ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Post quantum cryptography in practice: a survey

Adrian Valente

School of Computer and Communication Sciences

Semester Project

December 2017

Responsible	**Supervisor**
Prof. Serge Vaudenay	Prof. Serge Vaudenay
EPFL / LASEC	EPFL / LASEC

LASEC

Abstract

As fears of a usable quantum computer being developed have been growing, the field of post-quantum cryptography has flourished to offer alternatives to the asymmetric cryptographic primitives used today. Acknowledging the importance of this task, the NIST has launched a standardization process which has started at the end of 2017, and will represent the deepest scrutiny this field has ever received. As this important period is about to start, this report aims at giving a snapshot of the state of post-quantum cryptography, and a basic understanding of the different ideas used to build post-quantum cryptosystems, of their computational foundations, of the practical implementations that have been developed and of how they can be compared.

Acknowledgments

I want to thank specifically Serge Vaudenay for his guidance, and for making this work possible, as well as Damian Vizár, Betül Durak, and Gwangbae Choi for their help.

Contents

Introduction

As our electronic communications rely increasingly on the use of cryptography, it is important to assess the long-term security of our cryptosystems. The development of a sufficiently powerful quantum computer could completely disrupt that security by breaking some cryptographic primitives. It is thus essential that alternatives are seeked before such an event occurs.

In order to tackle that challenge, the American standard association (NIST) has launched a program to study different alternatives and eventually develop a standard that could be deployed in typical cryptography use cases. As of the 20th December 2017, 69 valid proposals have been submitted to this process, which will last several years.

Before going any further, let us first understand which cryptographic primitives would be broken by a quantum computer, and which would remain secure. The most used cryptographic primitives are the following: cryptographic hash functions, symmetric encryption schemes, asymmetric encryption schemes, digital signatures. The first two types of primitives are generally based on heuristic ideas that make breaking them almost as hard as solving a general search problem. Although quantum computers bring a quadratic improvement to general search algorithms with Grover's algorithm, the complexity of cryptanalysis would remain exponential, and doubling the key size would maintain the same security level.

The situation is more intricate for asymmetric encryption schemes and digital signatures. Those primitives are based on problems believed to be hard, for which a trapdoor (the private key) exists. To ensure the security of such a scheme under a quantum computer, the underlying problem must be hard to solve even for a quantum computer. However, all asymmetric encryption schemes and digital signatures used in practice today rely on two problems: integer factorization (RSA) or discrete logarithm, either on integer rings or on elliptic curves (Diffie-Hellman, DSA, ECDSA). Those two problems can be solved in polynomial time by a quantum computer. Even if quantum computers remain out of reach for a long time, it seems unsecure to rely exclusively on those two problems, and be at the mercy of new ideas in number theory. Thus it becomes urgent to seek alternatives to those schemes. Given this security assessment we will focus on the rest of this report on two cryptographic primitives: digital signature schemes and key encapsulation mechanisms based on asymmetric encryption. These are indeed the two essential bricks of our network security that are likely to be broken by a quantum computer. The precise definitions of these primitives can be found in appendix A.

As we will see, many alternative cryptosystems have been imagined, since even before integer factorization and discrete logarithm were known to be quantum insecure: it started with the Merkle-Hellman knapsack cryptosystem which has been broken long ago, but cryptographers have produced a significant literature of cryptosystems based on very diverse hard problems. These so called post-quantum cryptosystems can broadly be divided into five categories, each of these being defined by a set of closely related hard problems:

- **Multivariate cryptography**, which relies on the difficulty of solving multivariate polynomials of degree at least 2;
- **Lattice-based cryptography**, which relies on the difficulty of finding shortest vectors in a lattice, and the closely related learning with errors problem;
- **Code-based cryptography**, which relies on the difficulty of decoding a random code;
- **Hash-based cryptography**, which only provides signature schemes, relying on the security of hash functions;
- the newer field of **supersingular isogeny-based cryptography**, which so far has only given a key exchange scheme, based on the difficulty of finding isogenies between elliptic curves. We will only briefly review this one.

These methods do not always provide all the primitives that interest us. More precisely, the methods that can provide signatures and key exchange mechanisms are:

	Multivariate	Lattice-based	Code-based	Hash-based	Isogeny-based
Signatures	Yes	Yes	Possible, but not much studied	Yes	*a priori* no, but shown possible [69]
KEM	Possible, but not much studied	Yes	Yes	No	Yes

We will in the body of the report review the literature concerning post-quantum cryptosystems by following this division, which makes more sense than a separation between digital signatures and asymmetric encryption. Thus we will in each of the chapters of this report focus on one of these categories, by exposing the underlying problems, the history of its development, existing practical implementations, and their advantages and drawbacks.

Chapter 1

Multivariate Cryptography

The field of Multivariate Cryptography relies on the hardness of solving multivariate polynomial equations in finite fields. It has started with the cryptosystem C^*, developed by Matsumoto and Imai in 1988 [6], and has since been broken. However some competitive signature schemes based on the same ideas have been developed, and remain secure. Some encryption schemes based on multivariate equations have also been developed, but are a lot less competitive so far.

1.1 Main idea and underlying problems

Let $m, n \in \mathbb{N}$, q be a finite power of a prime number, $p_1, \dots, p_m$ be m polynomials in n variables with coefficients in a finite field $\mathbb{F}_q$. Let

$$\mathcal{P} : \mathbf{x} = (x_1, \dots, x_n) \mapsto (p_1(\mathbf{x}), \dots, p_m(\mathbf{x}))$$

be the associated map from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$. The problem of inverting $\mathcal{P}$ is known to be hard. More specifically, let us focus on the situation where the polynomials are all quadratic. We define the $\mathcal{MQ}$ problem to be :

Definition 1. *[62] On input $p_1, \cdots, p_m$ m polynomials in n variables with coefficients in a finite field $\mathbb{F}_q$ where $n, m \in \mathbb{N}$ and q is a finite power of a prime number, the $\mathcal{MQ}$ problem asks to find $\mathbf{x}^* \in \mathbb{F}_q$ such that $p_1(\mathbf{x}^*) = \cdots = p_n(\mathbf{x}^*) = 0$.*

This problem is known to be NP-complete in the general case [13], but we know very little of its hardness on specific cases of equations, which are used in practice.

In order to create a trapdoor function from the $\mathcal{MQ}$ problem, let us take a map $\mathcal{P}$ with some structure that makes it easily invertible. The idea is to hide that additional structure by two affine transformations before and after applying $\mathcal{P}$. Let S and T be affine transformations of ${\mathbb{F}_q}^n$ (resp. ${\mathbb{F}_q}^m$).

Then we define $\mathcal{P}^{pub} = T \circ \mathcal{P} \circ S$. By using $\mathcal{P}^{pub}$ as the public-key and $(\mathcal{P}, S, T)$ as the private key, we have created a cryptographic trapdoor.

Finally, we can use this as an encryption cryptosystem by using the map $\mathcal{P}^{pub}$ to encrypt a message $M \in {\mathbb{F}_q}^n$ and decrypt it by applying $S^{-1} \circ \mathcal{P}^{-1} \circ T^{-1}$ (note that in order for $\mathcal{P}^{-1}$ to be defined, $\mathcal{P}$ must be injective, and thus $m \geq n$). Similarly, we can build a signature scheme (this time $\mathcal{P}$ must be surjective, and applying $\mathcal{P}^{-1}$ can be replaced by finding any solution to $\mathcal{P}(x) = M$).

1.2 History

After the C* cryptosystem of Imai and Matsumoto has been broken, two main branches of multivariate signature schemes have arisen.

One is the HFE algorithms family, based on an idea by Patarin [7]. They use polynomials that are easy to invert in a field extension of $\mathbb{F}_q$ thanks to the Berlekamp algorithm. This field extension is supposedly "hidden", hence the name HFE for Hidden Field Equations (see appendix B for details). However this first version has been broken, and multiple ideas have been proposed in order to fix it. One of the most important proposals has been the HFE_{v-} variant, also known as QUARTZ, developed by Patarin, Courtois and Goubin [12]. The idea of this algorithm is to add v *"vinegar"* variables, which are redundant variables in the initial field, and to remove a *"minus"* equations from the arrival field, thus creating a map $\mathcal{P} : {\mathbb{F}_q}^{n+v} \rightarrow {\mathbb{F}_q}^{n-a}$. This scheme is still secure today. A similar algorithm, also proposed by Patarin, Courtois and Goubin in 2001 is the SFLASH signature scheme [11], who launched the category of C^{*-} cryptosystems. However, this scheme has been cryptanalyzed in 2005.

Another family of multivariate cryptosystems are the Oil & Vinegar schemes, also based on an idea by Patarin [8]. The idea of these schemes is to chose two numbers v and o. The main map will be quadratic in v *"vinegar"* variables, but only linear in the o *"oil"* variables, thus making it easy to invert (see appendix B for details). The first version proposed was broken, but Kipnis, Patarin and Goubin have proposed some conditions on v and o that made the scheme, now named UOV (for Unbalanced Oil & Vinegar), secure [9]. Other schemes have improved on this design, like the Rainbow scheme [14] which adds several layers of oil and vinegar variables.

1.3 Recent papers and implementations

- **Rainbow** (2005, [14]): although a bit old, this signature scheme remains the best instantiation of an Oil & Vinegar based algorithm. It

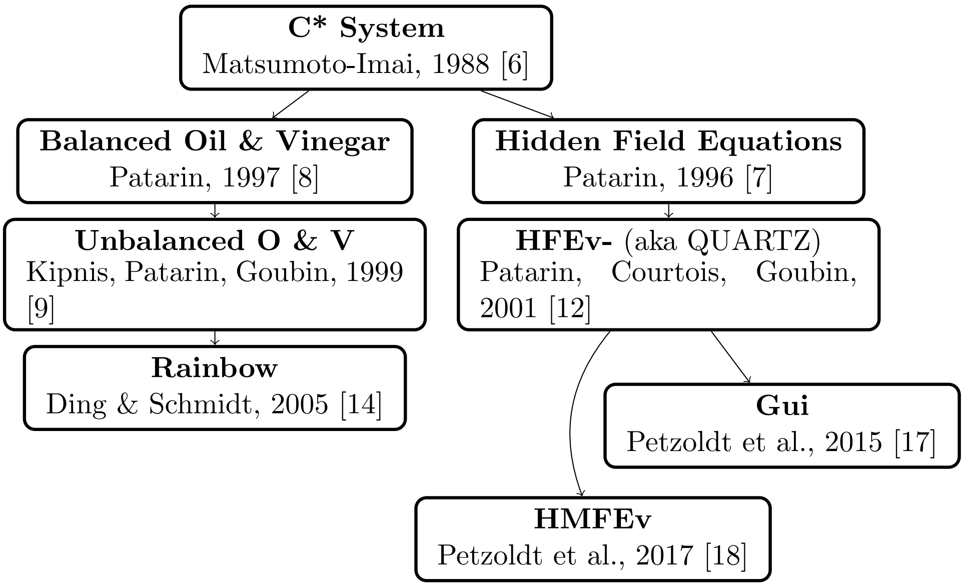

C* System
Matsumoto-Imai, 1988 [6]
Balanced Oil & Vinegar
Patarin, 1997 [8]
Hidden Field Equations
Patarin, 1996 [7]
Unbalanced O & V
Kipnis, Patarin, Goubin, 1999 [9]
HFEv- (aka QUARTZ)
Patarin, Courtois, Goubin, 2001 [12]
Rainbow
Ding & Schmidt, 2005 [14]
Gui
Petzoldt et al., 2015 [17]
HMFEv
Petzoldt et al., 2017 [18]

Figure 1.1: Evolution of multivariate cryptography

generalizes the Unbalanced Oil & Vinegar concept by adding several layers of vinegar variables, and the authors claim this improves the efficiency of the scheme.

- **Gui** (2015, [17]): this signature scheme is an improvement of the HFE_{v-} algorithm QUARTZ [12], for which the authors carefully choose the parameters.

- **HMFEv** (2017, [18]): this signature scheme is also HFE based. It is essentially a *"vinegar"* variant of the MultiHFE algorithm [15] which had been broken. The authors claim that their scheme is not vulnerable to the same attacks, and gives the possibility to use fields with bigger characteristics than Gui/QUARTZ (which are mainly restricted to binary fields), and thus gain efficiency.

1.4 Security assessment

There are two ways to attack a multivariate cryptosystem: either by solving the multivariate polynomial equations (these are the *algebraic attacks*, which use Gröbner bases), or by recovering the secret polynomial (this is the purpose of linearization attacks).

[1] Figures from [14]

[2] Figures for both algorithms from [18], table 3

	Rainbow[1]	Gui-96[2]	HMFEv[2]
Public Key Size (Kbytes)	15	61.6	22.5
Signature Size (bytes)	264	126	218
Signature Time (ms)	-	0.07	0.20
Verification Time (ms)	-	0.02	0.01

Table 1.1: Performance comparison of multivariate signature schemes for a security estimate of 80 bits

Linearization attacks were notably used by Kipnis and Shamir to break the HFE cryptosystem [10]. In their attack, they recover a univariate representation of the $\mathcal{P}^{pub}$ polynomial, by using only the public key. The rest of the attack consists in finding a small rank matrix (hence the name MinRank attack, which is also found in the literature). The complexity of the whole attack over an HFE_{v-} scheme is approximately $\mathcal{O}(q^{n(\log_q(d)+v+a-1)}(n-a)^3)$ [17] (see B for a description of the parameters).

Algebraic attacks compute Gröbner bases, which are a standard tool for solving multivariate polynomial systems. In [13] and [16], the security of HFE_{v-} schemes against these attacks is analyzed, but a simple estimate is not given.

1.5 Advantages and drawbacks

The performances listed in table 1.1 show that multivariate signature schemes are a strong candidate for post-quantum signatures. They notably achieve the smallest signature sizes of all cryptographic families [12], and are sufficiently simple to be implemented in smartcards (see comments in [9], or the SFLASH scheme [11]).

From a security perspective, although multivariate schemes have a long history of successfully broken proposals, some simple schemes like HFE_{v-} and UOV have resisted cryptanalysis for almost 20 years, and the recent security assessments for HFE_{v-} are very encouraging [16]. It is true that practical multivariate schemes do not have a formal proof of security, but the underlying $\mathcal{MQ}$ problem is known to be NP-complete, which can be an argument in favor of multivariate cryptography.

Finally, let us note that variants of existing multivariate schemes are easy to design (altough not always secure), and can easily be kept secret. As Kipnis, Patarin and Goubin note in [9]:

> "When a new scheme is found with multivariate polynomials,

> we do not necessary have to explain how the trapdoor has been introduced. Then we will obtain a kind of 'Secret-Public Key scheme' ! The scheme is clearly a Public Key scheme since anybody can verify a signature from the public key (or can encrypt from the public key) and the scheme is secret since the way to compute the secret key computations (i.e. the way the trapdoor has been introduced) has not been revealed and cannot be guessed from the public key. For example, we could have done this for HFEV (instead of publishing it)."

Even though we are mostly interested in public algorithms, and that Kerckoff's laws state that a cryptosystem has to be as secure as if it were public, it is always interesting to know which kind of algorithms might be used more secretly.

Chapter 2

Lattice-based Cryptography

The field of lattice-based cryptography originates in various ideas developed at the end of the 1990's, most notably a founding article by Ajtai [19], and the NTRU cryptosystem [22]. It is based on the hardness of several lattice problems, (most notably the shortest vector problem) and of the closely related LWE (Learning With Errors)-type problems. A lot of cryptographic primitives based on these problems have been developed, notably encryption schemes, key encapsulation mechanisms and signature schemes.

2.1 Underlying problems

The field of lattice-based cryptography spans an important number of computing problems, and very diverse cryptosystems. We can however understand most of them by focusing on two closely related problems: the Learning With Errors (LWE) problem, and the Shortest Vector Problem (SVP). In what follows, we will denote by $\langle \cdot, \cdot \rangle$ the scalar product of two vectors.

Definition 2. *[26] [32] Let $p, n \in \mathbb{N}$, $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_1, \cdots, \mathbf{a}_m$ independently and uniformly sampled from $\mathbb{Z}_p^n$, and $e_1, \cdots, e_m \in \mathbb{Z}_p$ be chosen independently according to a distribution χ. We define the b_i by:*

$$\langle \mathbf{s}, \mathbf{a}_1 \rangle + e_1 = b_1 \pmod{p}, \cdots, \langle \mathbf{s}, \mathbf{a}_m \rangle + e_m = b_m \pmod{p}$$

Given the $\mathbf{a}_i$ and b_i the $LWE_{p,\chi}$ problem (also called Search-LWE problem) asks to retrieve $\mathbf{s}$.

We can also define the Decision LWE problem:

Definition 3. *[36] [32] Let $p, n \in \mathbb{N}$, $\mathbf{s} \in \mathbb{Z}_p^n$. We define the oracles:*

- *$O_{\chi,\mathbf{s}}$: samples $\mathbf{a}$ uniformly in $\mathbb{Z}_p^n$, e from $\chi(\mathbb{Z}_p)$, returns $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$*
- *U: samples $\mathbf{a}$ uniformly in $\mathbb{Z}_p^n$, u uniformly in $\mathbb{Z}_p$, returns $(\mathbf{a}, u)$*

The decision LWE problem asks to distinguish $O_{\chi,\mathbf{s}}$ from U.

Note that in the case of LWE, search and decision problems are equivalent [32]. The particular case when $p = 2$ defines the *LPN problem* (for Learning Parity with Noise).

Let us also define elementary lattice problems:

Definition 4. *The lattice of basis $(\mathbf{v}_1, \cdots, \mathbf{v}_n)$ where the $\mathbf{v}_i$ are linearly independent vectors of $\mathbb{R}^n$ is defined by the set:*

$$L = \{\sum_{i=1}^{n} x_i \mathbf{v}_i \ : x_i \in \mathbb{Z}\}$$

In the following, let us denote by $\lambda_1(L)$ the length of the shortest nonzero vector of L.

Definition 5. *Given a lattice L (defined by some basis), the SVP problem asks to find a nonzero vector of shortest length in a lattice (ie such that $\|v\| = \lambda_1(L)$). The SVP_γ approximation problem asks to find a vector of L such that $\|v\| < \gamma\lambda_1(L)$.*

The decisional version of the SVP problem is known as the $GapSVP_\gamma$ problem:

Definition 6. *[26] Given a lattice L and a number d, the $GapSVP_\gamma$ problem asks to answer YES if $\lambda_1(L) \leq d$ and NO if $\lambda_1(L) > \gamma d$.*

It has been proven by Regev in [26] that if an efficient algorithm solves $LWE_{p,\chi}$ for certain distributions χ and values of p, then there exists an efficient quantum algorithm that solves $GapSVP_\gamma$ in the worst case. A similar reduction has later been found in the classical case [28]. Formalizing it would lead us too far, but it is important to note how this average case/worst case reduction is a strong security property for systems based on the LWE problem.

The LWE problem can be rewritten with a ring product instead of the dot product. In that case, we simply have one equation $as = b + e$ where s, a, b, and e all belong to a ring R. Typically, such a ring can be $R = \mathbb{Z}_p[X]/(X^n + 1)$ for some n. Thus, the need to have many a_i for a single s disappears. The problem is then called Ring-LWE, and it has been proven [30] that solving it on average is equivalent to solving the SVP on a certain class of lattices (ideal lattices). This variation is believed so far to remain a strong problem, although its additional algebraic structure has raised concerns [40].

Very recently, the use of an integer ring instead of a polynomial ring has been proposed (the modulus being a Mersenne prime) [38], but the suggested cryptosystem has quickly been broken [39].

2.2 Main idea

Lattice problems are used in a variety of ways by cryptographers, leading to signatures, encryption schemes or directly key encapsulation mechanisms. We will expose a simple version of an encryption scheme based on the plain LWE problem, proposed by Regev [26]. It is inefficient, because it encrypts bit-by-bit, but many more efficient lattice-based cryptosystems are based on similar ideas.

- Key generation: let us choose randomly $\mathbf{s} \in {\mathbb{Z}_p}^n$ and $\mathbf{A} \in {\mathbb{Z}_p}^{n \times m}$. We sample the errors $\mathbf{e}$ from a distribution χ on ${\mathbb{Z}_p}^n$ and compute the values $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$. The private key will be $\mathbf{s}$, and the public key, $(\mathbf{A}, \mathbf{b})$.
- Encryption: Bob wishes to encrypt a single bit $\mu \in \{0, 1\}$. Bob chooses a random $\mathbf{x} \in \{0, 1\}^m$, computes $\mathbf{u} = \mathbf{x}^T\mathbf{A}$ and $v = \mathbf{x}^T\mathbf{b} + \lfloor \frac{p}{2} \rceil \mu$. The ciphertext is $(\mathbf{u}, v)$.
- Decryption: Alice computes $-\langle \mathbf{u}, \mathbf{s} \rangle + v = \mathbf{x}^T\mathbf{e} + \lfloor \frac{p}{2} \rceil \mu$, which will enable her to distinguish between the two possible values of μ provided that $|\mathbf{x}^T\mathbf{e}| \leq \lfloor \frac{p}{4} \rceil$. The cryptosystem thus has a certain decryption failure probability which can be made as small as possible.

We can then build up over this idea, for example by taking elements from a ring like $R = \mathbb{Z}_p[X]/(X^n + 1)$, or from a module like R^k, which enables to encrypt longer messages for each ciphertext (see LPR cryptosystem in appendix C) . It is also possible to use deterministic noise from rounding operations instead of the χ distribution.

2.3 History

Lattice-based cryptography finds its roots in two cryptosystems, one very theoretical and unpractical one designed by Ajtai and Dwork [20], based on the Hidden Hyperplane Problem, and the heuristic and quite efficient NTRU algorithm [22], which was later found to be a Ring-LWE-based scheme. Around the same time, the GGH cryptosystem [21] [23] was proposed, based on the hardness of finding a ”good” basis for a lattice, but it was broken without leading to safer schemes. The NTRU cryptosystem has successfully resisted cryptanalysis until today, albeit with frequent modifications to its parameter set ([43] for the latest version).

The work from Ajtai and Dwork has been improved [24] and finally led to LWE-based cryptosystems [26] [25]. This brought an improvement over Ajtai and Dwork cryptosystem, bringing public key size from $O(n^4)$ to $O(n^2)$ and ciphertext size from $O(n^2)$ to $O(n)$.

Since 2010, there has been increased focus on the Ring-LWE variant, based on which Lyubashevsky, Peikert and Regev have proposed an efficient algorithm with key size in $O(n)$ and a better encryption rate [30]. This idea led to efficient key exchange protocols [31] [34] like the NewHope proposal [33], and recent proposals for the NIST standardization procedure [40] [42] [44]. This hasn't stopped work with the standard LWE problem, which led to the Frodo proposal [36].

2.4 Recent Implementations and Papers

2.4.1 Key Exchange Protocols

- **NTRUEncrypt** (1998, [22], with parameters from 2017, [43]): This algorithm was first designed heuristically. It involves polynomials with coefficients in $\{-1, 0, 1\}$ and fixed numbers of each coefficient (see appendix C for details). Its security has long been difficult to assess. A security reduction has been written for a modified version of NTRU [27], but not for the standard algorithm. An open source implementation is available at `https://github.com/NTRUOpenSourceProject/ntru-crypto` and in the Open Quantum Safe project [4], but the algorithm is patented and is owned by Onboard Security.

- **New Hope** (2015, [33]): This key-exchange protocol improves the BCNS protocol described in [34], which itself was inspired by the Peikert protocol [31]. In [31], Peikert describes a reconciliation mechanism that can be used to create a CPA-secure key encapsulation mechanism (KEM) based on the Ring-LWE problem (the proof is given in Peikert's paper). The details of Peikert's protocol can be found in appendix C. The authors of [34] give a concrete implementation of the protocol, which can be found in the Open Quantum Safe project [4]. New Hope brings some improvements, notably in the noise sampling process and the computation of parameters. The code can also be found in the OQS project [4]. This algorithm was deployed by Google in an experimental version of Chrome in 2016, and it was found to be usable in practice [37].

- **Frodo** (2016, [36]): This key encapsulation mechanism is not based on the Ring-LWE problem like all other implementations, but on the plain LWE problem. This comes from an acknowledgment of the ignorance of the community on the security of Ring-LWE. The authors claim

	Kyber[1]	New Hope[1]	Frodo[2]	NTRUEncrypt[1]
Public Key Size (bytes)	1088	1824	11296	1027
Ciphertext size (Message size) (bytes)	1184 (32)	2048 (32)	11288	1022 (32)
Key generation time (ms)	0.15	0.11	1.13	3.02
Encryption time (ms)	0.19	0.16	1.34	0.78
Decryption time (ms)	0.21	0.04	0.13	0.16
Failure probability	2^{-142}	2^{-61}	$2^{-38.9}$	2^{-195}
Security estimate (bits)	161	128	130	128

Table 2.1: Performance comparison of lattice-based encryption algorithms

that using an LWE-based cryptosystem would be more prudent. It uses a reconciliation mechanism similar to the one used in New Hope, which makes the protocol forward secure. A security reduction to the Decision LWE problem is proved. Different sets of parameters are suggested, and the protocol is implemented in the liboqs library [4], and in the associated TLS fork [5].

- **Crystals - Kyber** (2017, [40]): This is one of the two algorithms of the CRYSTALS suite. The authors propose an IND-CPA-secure encryption scheme, from which they build a CCA-secure key exchange mechanism using the Fujisaki-Okamoto transform. Their scheme uses essentially the ideas of the LPR article, although it does not reduce to the Ring-LWE problem but to the slightly more general Module-LWE problem (where product is not over a ring R, but over a module R^k, where $R = \mathbb{Z}_q[X]/(X^n + 1)$). The authors claim that this removes some of the insecurity due to the use of ideal lattices. Parameters are proposed, and an implementation of the software can be found in the liboqs library [4], the associated TLS fork [5].

- Others: the **NTRU Prime** paper [35] is worth noting, since the authors give a summary of the essential security choices in NTRU algorithms, and propose an efficient NTRU-like algorithm with another ring choice, and parameters sets. An implementation is available at `http://ntruprime.cr.yp.to/software.html`.

[1] Figures from personal tests run on an Intel Core2 Duo, with the OQS software. Failure probability and security estimates from the respective articles

[2] Figures from [40], unknown computer for benchmark. Note that this article also provides timings for Kyber, which are roughly half of those showed here. So the timings for Frodo should be doubled to be compared with the three other algorithms.

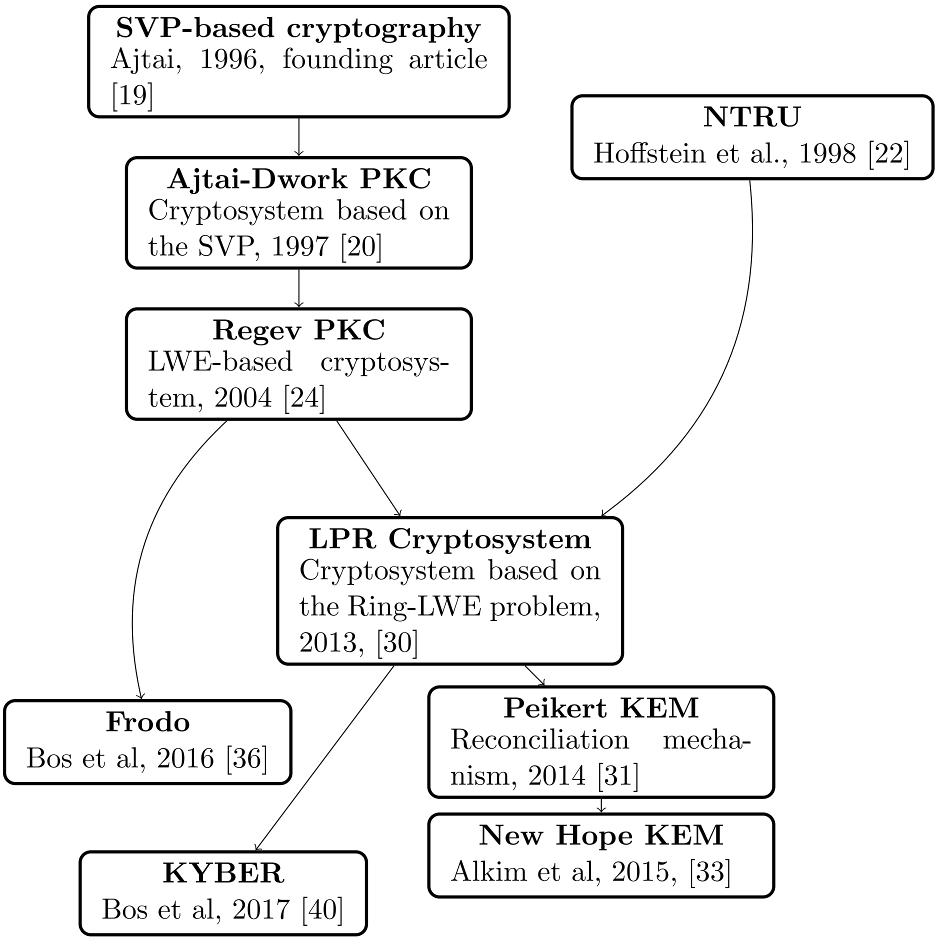

SVP-based cryptography
Ajtai, 1996, founding article [19]
NTRU
Hoffstein et al., 1998 [22]
Ajtai-Dwork PKC
Cryptosystem based on the SVP, 1997 [20]
Regev PKC
LWE-based cryptosystem, 2004 [24]
LPR Cryptosystem
Cryptosystem based on the Ring-LWE problem, 2013, [30]
Frodo
Bos et al, 2016 [36]
Peikert KEM
Reconciliation mechanism, 2014 [31]
New Hope KEM
Alkim et al, 2015, [33]
KYBER
Bos et al, 2017 [40]

Figure 2.1: Evolution of lattice-based cryptosystems

2.4.2 Signatures

- **BLISS** (2013, [29]): This signature scheme is based on a Fiat-Shamir transform of the Ring-LWE problem. An implementation is available at `http://bliss.di.ens.fr/`
- **Crystals - Dilithium** (2017, [41]): This signature scheme is build from the Module-LWE problem *via* the Fiat-Shamir transform. The authors propose parameters, but no implementation has been found. A comparison with other signature schemes can be found on table 6.1

2.5 Security assessment

As we have seen, lattice cryptosystems often have security reductions to some variant of the LWE problem, which is itself related to lattice problems. Thus, the attacks against them often require to solve either directly the LWE problem or a lattice problem.

The article [32] gives a thorough survey of cryptanalysis methods in lattice-based cryptography. We will simply recall here some points:

- it is possible to recover the secret key from an LWE instance by a lattice reduction algorithm. The preferred lattice reduction algorithm will be the BKZ algorithm, whose complexity is not well understood.
- in the case $p = 2$ (LPN problem), the BKW algorithm can recover the key in subexponential time ($2^{\mathcal{O}(n/\log n)}$).
- Other useful methods include Babai's Nearest Planes Algorithm, and the Arora-Ge method which uses Gröbner bases.
- It is not clear how using Ring-LWE instead of LWE impacts security. Some algorithms for solving problems on ideal lattices have better constants, but not a better asymptotic complexity.

2.6 Advantages and drawbacks

Based on the very rich literature and on the good performances both of lattice based key exchanges and signature schemes, one can say that lattices are one of the most serious candidates for post-quantum cryptography. Besides that, lattices have been an extremely promising source of atypical cryptographic primitices like fully homomorphic encryption, and it is probable that they will be part of cryptography's landscape for some time.

However, one has to be aware of the little understanding there is still today of the different problems underlying lattice-based cryptosystems. Indeed, even algorithms like BKZ are still very poorly understood, and the security estimates given for the cryptosystems are mostly based on experiments. It is probable that mathematical understanding of lattices will improve over the next years, and we can only hope that the cryptosystems will resist this improvements. As the authors of Frodo explain, it is possible that the most structured instances of lattice problems, like Ring-LWE, will be broken while the most general instances will resist. Thankfully, even less structured variants like Frodo are fairly competitive among all post-quantum cryptosystems (see table 6.2).

Chapter 3

Code-based cryptography

Code-based cryptography has started with the McEliece cryptosystem [45], in 1978, which has withstood cryptanalysis ever since. A lot of cryptosystems have been proposed to enhance the efficiency of the McEliece scheme, based on different types of codes, but most of them have been broken. However, recent proposals are serious candidates for a post-quantum key encapsulation mechanism.

3.1 Notions of algebraic coding theory

Before explaining the principles of code-based cryptography, let us recall a few concepts of algebraic coding theory. Let us fix a finite field $\mathbb{K} = \mathbb{F}_q$, and consider the vector space $E = \mathbb{K}^n$.

Definition 7. *A* $[n, k]$-linear code *over E is a linear subspace $\mathcal{C}$ of E of dimension k.*

Definition 8. *Given such a code, we say that a matrix $G \in \mathbb{K}^{k\times n}$ is a* generating matrix *for $\mathcal{C}$ if its rows generate $\mathcal{C}$.*

Definition 9. *We say that a matrix $H \in \mathbb{K}^{(n-k)\times n}$ is a* parity check matrix *for $\mathcal{C}$ if it is a generating matrix for $\mathcal{C}^\perp$.*

Definition 10. *The* minimal distance *of $\mathcal{C}$ is $d = \min\{wt(x) : x \in \mathcal{C} - \{0\}\}$ where $wt(\cdot)$ designates the Hamming weight. If $\mathcal{C}$ has minimal distance d, we say it is a $[n, k, d]$-linear code.*

Definition 11. *We say that $\mathcal{C}$ is t-*error correcting *if there exists a function $\mathcal{D} : E \to \mathcal{C}$ such that for all $x \in \mathcal{C}$, $e \in E$ such that $wt(e) \leq t$, then $\mathcal{D}(x + e) = x$.*

Proposition 1. *A $[n, k, d]$-linear code is t-error correcting if and only if $t < d/2$.*

3.2 Main idea and underlying problems

Most code-based cryptosystems follow the ideas developed by McEliece, *ie.* the following algorithm:

- Key Generation: Alice chooses $G \in \mathbb{F}_q^{k\times n}$, a generating matrix for an $[n,k,d]$ linear code for which an efficient decoding algorithm (we will call it $\mathcal{D}_G$) is known (for example, the original McEliece scheme uses Goppa codes). Furthermore, Alice chooses an invertible matrix $S \in \mathbb{F}_q^{k\times k}$ and a permutation matrix $P \in \mathbb{F}_q^{n\times n}$. The public key is $G^{pub} = SGP$, while the private key is (G, S, P).

- Encryption: To encrypt a message $m \in \mathbb{F}_q^k$, Bob chooses a random error vector $e \in \mathbb{F}_q^n$ such that $wt(e) < \frac{d}{2}$ (where $wt(\cdot)$ is the Hamming weight). The ciphertext is then $c = mG^{pub} + e$

- Decryption: Alice computes $m = S^{-1}\mathcal{D}_G(P^{-1}c)$

Thus the main idea is to choose a linear code with a particular structure that makes it easily invertible, and to hide that structure with the maps S and P. The security of this scheme relies on the hardness of the decoding problem for a general linear code which is known to be NP-hard, and on the assumption that G^{pub} will be indistinguishable from the generating matrix of a random linear code. This assumption is often false, and doubts have been raised even about the McEliece scheme after an algorithm has been found to distinguish G^{pub} from a random linear code generating matrix [53]. However, it is possible to enhance the hiding procedure, as has been shown by Wang in his RLCE scheme for example [55].

3.3 History

The first code-based cryptosystem has been proposed by McEliece in 1978 [45], based on Goppa codes. It had the disadvantage of needing extremely big keys. A dual variant has been proposed by Niederreiter in 1986 [46], using the parity check matrix instead of the generating matrix. The CFS signature scheme [48] is built over this variant. It achieves short signature sizes and verification times, but at the cost of a huge signature time (a few seconds).

Recently, the Moderate Density Parity Check codes have attracted attention as an alternative to the Goppa codes that enables to achieve shorter key lengths [54].

	McBits [1]	CAKE[2]
Public Key Size (Kilobytes)	311	65
Key Generation Time (ms)	169	-
Encryption Time (ms)	0.08	-
Decryption Time (ms)	0.27	-

Table 3.1: Performance comparison of code-based cryptosystems for a security estimate of 128 bits

3.4 Recent papers and implementation

- **CAKE** (2017, [56]): this paper proposes an IND-CPA secure key encapsulation mechanism (KEM), based on QC-MDPC codes (quasi-cyclic moderate density parity check codes) introduced in [54]. The advantage of those codes is that their generating matrix is already undistiguishable from random (the secret being a sparse parity check matrix), thus no additional matrices are needed. The authors finally propose an authenticated key exchange protocol
- **McBits** (2013, [52]): this paper is an implementation of the Niederreiter variant of the McEliece cryptosystem. The authors claim that the execution times of their algorithm are competitive, although the size of the keys remains quite important. The authors also propose an efficient implementation of the CFS signature scheme [48]

3.5 Security assessment

In the general case, the best attack against code-based cryptosystems is to find a short code element, using the Information Set Decoding algorithm, described by Stern in [47]. It is a probabilistic method that has successfully been used in [49] against McEliece with $n = 1024$. However, doubling the security parameter suffices to render the attack untractable.

Another line of attack against specific cryptosystems are *algebraic* (or structural) attacks: these exploit the particular structure of certain codes (like quasi-cyclic alternant codes) to reduce the key recovery problem to a set of algebraic equations that can be solved using Gröbner bases techniques. For example in [51], the authors successfully cryptanalyze variants of the McEliece cryptosystem. Code-based cryptosystems are delicate to design,

[1] Figures from personal tests run on Intel Core2 Duo with the OQS software

[2] Figures from [56]

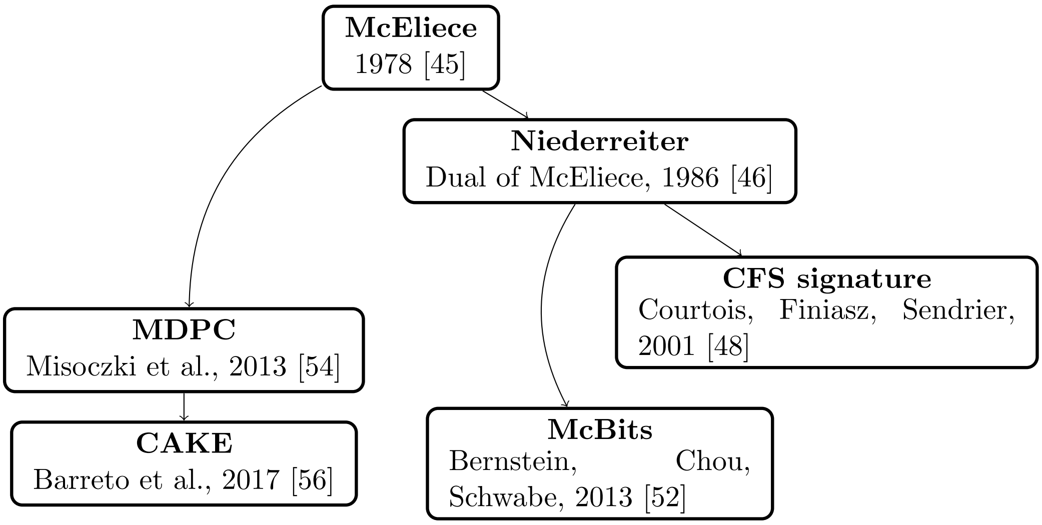

McEliece
1978 [45]
Niederreiter
Dual of McEliece, 1986 [46]
CFS signature
Courtois, Finiasz, Sendrier, 2001 [48]
MDPC
Misoczki et al., 2013 [54]
CAKE
Barreto et al., 2017 [56]
McBits
Bernstein, Chou, Schwabe, 2013 [52]

Figure 3.1: Evolution of code-based cryptography

since a lot of them have been subject to similar attacks. In [50], a review of different codes and why they might be poor choices is given.

One should also be aware that plaintext McEliece is subject to malleability, and plaintexts need a further transformation in order to render the cryptosystem CCA-secure.

3.6 Advantages and drawbacks

Code-based cryptosystems seem to have in recent years managed to overcome the curse of large keys that follows them since the McEliece cryptosystem, although they still keep the largest keys of all post-quantum key exchanges. To compensate for this disadvantage, code-based cryptosytems are probably the fastest asymmetric cryptosystems, even when compared with RSA, and with little optimization.

As a strong point one can note that the McEliece cryptosystem is the oldest cryptosystem cited in this report, and that the theory of linear codes is probably the most developed theory among those underlying post-quantum cryptosystems. One can thus be confident enough in the fact that McEliece will not be broken in the next years.

As a less enthusiastic note, one should not forget that the choice of underlying codes and of the hiding method is very fragile, as a long history of broken systems has shown. Thus, designing a code-based cryptosystem is a dangerous task. Moreover, since most practical proposals do not have proofs of security, or have their security based on *ad-hoc* problems, it is hard to assess the long-term security of most proposals.

Chapter 4

Hash-based cryptography

Hash-based cryptography differs from other post-quantum cryptosystems in an important aspect: hash-based schemes' security relies solely on the security of the underlying hash function. However, its only purpose so far has been to establish digital signatures. Hash-based digital signatures are based on two primitives: a **one-time signature scheme** (OTS) (that can be replaced by a few-times signature scheme (FTS)), and a **hash tree structure** to link many OTSs or FTSs to a single public key, like the Merkle hash tree [58].

4.1 Main idea and underlying problems

The main idea of hash-based signature schemes is to leverage one-time or few-time signature schemes in an algorithmic construction that links them to a single public verification key. This is the idea of the Merkle hash tree. Let us first understand how hash functions are used to make OTSs and FTSs, and we will then expose the Merkle tree construction.

4.1.1 One-time signature schemes

The idea of a one-time signature scheme has originated in an article from Lamport [57]. Let us understand his construction which has inspired following OTSs.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function and $g : \{0,1\}^* \to \{0,1\}^n$ be a hash function.

- Key generation: $2n$ elements from $\{0,1\}^n$ are sampled. We will note them ${x_0}^{(0)}, {x_0}^{(1)}, {x_1}^{(0)}, {x_1}^{(1)}, \ldots, {x_{n-1}}^{(0)}, {x_{n-1}}^{(1)}$. Let us note ${y_i}^{(j)} = f({x_i}^{(j)})$. The $({y_i}^{(j)})$ form the public key, while the $({x_i}^{(j)})$ form the private key.

- Signature: to sign a message m, Alice simply computes a hash $h = g(m) \in \{0,1\}^n$ of this message, then sends the set $({x_i}^{(h_i)})$ where the h_i are the bits of h.

- Verification: Bob simply has to apply the function f to the numbers he received in the signature, and verify that they are equal to $({y_i}^{(h'_i)})$ where h' is the hash of the message he received.

It is obvious that this scheme can only be used once since half of the private key is revealed in a signature. Several improvements can be made to this scheme: first the keys are quite large ($2n^2$). An idea from Winternitz is to establish a space-time tradeoff in order to get smaller keys.

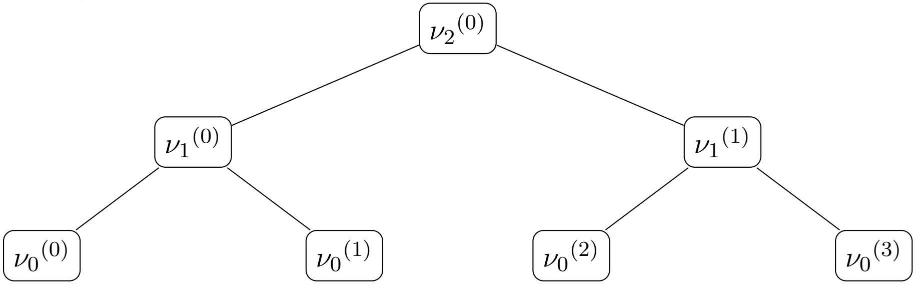
Let us finally note that the security of this scheme is equivalent to the pre-image resistance of f and the collision resistance of g [61].

4.1.2 Merkle signature scheme

The Merkle signature scheme has been first exposed in [58]. As we mentioned, the idea of this construction is to link many OTS key pairs to a single public verification key.

Let us first choose some parameters: let h be an integer, $g : \{0,1\}^* \to \{0,1\}^n$ be a hash function. We also assume that we are given an OTS scheme.

During key generation, we generate 2^h key pairs from our OTS, that we will note $(X_i, Y_i)_{0 \leq i < 2^h}$. Let us then construct a tree with the following principle:



where ${\nu_i}^{(j)} = g({\nu_{i-1}}^{(2j)} || {\nu_{i-1}}^{(2j+1)})$ for $0 < i < h$ and ${\nu_0}^{(j)} = g(Y_j)$ otherwise.

The leaves of this tree form the private key, while its root is the public key. To sign a message with this scheme, Alice has to choose a leaf which has not been used yet, and sign her message with the OTS key pair linked to this leaf. The OTS public key has to be sent along with the OTS signature, since Bob has not seen it yet. In order to verify the signature with the Merkle public key, Bob will also need to be able to compute the nodes on the path from the leaf to the root. Alice thus sends with the signature the exact minimum of nodes that Bob will need, namely the siblings of the nodes

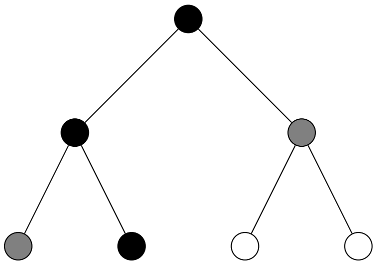


Figure 4.1: Notion of authentication path: if a message is signed with the bottommost black node, the values in the black path must be computed by the verifier. Thus, the signer has to provide the values in the grey nodes, which form the authentication path.

on the path to the root. This is called the *authentication path*, and it is the main bottleneck in the performance of this signature scheme (see figure 4.1).

Indeed, in order to compute this authentication path, either the signer keeps in memory the whole tree, which is not possible if it is large enough, or the signer has to recompute a significant number of nodes at each signature. For these reasons, a number of tree traversal algorithms have been devised, which reuse nodes from one signature to the other in order to compute authentication paths efficiently. However, the main drawback of these algorithms is that they require to maintain state, which is not always desirable in most applications. We will come back to this issue in the next section.

Finally, let us note that the security of this scheme is well understood, and is equivalent to the security of the OTS scheme used and the collision resistance of the hash function g [61].

4.1.3 Stateless hash-based signature schemes

As we have seen, Merkle-like signature schemes have to maintain a state, *a minima* to remember which leaves have already been used. If this requirement is not fullfilled, the security of the scheme breaks apart. However using a state is not something one expects from a cryptographic primitive, and it raises serious issues: for example, how to use the same key on different servers, and how to restore the state in case of power failure, or how to realize safe backups. For these reasons, stateless equivalents have been developed, most notably the SPHINCS proposal which attracted a lot of attention.

Ideally, in a stateless signature scheme, one would like to simply choose a random leaf and use it to sign a message, hoping not to fall two times on the same leaf. This idea had been brought by Goldreich. This is however highly inefficient since the security cannot be better than half the height of the tree. SPHINCS brings two improvements, which are replacing the OTS by a few-times signature scheme (FTS), and using a hyper-tree structure instead of a simple tree. Finally, the leaf FTS key that is used to sign a message is chosen by a pseudorandom function whose seed depends on the message.

An example of FTS is HORS, which is used in a modified version in SPHINCS. In HORS, we are given two integers $t >> k$, a one-way function $f : \{0,1\}^n \rightarrow \{0,1\}^n$. The signer chooses t random values x_i, and their hashes form the public key. To sign a document, he chooses k indices $i_1, \cdots, i_k$ based on a hash of the message, and sends the signature $(x_{i_1}, \cdots, x_{i_k})$. The hash function that chooses the indices must thus also obey to an interesting property called γ-subset resilience.

Definition 12. *[65] Let $\mathcal{H} = \{H_{i,t,k}\}$ be a family of functions where $H_{i,t,k}$ maps from $\{0,1\}^*$ to the set of subsets of $[0..t-1]$ of size at most k, let γ be an integer. Then we say that $\mathcal{H}$ is γ-subset resilient if for any probabilistic polynomial-time adversary, the probability of forging $(M_1, \cdots, M_{\gamma+1})$ such that $H_{i,t,k}(M_{\gamma+1}) \subseteq \bigcup_{j=1}^{\gamma} H_{i,t,k}(M_j)$ is negligible.*

To wrap it all up, an efficient stateless hash-based signature scheme such as SPHINCS works as follows:

- **Parameters**: a hash function g, an OTS, an FTS, two integers d and h, a pseudorandom generator f

- **Key Generation**: Generate a hyper-tree with d layers of Merkle trees (using hash function g) of height h, where the leaves of the first $d-1$ layers contain OTS key pairs, and the leaves of the final layer contain an FTS key pair. The roots of the trees at layer i are signed by the leaves of a tree at layer $i-1$. The secret key is the whole structure, while the public key is simply the hash value at the top of the topmost tree.

- **Signature**: choose a leaf in the final layer of trees with the pseudorandom function f. Sign the document with the associated FTS key pair. Generate the whole authentication path, up through the different layers of trees to the top. The signature is composed of the pseudorandom index computed, of the FTS signature, and of the authentication path.

- **Verification**: Using the authentication path, the verifier can compute the values on the path from the chosen leaf to the root, and check that the obtained root matches with the public key.

4.2 History

Hash-based signatures originated with the works of Lamport on OTSs [57] and of Merkle on hash tree structures [58].

A lot of work has afterwards been focused on enhancing key generation and tree traversal algorithms for Merkle schemes. In 2006, Buchmann et al. propose the CMSS algorithm [59] in which a hyper-tree structure enable asymptotic complexity improvements. Their idea, which is at the basis of follow-up work, is to use a tree of Merkle trees in which the roots of trees at level i are signed by a tree at level $i-1$. This scheme has been improved with the GMSS algorithm [60] and obtained better security proofs with the XMSS scheme [64].

There has also been work to make hash-based signatures stateless, as described in the previous section. An idea from Goldreich opened this field [63], which has led to the recent SPHINCS proposal [65]. Further work has been focused on improving the SPHINCS scheme, with for example Gravity-SPHINCS [66].

4.3 Recent papers and implementations

- **SPHINCS** (2015, [65]): This algorithm is a stateless hash-based signature scheme (as opposed to stateful classical Merkle trees). The authors introduce the HORST few-times signature scheme, and a hyper-tree optimisation of Goldreich's hash tree. The scheme is proved EUF-CMA under basic security assumptions for the building blocks (namely second-preimage resistance of the one-way function a hash function used, security of a pseudorandom number generator, security of two pseudorandom function families, and γ-subset resilience of a family of hash functions. See theorem 1 in [65] for more details). They propose a set of parameters called SPHINCS-256 which enables them to have 1KB public keys along with 41KB signatures. However, this scheme can only produce a limited number of signatures with one public key: for instance the SPHINCS-256 set of parameters is limited to 2^{50} signatures.

- **Gravity-SPHINCS** (2017, [66]): This paper gives very technical improvements to the SPHINCS scheme. We simply wish to mention here the concrete outcome of this improvements: with the set of parameters submitted to NIST, the signature size has been brought to only 26 Kbytes, the public key being of 32 bytes and it offers a signing capacity of 2^{64} messages.

4.4 Security assessment

As explained above, the security of hash-based signature schemes is very well understood, and reduces to basic properties of the hash functions and one-way functions used. However, there are constant factors in the security bounds that can be leveraged by the adversary: for example, finding a preimage of one out of T hash values is known to be a bit easier than finding a preimage of one single hash value. Other similar attacks are exposed in [65].

One should also note that in the case of stateful hash-based signatures, a failure in maintaining the state raises important security issues.

4.5 Advantages and drawbacks

Hash-based signatures are often cited as one of the most solid proposals for post-quantum signature schemes because they have been studied for a long time and their security is very well understood.

However, they come with numerous drawbacks: the most important of them, which was the presence of state, has been successfully tackled out in the recent years, at the cost of a relatively important signature cost. However the size of signatures continues to compare unfavourably to other schemes, and another issue is the complexity of the algorithms involved, that would require a thorough scrutiny against side-channel vectors of attacks.

Despite these issues, one should not forget how robust these signatures are: not only is their security well-founded, but they scale well in the security parameter, and if a security issue is discovered in one of its building blocks (hash functions, pseudorandom generators, etc.) it is very easy to change only this block. Thus, these schemes look like a good choice for the most sensitive situations, where some performance overhead would not matter much (for example signing OS updates...).

Chapter 5

Supersingular isogeny Diffie-Hellman

The idea of supersingular isogeny-based cryptography has been developed in a 2011 article by de Feo and Jao [67] (who were inspired by the work of Stolbunov). So far, it has mainly been applied to construct a Diffe-Hellman-like key exchange.

One should however note that in a very recent paper a signature scheme based on supersingular isogenies has been proposed, thanks to an Unruh's transform of the SIDH scheme [69].

5.1 Main idea

The main idea is to establish a commutative diagram like the one used in the classical Diffie-Hellman key exchange, where instead of raising a generator up to a secret power g^a we compute the quotient of an elliptic curve by a secret point $E/\langle P\rangle$.

More formally, the algorithm is roughly as follows:

- **Public parameters**: we choose a prime p of the form ${l_A}^{e_A}{l_B}^{e_B}f \pm 1$, a *supersingular* elliptic curve E_0 over $\mathbb{F}_{p^2}$, and four points P_A, Q_A, P_B, Q_B defining bases of the torsion subgroups of order ${l_A}^{e_A}$ and ${l_B}^{e_B}$ (*ie* such that $\langle P_A, Q_A\rangle = E_0[{l_A}^{e_A}]$ and $\langle P_B, Q_B\rangle = E_0[{l_B}^{e_B}]$).

- **Key Generation**: Alice chooses randomly m_A and n_A in $\mathbb{Z}/{l_A}^{e_A}\mathbb{Z}$, which define a secret point $S_A = m_A P_A + n_A Q_A$. This also defines an isogeny (*ie* a homomorphism) $\phi_A : E_0 \to E_A$ whose kernel is $\langle S_A\rangle$. She sends $(E_A, \phi_A(P_B), \phi_A(Q_B))$ to Bob.

- **Encapsulation**: Bob chooses similarly a secret point S_B and sends to Alice $(E_B, \phi_B(P_A), \phi_B(P_B))$. He also computes the secret E_{BA} as the image of an isogeny $\phi'_A : E_B \to E_{BA}$ whose kernel is $\langle m_B\phi_A(P_B) + n_B\phi_A(Q_B)\rangle$.

- **Decapsulation**: Alice computes the secret E_{AB} as the image of the isogeny whose kernel is $\langle m_A \phi_B(P_A) + n_A \phi_B(Q_A) \rangle$.

It is then non trivial but nevertheless true that E_{AB} and E_{BA} are isomorphic, and in particular share the same j-invariant. This j-invariant can then be used as a common secret, and for example be fed to a key derivation function.

Let us finally define a problem on which the key exchange protocols rely:

Definition 13. *[67] Given a tuple sampled with probability 1/2 from one of the following two distributions:*

- $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_{AB})$ *where the variables are defined as in the protocol defined above,*

- $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_C)$ *where all variables are defined as in the protocol defined above, except* E_C *which is:*

$$E_C \cong E_0 / \langle m'_A P_A + n'_A Q_A, m'_B P_B + n'_B Q_B \rangle$$

where m'_A, n'_A *(resp.* m'_B, n'_B*) are chosen uniformly at random from* $\mathbb{Z}/{l_A}^{e_A}\mathbb{Z}$ *(resp.* $\mathbb{Z}/{l_B}^{e_B}\mathbb{Z}$*) and not both divisible by* l_A *(resp.* l_B*),*

the Supersingular Decision Diffie-Hellman (SSDDH) problem asks to find from which distribution the tuple is sampled.
Given a tuple $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A))$ *sampled as in the protocol defined above, the Supersingular Computational Diffie-Hellman (SSCDH) problem asks to find the* j*-invariant of* E_{AB} *(as defined above).*

Note that the article [67] gives a thorough analysis of the security of a scheme based on these problems.

5.2 Recent papers and implementations

- **De Feo & Jao** (2011, [67]): this paper describes the first practical cryptosystem based on supersingular isogeny, and provides parameter selection and implementation results. A security reduction to the SSDDH (Supersingular Decision Diffie-Hellman) problem is provided. Their code has been inserted in the OQS project.

- **CLN** (Costello, Naehrig and Longa, 2016, [68]): this paper essentially brings technical improvements to the procedures without changing the global algorithm. Their code has also been inserted in the OQS project.

[1] Figures from personal tests run on an IntelCore 2 Duo, with the OQS software

[2] Figures from personal tests run on an IntelCore 2 Duo, with the OQS software

	FJ 2011	CLN 2016
Messages size (bytes)	1164	576
Round 1 computation time (ms)	346	274
Round 2 computation time (ms)	589	618
Round 3 computation time (ms)	281	259

Table 5.1: Performance comparison of SIDH key exchanges for a (quantum) security estimate of 128 bits [2]

Chapter 6

Global Comparison

This chapter aims at synthesizing the numerical results of previous chapters, and at giving a few guidelines on how to properly compare post-quantum cryptosystems.

The tables 6.1 and 6.2 regroup the performances of cryptosystems from every of the branches studied in this report, with classical algorithms as weel to serve as baselines. They aim at giving at the first glance an idea of the differences in terms of space and time requirements of the different algorithms. This is a fair starting point for a choice in post-quantum cryptosystems, as one may have constraints to enforce (in terms of power of computation, of throughput...).

6.1 General guidelines for the comparison of algorithms

Besides the obvious performance comparisons, one should keep in mind some general rules and pitfalls to avoid dangerous choices. This section aims at giving a few question that one should ask when studying an algorithm.

- Does the algorithm have a security reduction? Is the underlying problem well understood from a computational point of view? Are there any insights on the situation towards quantum computing and quantum complexity classes?

- Is the algorithm vulnerable to side-channel attacks? In particular, can the implementation leak timings, or intermediate values?

- Can the algorithm be backdoored? One can notably look at the concept of kleptography by Yung and Young [1].

- Can the algorithm be accelerated on dedicated hardware? In particular, can we imagine specialized chips for this algorithm?

- Is the algorithms robust to misuse (like reuse of a nonce), or to the use of a bad pseudorandom generator?

	Rainbow	Gui-96	HMFEv	Dilithium	SPHINCS-256	ECDSA P192
Public Key Size (Kbytes)	15	61.6	22.5	1.5	1	0.048
Private Key Size (Kbytes)	-	3.2	-	-	1	0.0072
Signature Size (bytes)	33	15	27	2700	41000	48
Signature Time (ms)	-	0.07	0.20	-	-	-
Signature Cost (Mcycles)	-	0.23	-	1.04	51.63	0.70
Verification Time (ms)	-	0.02	0.01	-	-	-
Verification Cost (Mcycles)	-	0.06	-	0.30	1.45	0.78
Security Estimate (bits)	80	80	80	100	128	96
Security Reduction	No	No	No	Module-LWE	Hash andone-way functions	-

Table 6.1: Global comparison of post-quantum signature schemes, for a security estimate of 80 bits[1]

[1] Sources: for Rainbow from [14] ; for Gui-96 and HFMEv from the comparison in [18] except for cycles ; for Dilithium from [41] ; for SPHINCS-256 from [65] ; and cycles for Gui-96 and ECDSA from [17].

	Kyber	New Hope	Frodo	NTRUEncrypt	McBits	CAKE	FJ11	CLN16	RSA 2048
Public Key Size (bytes)	1088	1824	11296	1027	311K	65K	1164	576	2048
Ciphertext size (Message size) (bytes)	1184 (32)	2048 (32)	11288	1022 (32)	-	65K	1164 (194)	576 (192)	2048
Key generation time (ms)	0.15	0.11	1.13	3.02	169	-	346	274	2.8
Key generation cost (Mcycles)	0.08	0.09	2.94	1.19	-	-	-	51	-
Encapsulation time (ms)	0.19	0.16	1.34	0.78	0.08	-	589	618	0.1
Encapsulation cost (Mcycles)	0.12	0.11	3.49	0.06	-	-	-	123	-
Decapsulation time (ms)	0.21	0.04	0.13	0.16	0.27	-	281	259	0.1
Decapsulation cost (Mcycles)	0.13	0.02	0.34	0.11	-	-	-	57	-
Security estimate (bits)	161	128	130	128	128	128	128	128	-
Formal Security Reduction	Module-LWE	R-LWE	LWE	No	ad-hoc	ad-hoc	SSDDH	SSDDH	-

Table 6.2: Global comparison of post-quantum key encapsulation mechanisms [2]

[2] Sources: for Kyber, New Hope, NTRUEncrypt, McBits, FJ11 and CLN16 figures come from personal test with the OQS software [4] (ran on Intel Core 2 Duo, 2,4GHz), except for the number of cycles. The number of cycles for lattice systems are given for AVX2 optimized implementations and come from [40]. Figures for Frodo also come from this article. Speeds for RSA mesured with `openssl speed`

Conclusion

We have seen in this work five well developed branches of post-quantum cryptography, which will all likely be represented at the NIST standardisation process, and which all have a decent chance of being standardized. Let us not forget that only a tiny part of existing post-quantum cryptosystems could be mentioned here, and that some outsider algorithms do not even fit into the categories that form the chapters of this report (for example, the recent Picnic signature scheme, based on sigma protocols and symmetric cryptography, is a promising alternative to the branches we studied). Moreover, it is a field that evolves amazingly fast, and no doubt that most of this report will be archaic very soon.

Although each of these technologies come with their advantages and weaknesses, the author has been more particularly fascinated by the field of lattice-based cryptography, which not only offers algorithms with very good performances, but also an impressive richness both of mathematical structures and computational problems. It would seem probable that at least some lattice-based algorithms will make it to the final stages of the NIST procedure, notably for KEMs. As for signatures, the stateless hash-based schemes also seem to be very promising technology, despite their important signature sizes, and it will probably have interesting use cases. Finally, one can hope that the fields of lattice and code-based cryptography will benefit from a deeper study of their structures that would give a more profound understanding of the cryptosystems and the attacks.

Bibliography

General

[1] Adam Young and Moti Yung. “Kleptography: Using Cryptography Against Cryptography”. In: *Advances in Cryptology - EUROCRYPT ’97.* Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, May 11, 1997, pp. 62–74. DOI: `10.1007/3-540-69053-0_6`.

[2] R. Cramer and V. Shoup. “Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack”. In: *SIAM Journal on Computing* 33.1 (Jan. 1, 2003), pp. 167–226. DOI: `10.1137/S0097539702403773`.

[3] Serge Vaudenay. *Cryptography and Security, Lecture Notes.* 2017.

Software

[4] *liboqs: C library for quantum-resistant cryptographic algorithms.* URL: `https://github.com/open-quantum-safe/liboqs` (visited on 11/11/2017).

[5] *openssl: Fork of OpenSSL that includes quantum-resistant algorithms and ciphersuites based on liboqs.* URL: `https://github.com/open-quantum-safe/openssl` (visited on 11/11/2017).

Multivariate Cryptography

[6] Tsutomu Matsumoto and Hideki Imai. “Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption”. In: *Advances in Cryptology - EUROCRYPT ’88.* Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, May 25, 1988, pp. 419–453. DOI: `10.1007/3-540-45961-8_39`.

[7] Jacques Patarin. “Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms”. In: *Advances in Cryptology - EUROCRYPT ’96.* Springer, Berlin, Heidelberg, May 12, 1996, pp. 33–48. DOI: `10.1007/3-540-68339-9_4`.

[8] Jacques Patarin. “The oil and vinegar algorithm for signatures”. In: Dagstuhl Workshop on Cryptography. 1997.

[9] Aviad Kipnis, Jacques Patarin, and Louis Goubin. “Unbalanced Oil and Vinegar Signature Schemes”. In: *Advances in Cryptology - EUROCRYPT ’99*. Springer, Berlin, Heidelberg, May 2, 1999, pp. 206–222. DOI: 10.1007/3-540-48910-X_15.

[10] Aviad Kipnis and Adi Shamir. “Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization”. In: *Advances in Cryptology - CRYPTO’ 99*. Springer, Berlin, Heidelberg, Aug. 15, 1999, pp. 19–30. DOI: 10.1007/3-540-48405-1_2.

[11] Jacques Patarin, Nicolas Courtois, and Louis Goubin. “FLASH, a Fast Multivariate Signature Algorithm”. In: *Topics in Cryptology - CT-RSA 2001*. Cryptographers’ Track at the RSA Conference. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Apr. 8, 2001, pp. 298–307. DOI: 10.1007/3-540-45353-9_22.

[12] Jacques Patarin, Nicolas Courtois, and Louis Goubin. “QUARTZ, 128-Bit Long Digital Signatures”. In: *Topics in Cryptology - CT-RSA 2001*. Springer, Berlin, Heidelberg, Apr. 8, 2001, pp. 282–297. DOI: 10.1007/3-540-45353-9_21.

[13] Nicolas T. Courtois, Magnus Daum, and Patrick Felke. “On the Security of HFE, HFEv- and Quartz”. In: *Public Key Cryptography - PKC 2003*. Springer, Berlin, Heidelberg, Jan. 6, 2003, pp. 337–350. DOI: 10.1007/3-540-36288-6_25.

[14] Jintai Ding and Dieter Schmidt. “Rainbow, a new multivariable polynomial signature scheme”. In: ACNS. Vol. 5. Springer, 2005, pp. 164–175.

[15] Chia-Hsin Owen Chen et al. *Odd-Char Multivariate Hidden Field Equations*. 543. 2008. URL: https://eprint.iacr.org/2008/543.

[16] Jintai Ding and Bo-Yin Yang. “Degree of Regularity for HFEv and HFEv-”. In: *Post-Quantum Cryptography*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, June 4, 2013, pp. 52–66. DOI: 10.1007/978-3-642-38616-9_4. (Visited on 12/03/2017).

[17] Albrecht Petzoldt et al. “Design Principles for HFEv- Based Multivariate Signature Schemes”. In: *Advances in Cryptology – ASIACRYPT 2015*. Springer, Berlin, Heidelberg, Nov. 29, 2015, pp. 311–334. DOI: 10.1007/978-3-662-48797-6_14.

[18] Albrecht Petzoldt et al. “HMFEv - An Efficient Multivariate Signature Scheme”. In: *Post-Quantum Cryptography*. Springer, Cham, June 26, 2017, pp. 205–223. DOI: 10.1007/978-3-319-59879-6_12.

Lattice-based Cryptography

[19] M. Ajtai. “Generating Hard Instances of Lattice Problems (Extended Abstract)”. In: *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing.* STOC ’96. New York, NY, USA: ACM, 1996, pp. 99–108. DOI: 10.1145/237814.237838. (Visited on 10/26/2017).

[20] Miklós Ajtai and Cynthia Dwork. “A Public-key Cryptosystem with Worst-case/Average-case Equivalence”. In: *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing.* STOC ’97. New York, NY, USA: ACM, 1997, pp. 284–293. DOI: 10.1145/258533.258604. URL: http://doi.acm.org/10.1145/258533.258604.

[21] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. “Public-key cryptosystems from lattice reduction problems”. In: *Advances in Cryptology - CRYPTO ’97.* Springer, Berlin, Heidelberg, Aug. 17, 1997, pp. 112–131. DOI: 10.1007/BFb0052231.

[22] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. “NTRU: A ring-based public key cryptosystem”. In: *Algorithmic Number Theory: Third International Symposiun, ANTS-III Portland, Oregon, USA, June 21-25, 1998 Proceedings.* Springer Berlin Heidelberg, 1998, pp. 267–288. URL: https://doi.org/10.1007/BFb0054868.

[23] Daniele Micciancio. “Improving Lattice Based Cryptosystems Using the Hermite Normal Form”. In: *Cryptography and Lattices.* Springer, Berlin, Heidelberg, 2001, pp. 126–145. URL: https://link.springer.com/chapter/10.1007/3-540-44670-2_11.

[24] Oded Regev. “New Lattice-based Cryptographic Constructions”. In: *J. ACM* 51.6 (Nov. 2004), pp. 899–942. DOI: 10.1145/1039488.1039490.

[25] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. “A Framework for Efficient and Composable Oblivious Transfer”. In: *Advances in Cryptology - CRYPTO 2008.* Springer, Berlin, Heidelberg, Aug. 17, 2008, pp. 554–571. DOI: 10.1007/978-3-540-85174-5_31.

[26] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Journal of the ACM (JACM)* 56.6 (2009), p. 34. ISSN: 0004-5411.

[27] Damien Stehlé and Ron Steinfeld. “Making NTRU as Secure as Worst-Case Problems over Ideal Lattices”. In: *Advances in Cryptology - EUROCRYPT 2011.* Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, May 15, 2011, pp. 27–47. DOI: 10.1007/978-3-642-20465-4_4.

[28] Zvika Brakerski et al. "Classical Hardness of Learning with Errors". In: *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*. ACM, 2013, pp. 575–584. DOI: 10.1145/2488608.2488680.

[29] Léo Ducas et al. "Lattice Signatures and Bimodal Gaussians". In: *Advances in Cryptology - CRYPTO 2013*. Springer, Berlin, Heidelberg, 2013, pp. 40–56. URL: https://link.springer.com/chapter/10.1007/978-3-642-40041-4_3.

[30] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *J. ACM* 60.6 (Nov. 2013), 43:1–43:35. DOI: 10.1145/2535925.

[31] Chris Peikert. "Lattice Cryptography for the Internet". In: *Post-Quantum Cryptography*. Lecture Notes in Computer Science. Springer, Cham, Oct. 1, 2014, pp. 197–219. DOI: 10.1007/978-3-319-11659-4_12.

[32] Martin R. Albrecht, Rachel Player, and Sam Scott. *On the concrete hardness of Learning with Errors*. 046. 2015. URL: https://eprint.iacr.org/2015/046.

[33] Erdem Alkim et al. *Post-quantum key exchange - a new hope*. 1092. 2015. URL: https://eprint.iacr.org/2015/1092.

[34] J. W. Bos et al. "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem". In: *2015 IEEE Symposium on Security and Privacy*. May 2015, pp. 553–570. DOI: 10.1109/SP.2015.40.

[35] Daniel J. Bernstein et al. *NTRU Prime: reducing attack surface at low cost*. 461. 2016. URL: https://eprint.iacr.org/2016/461 (visited on 11/12/2017).

[36] Joppe Bos et al. "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. ACM, 2016, pp. 1006–1018. DOI: 10.1145/2976749.2978425.

[37] Adam Langley. *ImperialViolet - CECPQ1 results*. Nov. 28, 2016. URL: https://www.imperialviolet.org/2016/11/28/cecpq1.html (visited on 11/12/2017).

[38] Divesh Aggarwal et al. *A New Public-Key Cryptosystem via Mersenne Numbers*. 481. 2017. URL: https://eprint.iacr.org/2017/481.

[39] Marc Beunardeau et al. *On the Hardness of the Mersenne Low Hamming Ratio Assumption*. 522. 2017. URL: https://eprint.iacr.org/2017/522.

[40] Joppe Bos et al. *CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM*. 634. 2017. URL: https://eprint.iacr.org/2017/634.

[41] Leo Ducas et al. *CRYSTALS – Dilithium: Digital Signatures from Module Lattices.* 633. 2017. URL: `https://eprint.iacr.org/2017/633`.

[42] Mike Hamburg. "Post-quantum cryptography proposal: ThreeBears (draft)". In: (2017).

[43] Jeff Hoffstein et al. "Choosing Parameters for NTRUEncrypt". In: *Topics in Cryptology - CT-RSA 2017.* Springer, Cham, Feb. 14, 2017, pp. 3–18. DOI: `10.1007/978-3-319-52153-4_1`.

[44] Markku-Juhani O. Saarinen. *HILA5: On Reliability, Reconciliation, and Error Correction for Ring-LWE Encryption.* 424. 2017. URL: `https://eprint.iacr.org/2017/424`.

Code-based Cryptography

[45] Robert J McEliece. "A public-key cryptosystem based on algebraic coding theory". In: (1978).

[46] H Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory". In: *Prob. Contr. Inform. Theory* 15.2 (1986).

[47] Jacques Stern. "A method for finding codewords of small weight". In: *Coding Theory and Applications.* International Colloquium on Coding Theory and Applications. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Nov. 2, 1988, pp. 106–113. DOI: `10.1007/BFb0019850`.

[48] Nicolas T. Courtois, Matthieu Finiasz, and Nicolas Sendrier. "How to Achieve a McEliece-Based Digital Signature Scheme". In: *Advances in Cryptology - ASIACRYPT 2001.* Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Dec. 9, 2001, pp. 157–174. DOI: `10.1007/3-540-45682-1_10`.

[49] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. *Attacking and defending the McEliece cryptosystem.* 318. 2008. URL: `https://eprint.iacr.org/2008/318`.

[50] Raphael Overbeck and Nicolas Sendrier. "Code-based Cryptography". In: *Post-Quantum Cryptography.* Springer Berlin Heidelberg, 2009, pp. 95–146.

[51] Jean-Charles Faugere et al. "Algebraic Cryptanalysis of McEliece Variants with Compact Keys." In: *Eurocrypt.* Vol. 6110. Springer. 2010, pp. 279–298.

[52] Daniel J. Bernstein, Tung Chou, and Peter Schwabe. “McBits: Fast Constant-Time Code-Based Cryptography”. In: *Cryptographic Hardware and Embedded Systems - CHES 2013*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Aug. 20, 2013, pp. 250–272. DOI: 10.1007/978-3-642-40349-1_15.

[53] J. C. Faugère et al. “A Distinguisher for High-Rate McEliece Cryptosystems”. In: *IEEE Transactions on Information Theory* 59.10 (Oct. 2013), pp. 6830–6844. DOI: 10.1109/TIT.2013.2272036.

[54] R. Misoczki et al. “MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes”. In: *2013 IEEE International Symposium on Information Theory*. July 2013, pp. 2069–2073. DOI: 10.1109/ISIT.2013.6620590.

[55] Yongge Wang. “Quantum resistant random linear code based public key encryption scheme RLCE”. In: Information Theory (ISIT), 2016 IEEE International Symposium on. IEEE, 2016, pp. 2519–2523.

[56] Paulo S. L. M. Barreto et al. *CAKE: Code-based Algorithm for Key Encapsulation*. 757. 2017. URL: https://eprint.iacr.org/2017/757.

Hash-based Cryptography

[57] Leslie Lamport. *Constructing digital signatures from a one-way function.* Technical Report CSL-98, SRI International Palo Alto, 1979.

[58] Ralph C. Merkle. “A Certified Digital Signature”. In: *Advances in Cryptology - CRYPTO’ 89 Proceedings*. Springer, New York, NY, Aug. 20, 1989, pp. 218–238. DOI: 10.1007/0-387-34805-0_21.

[59] Johannes Buchmann et al. “CMSS - An Improved Merkle Signature Scheme”. In: *Progress in Cryptology - INDOCRYPT 2006*. Springer, Berlin, Heidelberg, Dec. 11, 2006, pp. 349–363. DOI: 10.1007/11941378_25.

[60] Johannes Buchmann et al. “Merkle Signatures with Virtually Unlimited Signature Capacity”. In: *Applied Cryptography and Network Security*. Springer, Berlin, Heidelberg, 2007, pp. 31–45. URL: https://link.springer.com/chapter/10.1007/978-3-540-72738-5_3.

[61] Johannes Buchmann, Erik Dahmen, and Michael Szydlo. “Hash-based Digital Signature Schemes”. In: *Post-quantum Cryptography*. Springer Berlin Heidelberg, 2009, pp. 35–94.

[62] Jintai Ding and Bo Yin-Yang. “Multivariate Public-Key Cryptography”. In: *Post-Quantum Cryptography*. Springer Berlin Heidelberg, 2009, pp. 193–242.

[63] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications.* Cambridge university press, 2009.

[64] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. “XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions”. In: *Post-Quantum Cryptography.* Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Nov. 29, 2011, pp. 117–129. DOI: `10.1007/978-3-642-25405-5_8`.

[65] Daniel J. Bernstein et al. “SPHINCS: Practical Stateless Hash-Based Signatures”. In: *Advances in Cryptology – EUROCRYPT 2015.* Springer, Berlin, Heidelberg, Apr. 26, 2015, pp. 368–397. DOI: `10.1007/978-3-662-46800-5_15`.

[66] Jean-Philippe Aumasson and Guillaume Endignoux. *Improving Stateless Hash-Based Signatures.* 933. 2017. URL: `https://eprint.iacr.org/2017/933` (visited on 12/10/2017).

Supersingular Isogeny Diffie-Hellman

[67] David Jao and Luca De Feo. “Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies”. In: *Post-Quantum Cryptography.* Springer, Berlin, Heidelberg, Nov. 29, 2011, pp. 19–34. DOI: `10.1007/978-3-642-25405-5_2`.

[68] Craig Costello, Patrick Longa, and Michael Naehrig. *Efficient algorithms for supersingular isogeny Diffie-Hellman.* 413. 2016. URL: `https://eprint.iacr.org/2016/413`.

[69] Youngho Yoo et al. *A Post-Quantum Digital Signature Scheme Based on Supersingular Isogenies.* 186. 2017. URL: `https://eprint.iacr.org/2017/186`.

Appendix A

Cryptographic concepts and definitions

In order to avoid misinterpretations, we will define in this appendix all the relevant cryptographic concepts.

Let us start with the *public-key cryptosystem*:

Definition 14. *[3] A* public-key cryptosystem *is a tuple* $(\mathtt{Gen}, \mathcal{M}, \mathtt{Enc}, \mathtt{Dec})$ *where :*

- $\mathcal{M}$ *is a plaintext domain,*
- $\mathtt{Gen}$ *is a probabilitic algorithm that outputs a public key/secret key pair* (pk, sk)*,*
- $\mathtt{Enc}$ *is a probabilistic algorithm that on input* (pk, m) *where* $m \in \mathcal{M}$ *outputs a ciphertext c,*
- $\mathtt{Dec}$ *is a deterministic algorithm that on input* (sk, c) *outputs either an element of* $\mathcal{M}$ *or an error symbol* $\perp$*,*

and such that:

$$\forall m \in \mathcal{M}, \Pr[\mathtt{Dec}(sk, \mathtt{Enc}(pk, m)) = m] = 1$$

where $(pk, sk) = \mathtt{Gen}()$ *and the probability is over the randomness used in* $\mathtt{Gen}$ *and* $\mathtt{Enc}$

In our line of work, we will however extend the definition to include algorithms such that this probability is very close to 1.

However, using exclusively a public-key cryptosystem to establish a secure channel of communication would be too costly. Most cryptographic protocols (like SSL/TLS) work by using public-key cryptography to establish a common secret over an insecure channel, and then use the common secret in a symmetric encryption scheme. This is called *hybrid encryption.* The primitive that enables the establishment of a common secret is called a *key encapsulation mechanism.*

Definition 15. *[2] [56] A* key encapsulation mechanism *is a tuple* $(\texttt{KeyGen}, \texttt{Encaps}, \texttt{Decaps})$ *where:*

- $\texttt{KeyGen}$ *is a probabilistic algorithm that outputs a public key/secret key pair* (pk, sk)*,*
- $\texttt{Encaps}$ *is a probabilistic algorithm that on input* pk *outputs a ciphertext* c *and a symmetric key* K*.*
- $\texttt{Decaps}$ *is a deterministic algorithm that on input* (sk, c) *outputs either a symmetric key* K *or a failure symbol* $\perp$*.*

and such that if $\texttt{Gen}() = (pk, sk)$,$\texttt{Encaps}(pk) = (c, K)$*,* $\texttt{Decaps}(sk, c) = \hat{K}$*, then* $\hat{K} = K$ *with probability 1, where the probability is over the randomness used by* $\texttt{Gen}$ *and* $\texttt{Encaps}$*.*

An important property of hybrid cryptosystems is that their security relies solely on the security of the KEM and of the symmetric cryptosystem that compose it (proven by Cramer and Shoup in [2]).

Let us now define the notion of signature scheme:

Definition 16. *[3] A* digital signature scheme *is a tuple* $(\texttt{Gen}, \mathcal{D}, \texttt{Sig}, \texttt{Ver})$ *where:*

- $\mathcal{D}$ *is a message domain,*
- $\texttt{Gen}$ *is a probabilistic algorithm outputting a public key/secret key pair* (pk, sk)*,*
- $\texttt{Sig}$ *is a probabilistic algorithm that on input* (sk, m) *where* $m \in \mathcal{D}$ *outputs a signature* s*,*
- $\texttt{Ver}$ *is a deterministic algorithm that on input* (pk, s) *outputs 0 (rejects) or 1 (accepts)*

and such that if $(pk, sk) = \texttt{Gen}()$*:*

$$\forall m \in \mathcal{D}, \Pr[\texttt{Ver}(pk, \texttt{Sig}(sk, m)) = 1] = 1$$

where the probability is over the randomness used in $\texttt{Gen}$ *and* $\texttt{Sig}$*.*

Appendix B

Mathematical details of multivariate signature schemes

Let us in this appendix formalize the two main studied multivariate signature schemes: the Oil & Vinegar scheme, and the HFE scheme.

B.1 Oil & Vinegar scheme

The parameters of this scheme are a finite field $\mathbb{F}_q$, $o, v \in \mathbb{N}$. We define $V = [1..v]$ and $O = [v+1..n = o+v]$.

To generate a key, we choose a map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^o$ defined by o polynomials

$$p_k = \sum_{i,j \in V} a_{ij}^{(k)} x_i x_j + \sum_{i \in V, j \in O} b_{ij}^{(k)} x_i x_j + \sum_{i \in V \cup O} c_i^{(k)} x_i + d^{(k)}$$

Thus this map is linear in o variables and quadratic in v other variables. It is easy to find $\mathbf{x}$ such that $\mathcal{P}(\mathbf{x}) = 0$ by randomly fixing $x_1, \cdots, x_v$ and then solving the o remaining linear equations.

This map is then hidden by affine transformations as explained in section 1.1.

B.2 HFE scheme

The acronym HFE stands for Hidden Field Equations, the main idea of this scheme being to use a map which is easy to invert in an extension field of $\mathbb{F}_q$.

The parameters of this scheme are a finite field $\mathbb{F}_q$, a finite extension of this field $\mathbb{K} = \mathbb{F}_{q^n}$ and $d \in \mathbb{N}$.

To generate a key, we define a map $\bar{\mathcal{P}}$ over the extension field $\mathbb{K}$ of the form:

$$\bar{\mathcal{P}}(X) = \sum^{q^i+q^j \leq d} a_{ij} X^{q^i+q^j} + \sum^{q^i \leq d} b_i x^{q^i} + c$$

Since $x \mapsto x^q$ is a linear map in $\mathbb{K}$, $\mathcal{P}$ is a quadratic polynmial. Furthermore, there is an algorithm (the Berlekamp algorithm) that makes $\bar{\mathcal{P}}$ easy to invert.

If we call ϕ the canonical embedding $\phi : \mathbb{F}_q^n \to \mathbb{K}$, we have that $\mathcal{P} = \phi^{-1} \circ \bar{\mathcal{P}} \circ \phi$ is a quadratic map of $\mathbb{F}_q^n$. The public key is then built as explained in section 1.1.

The $v-$ variant of this scheme is then built as follows:

- for the vinegar variant, we replace the b_i by a linear function $b_i(V_1, \cdots, V_v)$ and c by a quadratic function $c_i(V_1, \cdots, V_v)$ where the V_i are additional variables. To invert the function, we simply try random values for the V_i until a solution exists.
- for the minus variant, we simply remove a equations from the signature.

Thus, the final signature scheme is a quadratic map $\mathcal{S} : \mathbb{F}_q^{n+v} \to \mathbb{F}_q^{n-a}$.

Appendix C

A closer look at three lattice-based cryptosystems

As an additional material, we wish to describe in more detail three promising PKEs and KEMs based on lattices. These will all be linked to the Ring variant of the LWE problem.

C.1 The LPR cryptosystem

This PKE has been merely outlined in the paper introducing the Ring-LWE problem for cryptography [30]. It follows the outline of the Regev cryptosystem (see section 2.2), but using elements over a ring. It has led to very concrete implementations like Kyber.

- **Parameters**: let n, q be fixed integers, $R = \mathbb{Z}_q[X]/(X^n + 1)$.
- **Key Generation**: $a \in R$ is sampled uniformly, $s, e \in R$ are sampled according to some distribution χ, $b := as + e$. The public key is (a, b), the private key is s.
- **Encryption**: the input is $m \in \{0, 1\}^n$. Let $r, e_1, e_2 \in R$ be sampled according to χ. The encrypter computes:

  $$u := ar + e_1 \quad \text{and} \quad v := br + e_2 + \lfloor \frac{q}{2} \rceil m$$

- **Decryption**: The decrypter computes

  $$m' = v - us = \lfloor \frac{q}{2} \rceil m + \epsilon$$

  where $\epsilon = re - se_1 - e_2 < q/4$ with high probability (depends on the choice of χ). Thus the decrypter simply rounds each coefficient of m' to the closest value between 0 and $\lfloor \frac{q}{2} \rceil$ to recover m.

C.2 Peikert's KEM

In [31], Chris Peikert describes a key exchange mechanism that is based not on an encryption scheme but on a reconciliation mechanism. This mechanism has been reused in other proposals like New Hope and Frodo. It uses smart rounding and cross-rounding functions, that we will describe first. Let us work in $\mathbb{Z}_q$. We define for $v \in \mathbb{Z}_q$:

- $\lfloor v \rceil_2 = \lfloor \frac{2}{q} v \rceil$
- $\langle v \rangle_2 = \lfloor \frac{4}{q} v \rceil \bmod 2$

and finally for $w \in \mathbb{Z}_q$, we define the function *rec* by:

$$rec(w, 0) = \begin{cases} 0 & \text{if } w \in [-\frac{q}{8}, \frac{3q}{8}) \\ 1 & \text{if } w \in [\frac{3q}{8}, \frac{7q}{8}) \end{cases}$$

and

$$rec(w, 1) = \begin{cases} 0 & \text{if } w \in [-\frac{3q}{8}, \frac{q}{8}) \\ 1 & \text{if } w \in [\frac{q}{8}, \frac{5q}{8}) \end{cases}$$

This is better understood with a drawing, as in Peikert's paper (see figure C.1).

Thus we have the following property:

Proposition 2. *if* $w = v + e \bmod q$ *and* $|e| < q/8$ *(where* $|e|$ *matches the absolute value in* $\mathbb{Z}$ *if we embed* $\mathbb{Z}/q\mathbb{Z}$ *in* $[-q/2..q/2[$*) then* $rec(w, \langle v \rangle_2) = \langle v \rangle_2$.

Let us now describe the KEM:

- **Parameters**: same as LPR cryptosystem
- **Key Generation**: same as LPR cryptosystem
- **Encapsulation**: let $r, e_1, e_2 \in R$ be sampled according to χ,

  $$u := ar + e_1 \quad \text{and} \quad v := br + e_2 m$$

  Bob sends to Alice $(u, \langle v \rangle_2)$ and computes its key $K := \lfloor v \rceil_2$
- **Decapsulation**: Alice computes $K' = rec(us, \langle v \rangle_2)$ which will be equal to K.

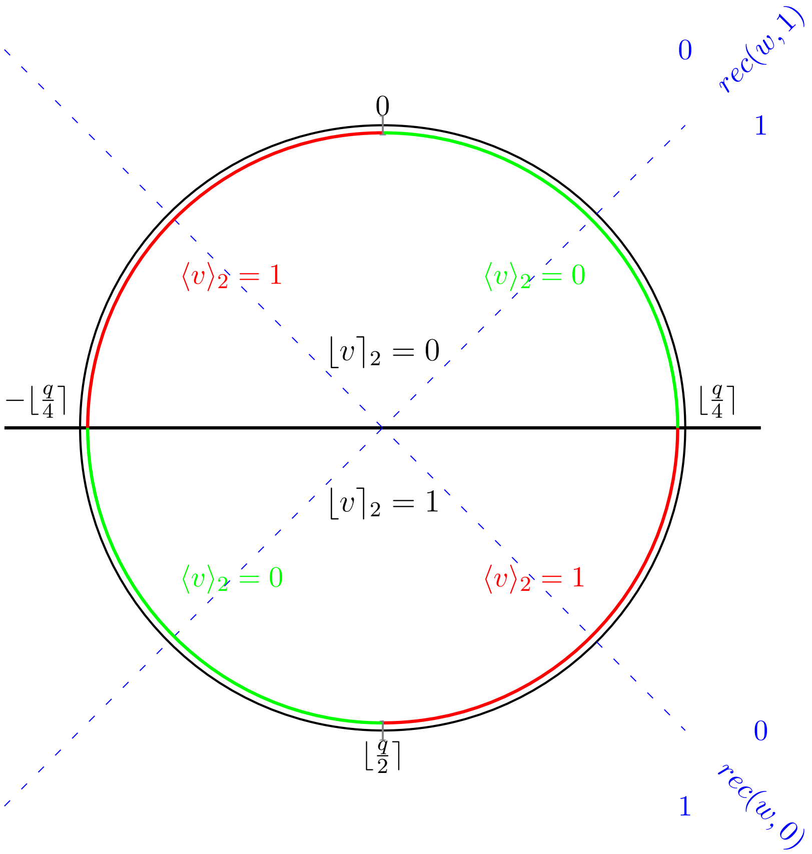


Figure C.1: Description of Peikert's reconciliation mechanism over a wheel representation of $\mathbb{Z}_q$

C.3 NTRU cryptosystem

For this cryptosystem, the dependance on the LWE problem, or even on a lattice problem, is not immediately apparent. It relies primarily on the fact that on a ring $\mathbb{Z}_q[X]/(X^N - 1)$, given two "small" polynomials f and g (e.g. polynomials having a lot of 0 coefficients, and whose remaining coefficients are small), it is hard to distinguish the distribution of $h = g/f$ from the uniform distribution over R_q. This problem can be linked to the problem of finding small vectors in a lattice defined by the cyclic shifts of h (see the security assessment in the original article for more precisions).

- **Parameters**: Two prime numbers p and q (usually p will be small, e.g. 3), an integer N. Let $R = \mathbb{Z}[X]/(X^N - 1)$, $R_q = \mathbb{Z}_q[X]/(X^N - 1)$, $R_p = \mathbb{Z}_p[X]/(X^N - 1)$.
- **Key Generation**: Let $f, g \in R$ be two random small (in a sense to be defined) polynomials, such that f is invertible in R_q and in R_p (where we consider f as an element of R_q by reducing each of its coefficients modulo q, and id. for R_p). Let F_p (resp. F_q) be the inverse of f in R_p (resp. R_q). Let:
$$h := F_q \cdot g \in R_q$$
The public key is h, the private key is f.
- **Encryption**: the input has to be mapped to a polynomial $m \in R_q$. Let $\phi \in R_q$ be a random small polynomial. The ciphertext is:
$$c := p\phi \cdot h + m$$
- **Decryption**: let :
$$a := f \cdot c \pmod{p} \in R_p$$
$$m' := F_p \cdot a$$