithm which generates both a private and a public key
- A signing algorithm which generates a signature given a message and a private key
- A signature verifying algorithm which will either confirm or infirm the authenticity of the message given its public key and signature

.

2.2 Lattices-based cryptography

Lattices-based cryptography is a research field concerned with finding cryptographic tools and protocols relying on lattices-based hard problems [29] . Following a short introduction on lattices, the various hard problems constituting the basis of the lattices-based schemes analyzed in this paper will be presented.

Lattice

Let $n, d \in Z$ and $n \leq d$ and $\{b_1, ..., b_n\} \in R_d$ as linearly independent vectors. A lattice $\Lambda \subset R_d$ can be defined as follows :

$$\Lambda = \delta(b_1, ..., b_n) := \sum b_i x_i : x_i \in Z \ \forall i$$

A lattice can be represented as a regular infinite grid of points in the d-dimensional space and can be described by a finite number of linearly independent vectors. [2]

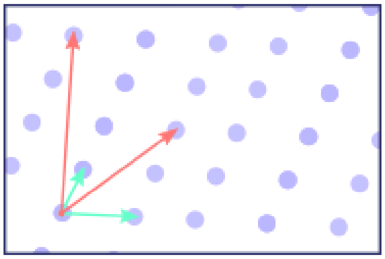

Figure 2: A lattice Λ represented as a grid, with two distinct basis

The set of vectors $\{b_1, ..., b_n\}$ is said to constitute the *basis* of the lattice Λ.

A basis is considered to be good when the basis vectors are orthogonal and have relatively small norms. On the other hand, a basis with non-orthogonal long vectors will be called a bad basis. For example, in [2], the green basis is a good basis, and the red basis is a bad basis.

The Closest Vector Problem

Given a lattice Λ with a basis v_i and a random point p, the CVP consists of finding the lattice point $p' \subset \Lambda$ closest to the point p.

The Shortest Vector Problem

Given a lattice Λ with a basis v_i and a vector w, the SVP [28] consists of finding the shortest non-zero lattice vector u minimizing $|u|$.

The Shortest Integer Solution problem

The SIS problem consists of finding the smallest non-trivial solution to a matrix α for $\alpha * x = 0$. It usually requires to only find a solution smaller than ϵ instead of the shortest. Constraining the solution to be smaller than ϵ make it unsolvable easily by Gaussian elimination.

The LWE problem

The learning with errors problem was first introduced by Oded Regev in 2005 [34]. It became the basis of various cryptographic applications over the recent years after Regev provided quantum reductions from the above mentioned problems to LWE which supported its usefulness. A basic LWE-based scheme, works as follow : after drawing a secret key $s \subset Z_q^n$ from a uniform distribution, the public key is generated and will be constituted of the pair of vectors (X, Y).

First, a vector X with $x_i \subset Z_q^n$ with q being a prime number is generated, followed by an error vector e. We can define next Y as $y_i = x_i s + e_i \mod (q)$. A LWE distribution generates a sample from a secret vector s and a distribution χ. The sample (a, b) is made of a uniformly sample a and $b = (a, s) + e$ with e drawn from the distribution χ.

There are two different versions of LWE, the search version which attempts to retrieve the secret key s and the decision version which attempts to distinguish between polynomials drawn from the LWE distribution and polynomials uniformly generated.

Due to the lack of underlying algebraic structure, the operations performed over LWE-based cryptosystems can be very expensive and slow which led to the creation of RLWE (Ring-LWE) and MLWE (Module-LWE) constraining the ring over which LWE distributes its sample. In opposition to LWE, in RLWE the operations are performed over a finite field $Z/qZ = F_q$ with q as a prime integer. MLWE constraints the elements further using module over the elements of a same ring.

2.3 Code-based cryptography

A short introduction to code-based cryptography and underlying concepts of the investigated cryptosystems will be provided here.

In 1978, McEliece constructed a cryptosystems based on binary Goppa codes [12]. Since, several code-based cryptosystems have been proposed. Code-based cryptosystems are based on the concept that linear code can be represented as a generator matrix which will thus constitutes the public key. A code based system works by adding noise to a message and using an error-decoding code to decrypt it. For reference, a code is represented by the list $[n, k, d]$ with n being the total bits codeword, k the message bit and d the distance.

More formally, if Alice wants to send a secured message m to Bob, she first encrypts it :

$$y = mG' + e$$

with e being a random error vector with weight w and length n [32]. The public key G' will be the $k \times n$ matrix SGP with P being a $k \times k$ permutation matrix. Alice sends y to Bob who will decrypt it by computing

$$yP^-1 = (mS)G + eP^-1$$

In this paper, pqsigRM and RaCoSS-R are analyzed. The former is based on Reed-Muller code and the latter is based on the Syndrom Decoding problem over random linear code.

Reed-Muller code

Reed muller is a class of cyclic multiple-error-correction codes. Formally, for any p and $r \subset Z$ with

$$0 \leq r \leq p$$

there exists a binary Reed-Muller code $RM(r, p)$ of the r-order [35]. It carries the following parameters :
a block-length :

$$n = 2^p$$

the dimension :

$$k = \sum_{i}^{r} \binom{p}{j}$$

and a minimum distance :

$$d_{min} = 2^p - r$$

.

A $RM(r, p)$ code corrects up to $t_e = 2^p - r - 1 - 1$ errors.

Syndrome Decoding problem

The syndrom decoding consists of a method to decode linear codes over a noisy channel.

Considering a code (n, k, d), a parity-check matrix $H \subset F_2^{(n-k)*n}$ and the "syndrome": a vector $v \subset F_2^{(n-k)}$. The problem consists of finding a vector $s \subset F_2^n$ with a hamming weight $\leq d$ such that $Hs^T = v^T$.

3 Methodology

First, the formal description and white papers of each cryptographic system were analyzed individually in order to gain knowledge over the underlying theoretical concepts of the various algorithms. The related work on the subject was then analyzed. Since the beginning of the NIST contest, extensive research has been performed on the candidates. The candidates have been analyzed individually and a few papers compare some cryptosystems belonging to the lattices-based digital family [31] [20] [21]. Some attacks were found to be effective after security review of some cryptosystems [27] and more general security review of lattices-based schemes [18]. Extensions and improvements of some cryptosystems were proposed as well [22] [24]. While the literature regarding the code-based family is not as extensive, some surveys were still conducted [23]. This is due to both the fact that lattices-based were considered as one of the most promising concept for post-quantum cryptosystems and no code-based cryptosystems were accepted into the third round. Moreover, the NIST has published status reports reports following the end of the first rounds [30] [4].

After, a theoretical evaluation was conducted to compare the cryptosystems using a subset of metrics that were derived from the preparation of the research of the project.

Next, an analysis was conducted with several goals in mind. First, the theoretical performance of each cryptosystem was compared to its practical performance. Their implementation was then analyzed at a lower-level to evaluate the efficiency of their distinguishable features.

4 Investigation over selected schemes

This section provides an extensive comparison of the selected lattices-based and code-based digital schemes.

4.1 Lattices-based digital schemes

After a brief introduction to each lattices-based digital schemes, a performance, efficiency and security analysis will be provided followed by a discussion.

Lattices-based schemes are generally either hash-and-sign based or constructed using Fiat-Shamir protocol.

NTRU is a hash-and-sign based cryptosystem that was originally constituted of NTRUSign (for digital signatures) and NTRUEncrypt (for encryption). Originally, NTRU represents a specific class of lattices (The Nth TRUncate polynomial ring) and is the basis of the algorithm scheme of the same name based on the shortest vector problem [19].

NTRU's algorithms have evolved along the years and **pqNTRUSign** [13] - a modular lattice scheme which is a modified version of NTRUSign published after it was broken - was a candidate of the NIST contest. NTRU is hash-and-sign based (as opposed to BLISS which is based on Fiat-Shamir heurisitics) which implies a potential leak of information that can lead to the recovery of the private key due to not having a zero-knowledge signature scheme. Following attacks exploiting information leaked from a candidate signature to recover the secret key, rejection sampling was introduced and is used in **pqNTRUSign** to protect information.

For a given lattice Λ with a trapdoor Θ - a short basis of row vectors - and a messsage m, the goal is to find a vector v such that $v \subset \Lambda$ and $v \equiv hash(m) \mod p$.

Its signing algorithm works by first hashing the message and PK into a vector, then generating a mask vector from a sampler.

It verifies the authenticity by reconstructing the lattice vector and checking if the hash of the vector is a basis of the lattice.

Among the two parameters sets provided, Uniform-1024 and Gaussian-1024 (using a bimodal Guassian distribution for its rejection sampler), uniform-1024 was used for the original pqNTRUSign scheme and was chosen for the analysis in this paper.

BLISS [15] is the only cryptosystem investigated in this paper that was not proposed to the NIST contest among the selected list of lattices-based schemes. However, it was once considered as one of the most promising cryptosystem in the lattices-field and constituted an inspiration for Dilithium and qTesla. When it was first proposed, BLISS was the most advanced scheme using the heuristics of Fiat-Shamir with aborts [26] on an identification scheme (using rejection sampling) which allows for transforming an identification scheme into a signature scheme. Its main contribution is the use of bimodal gaussian distribution for the rejection sampling which efficiently reduces the rejection rate.

While its original key generation algorithm lies on the $R - SIS^K_{q,n,m,\beta}$ problem, the most efficient implementation is using a NTRU-based algorithm and will be the one considered here.

The generation algorithm of BLISS outputs a pair of key (S, P). The secret key S is a short matrix $\subset \mathbf{Z^m_{2q}xn}$ and the public key is a matrix P $\subset \mathbf{P^n_{2q}xm}$ such that AS = qI_n

Dilithium [14] is part of the Crystals suite along with Kyber (for encryption). It is based on the hardness of Module-SIS and Module-LWE problems. Its operations are performed over the ring $Z_q[X]/(X^{256} + 1)$ with $q = 8,380,416$.

The idea behind Dilithium was to avoid having to rely on gaussian sampling and NTRU assumptions which led to minimize the size of the public key and the signature while offering a better security. It was thus constructed using the heuristics of Fiat-Shamir with aborts which allows for a generic transformation of a zero-knowledge proof into a non-interactive zero-knowledge proof.

It is one of the most serious competitor for the NIST contest. Dilithium provides different parameters set to fit all NIST security levels classification.

qTesla [5] is also constructed using Fiat-Shamir with aborts. It is based on the hardness of Ring-LWE. qTesla provides two parameters sets (the category level refers to NIST post-quantum security classification [11]):

- qTesla-p-I with a security level category 1
- qTesla-p-III with a security level category 3

It embeds a simplified gaussian sampler which is only required during the key generation process. Using cumulative distribution table of the normal distribution it succeeds in having an efficient constant-time sampler.

Falcon [16] uses NTRU lattices class and is based on GPV (Gentry-Peikert-Vaikuntanathan) [17] Gaussian sampling framework.

The GPV framework was introduced as a hash-and-sign algorithm which successfully attempts to fix the flaws found in NTRUSign. Falcon also introduced Fast Fourier Sampling improving the sampling recommended by GPV (which was based on Babai's algorithm) and manages to make the sampling process faster. **Falcon** has a non constant-time implementation which is due to its use of a discretized sampler. Falcon also uses NTRU lattice as a lattice class. However, it offers a major improvement to the gaussian sampling of NTRU by making use of the Falcon tree data structure.

Its pair of keys (p, k) is constituted of a long basis for its public key and a short basis for its private key. Its signing process works as follow: The signer generates a random value, computes a target $c{=}Hash(msg, salt)$. It then uses its knowledge of the good basis to compute a lattice point v close to the target c. Finally it outputs the signature $(salt, s)$ with $s = c - v$.

To verify, it first verifies if the vector s is short and then if $c - s$ is actually a correct point on the lattice generated by the good basis.

Two parameters sets are provided :

- Falcon-512 with a security level 1
- Falcon-1024 with a security level 5

Performance

The tables below give an indication of the performance and the speed of the investigated lattices-based signature schemes. The missing values represent values that were not successfully retrieved due to implementation bugs (in the case of pqNTRUSign).

The following run-time analysis is based on the number of cycles for each operation.

Impl.	keygen	Sign	Verify
BLISS-BII	745137	4388632	862106
pqNTRUSign	-	-	-
Dilithium3	544232	2348703	522267
Falcon-1024	284731	1925901	421802
qTesla-p-III	603481	3591433	703246

Falcon and dilitihium appears to be the most efficient scheme. BLISS-BII appears be least efficient.

Impl.	Gen/s	Sign/s	Verify/s
BLISS-BII	-	4.91k	14.231K
pqNTRUSign	-	-	-
Dilithium3	13.676k	3.679k	18.641k
Falcon-1024	14.336k	5.9481k	27.933k
qTesla-p-III	12.642k	3.0819k	17.453k

Security

Impl.	Sec.	Sig. Size	SK size	PK size
BLISS-BII	128 bits	4.987 kb	2.321 kb	6.872 kb
pqNTRUSign	-	-	-	-
Dilithium3	128 bits	2.519 kb	2.121 kb	1.739 kb
Falcon-1024	128 bits	1.342 kb	1.7654 kb	1.847 kb
qTesla-p-III	160 bits	5.558 kb	12.172 kb	37.382 kb

Falcon has the shortest combination secret key and public key, followed by dilithium and qTesla. qTesla and pqNTRUSign carry a particularly big overhead which makes them practically less versatile and decrease the flexibility for their implementations (as it therefore requires more memory)

Resiliency to attacks
As mentioned before, BLISS was not proposed during the NIST contest while it was one of the pioneer lattices scheme. After an apparent resilience to attacks when considered in a purely theoretical context, a cache side-channel attack was successfully conducted on the strongSwan IPsec-based VPN suite which implemented BLISS [33]. The attack was an improvement of a previously proposed attack (in a theoretical context) [7] which consisted of exploiting the knowledge of the cache to infer information about the gaussian sampler that will lead to the reconstruction of a set of linear equations. It can then be used to recover the full signing key.

As having a non-constant sampler often induces vulnerabilities to timing attacks and the implementation of a constant-time sampler can heavily decrease the efficiency, a fault attack was similarly found to be successful on Falcon [27]. Ensuring a constant-time sampling algorithm was proposed to solve the issue while causing a 5% slowdown on the key generation process.

Flacon embeds a variant of the Gaussian sampling framework introduced in GPV which act like an oracle that samples lattice points p - according to a discrete gaussian distribution - centered around an arbitrary point a on a lattice Λ with a secret short basis B. As it samples, it does not leak information about the inherent structure of the lattice and is less vunerable to attack exploiting the algebraic structure of the NTRU lattices.

Fault attacks were found to be effective on deterministic lattices-based schemes. While Dilithium do not embed a gaussian sampler, a fault-attack was presented on Dilithium and countermeasures were proposed which induced a 20% slowdown on the performance. Similarly qTesla has a constant-time sampler and does not rely on an underlying lattices structure so it is less prone to timing attacks [8]. However, a fault attack could potentially harm qTesla. A countermeasure proposed consists of injecting randomness in each signing operation to avoid fault to allow the revealing of the secret key if the hash value is faulted.

No attacks were found yet on pqNTRUSign. pqNTRUSign is originally a redesign of a previously successfully attacked scheme. However, it did not pass the first round and the lack of literature around this scheme might be a reason for the lack of investigation around its security.

Discussion
While the simplicity of the qTesla's design is its main advantage, it causes an overhead regarding the size of the keys. Falcon's intricated design (embedding the falcon tree) makes it the fastest algorithm for signature generation. qTesla has the second fastest verification and generation processes. As no extensive vulnerabilities investigation has been conducted on Falcon yet, its extensive use GPV (and thus gaussian sampling) could potentially expose it to cache-side channel attacks, as they've been proven efficient on such process.

Dilithium and Falcon are the most promising in their own category. They have drawbacks attached to this feature. It could be imagined that they are not absolutely better but it relies a lot on the specific environment and context.

4.2 Code-based schemes

pqsigRM [25] was proposed as an improvement of CFS [6] - a goppa code-based scheme by using a modified Reed-Muller code instead. It takes advantage of the closet coset decoding provided by reed-muller to increase its efficiency.

The following parameter sets are provided

- pqsigRM-5-11 achieving security level 1
- pqsigRM-6-12 achieving security level 3
- pqsigRM-6-13 achieving security level 5

It improves the CFS scheme by disentangling the signing time to the error correction capability and make it parametrisable. However its key size $(n - k) * k$ is relatively large.

The **RaCoSS-R** (Random Code-based Signature Scheme) scheme provides an efficient implementation based on the syndrome decoding problem. However, it was broken shortly after it was proposed and no working fix was proposed.

2 different parameter sets are provided, one in the reference implementation, one as an optimized version.

Performance

Impl.	keygen	Sign	Verify
RaCoSS	231.763	3.258k	324.873
pqsigRM-5-11	361.942	4.139k	472.523

Impl.	Gen/s	Sign/s	Verify/s
RaCoSS	165426	276857	331678
pqsigRM-5-11	124815	232526	241458

Security

Impl.	Sec.	Sig. Size	SK size	PK size
RaCoSS	128 bits	7.2 kb	4.36 kb	8.72 kb
pqsigRM-5-11	128 bits	15.97 kb	14.23kb	12.43 kb

Resiliency to attacks
RaCoSS was found to be vulnerable to a fault attack and was not selected for the second round [36]. No attack were found on pqsigRM and it was proven secure under EUF-CMA. (It directly benefits from the security of CFS that was previously proved).

Discussion
The security size of RaCoSS is small in comparison to pqsigRM. The reference implementation of RaCoSS is the most efficient when it comes to key generating and verification. However, RaCoSS has been proven vulnerable multiple times. Thus, pqsigRM seems the most promising code-based digital scheme.

5 Discussion

Overall, the lattices-based digital scheme are more promising than the code-based schemes. The simplicity of the underlying mathematical concepts offers flexibility for implementation. The schemes built with Fiat-shamir with aborts are generally more secure as they don't suffer from vulnerabilities related to timing attacks. However, Falcon seems like it could be interesting although some of its features might make it unusable in practical situation. This might mean that there is a need for standardizing multiple cryptosystem as the best one to use depends heavily on the needs and the device.

6 Responsible Research

As a bachelor thesis, this paper is not linked directly to NIST or any of the candidates. Each of the submission was analyzed objectively based on their official specifications provided by NIST.

6.1 Reproducibility of the experiment

This paper did not require extensive non-theoretical processes. The information reported here are gathered from the official NIST website and literature study on the topic.

The section that are potentially reproducible concern the performances and security. In order to gather the data reported in section. All benchmarks were obtained on one core of an Intel Core i5-650. The implementations used to report the benchmarks for the candidates schemes can be found here [9] and the implementation for BLISS can be found here [1].

7 Conclusions and Future Work

Among the 7 analyzed schemes, only 2 are still candidates for the contest : Dilithium and Falcon.

The code-based algorithms are still prone to attacks and further investigation could be performed in order to propose improvements. The efficiency and security offerd by the lattices-based algorithm seem to beat the one of the code-based family. Dilithium seem like the second most efficient algorithm while offering a beneficial compromise allowing a flexible and easy implementation. Falcon on the other hand proposed the falcon tree which provides an innovative solution for hash-and-sign based algorithms regarding the efficiency. On the other hand, more investigation on cache-sided attack should be performed as it has commonly been successful on non fiat shamir based algorithms (as they are not non-interactive non-knowledge) proofs.

While the NIST aims to pick one algorithm for the standardization the advantages offered by falcon and dilihtium might be a sign that the best cryptosystem is context-dependent and while dilithium might find a more wide-scale use, falcon could be useful when compactness is the primary need.

References

[1] Bliss: Bimodal lattice signature schemes.

[2] *Constellation Shaping, Nonlinear Precoding, and Trellis Coding for Voiceband Telephone Channel Modems with Emphasis on ITU-T Recommendation V.34*$_2$002. 2002.

[3] 2020 Adrian ChoSep. 15. Ibm promises 1000-qubit quantum computer-a milestone-by 2023, Sep 2020.

[4] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Yi-Kai Liu. Status report on the first round of the nist post-quantum cryptography standardization process, 2019-01-31 2019.

[5] Erdem Alkim, Paulo SLM Barreto, Nina Bindel, Juliane Krämer, Patrick Longa, and Jefferson E Ricardini. The lattice-based digital signature scheme qtesla. In *International Conference on Applied Cryptography and Network Security*, pages 441–460. Springer, 2020.

[6] Paulo SLM Barreto, Pierre-Louis Cayrel, Rafael Misoczki, and Robert Niebuhr. Quasi-dyadic cfs signatures. In *International Conference on Information Security and Cryptology*, pages 336–349. Springer, 2010.

[7] 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.

[8] Leon Groot Bruinderink and Peter Pessl. Differential fault attacks on deterministic lattice signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 21–43, 2018.

[9] Information Technology Laboratory Computer Security Division. Post-quantum cryptography.

[10] Information Technology Laboratory Computer Security Division. Round 1 submissions - post-quantum cryptography: Csrc.

[11] Information Technology Laboratory Computer Security Division. Security (evaluation criteria) - post-quantum cryptography: Csrc.

[12] Nicolas T Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–174. Springer, 2001.

[13] Dipayan Das, Jeffrey Hoffstein, Jill Pipher, William Whyte, and Zhenfei Zhang. Modular lattice signatures, revisited. *Designs, Codes and Cryptography*, 88(3):505–532, 2020.

[14] Léo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals–dilithium: Digital signatures from module lattices. 2018.

[15] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. Cryptology ePrint Archive, Report 2013/383, 2013. https://eprint.iacr.org/2013/383.

[16] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. *Submission to the NIST's post-quantum cryptography standardization process*, 36, 2018.

[17] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008.

[18] Qian Guo, Thomas Johansson, and Alexander Nilsson. A generic attack on latticebased schemes using decryption errors. Technical report, Cryptology ePrint Archive, Report 2019/043. 2019 Cited on, 2019.

[19] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *Algorithmic Number Theory*, pages 267–288, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[20] Malik Imran, Zain Ul Abideen, and Samuel Pagliarini. An experimental study of building blocks of lattice-based nist post-quantum cryptographic algorithms. *Electronics*, 9(11), 2020.

[21] Malik Imran, Zain Ul Abideen, and Samuel Pagliarini. A systematic study of lattice-based nist pqc algorithms: from reference implementations to hardware accelerators. *arXiv preprint arXiv:2009.07091*, 2020.

[22] Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. Pushing the speed limit of constant-time discrete gaussian sampling. a case study on the falcon signature scheme. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, New York, NY, USA, 2019. Association for Computing Machinery.

[23] Alexandr Kuznetsov, Anastasiia Kiian, Mariia Lutsenko, Iryna Chepurko, and Sergii Kavun. Code-based cryptosystems from nist pqc. In *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pages 282–287, 2018.

[24] Wai-Kong Lee, Sedat Akleylek, Wun-She Yap, and Bok-Min Goi. Accelerating number theoretic transform in gpu platform for qtesla scheme. In *International Conference on Information Security Practice and Experience*, pages 41–55. Springer, 2019.

[25] Yongwoo Lee, Wijik Lee, Young Sik Kim, and Jong-Seon No. Modified pqsigrm: Rm code-based signature scheme. *IEEE Access*, 8:177506–177518, 2020.

[26] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 598–616. Springer, 2009.

[27] Sarah McCarthy, James Howe, Neil Smyth, Séamus Brannigan, and Máire O'Neill. Bearz attack falcon: Implementation attacks with countermeasures on the falcon signature scheme. *IACR Cryptol. ePrint Arch.*, 2019:478, 2019.

[28] Daniele Micciancio. *Closest Vector Problem*, page 79–80. Springer US, 2005.

[29] Daniele Micciancio. *LATTICE BASED CRYPTOGRAPHY*, page 347–349. Springer US, 2005.

[30] Dustin Moody, Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Jacob Alperin-Sheriff. Status report on the second round of the nist post-quantum cryptography standardization process, 2020-07-22 2020.

[31] Hamid Nejatollahi, Nikil Dutt, Sandip Ray, Francesco Regazzoni, Indranil Banerjee, and Rosario Cammarota. Post-quantum lattice-based cryptography implementations: A survey. *ACM Comput. Surv.*, 51(6), January 2019.

[32] Raphael Overbeck and Nicolas Sendrier. *Code-based cryptography*, pages 95–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[33] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. To bliss-b or not to be: Attacking strongswan's implementation of post-quantum signatures. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1843–1855, 2017.

[34] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery.

[35] A.M. Steane. Quantum reed-muller codes. *IEEE Transactions on Information Theory*, 45(5):1701–1703, 1999.

[36] Keita Xagawa. Practical attack on racoss-r. Cryptology ePrint Archive, Report 2018/831, 2018. https://eprint.iacr.org/2018/831.