



Università di Genova

Dipartimento di Matematica

PhD Program in Mathematics and Applications
Curriculum Mathematical Methods for Data Analysis

Cycle XXXVIII

PhD Thesis

Lattices, Isogenies and Adaptor Signatures in Post-Quantum Cryptography

Andrea Sanguineti

Dr. Rer. Nat. Alessio Caminata
Prof. Emanuela De Negri

Contents

1	Introduction	3
1.1	Hard Problems in Classical Cryptography	4
1.2	NIST Post-Quantum Cryptography Standardization and Post-Quantum Areas	5
1.3	Outline of the Thesis	6
1.4	Principal results	6
1.5	Notations	7
2	Cryptographic Preliminaries	9
2.1	Cryptosystems and Digital Signatures	9
2.2	Identification schemes and the Fiat-Shamir Transformation	14
2.3	Key Encapsulation Mechanisms and Diffie-Hellman Key Exchange	19
2.4	Identity-based Broadcast Encryption schemes (IBBE)	21
2.5	Hierarchical Identity-based Broadcast Encryption schemes (HIBBE)	25
2.6	Adaptor Signatures	26
3	Lattices	34
3.1	Definitions and Properties	34
3.2	Principal Lattice Problems and Babai's Algorithm	39
3.3	Gauss Reduction Algorithm and LLL Reduction Algorithm	42
3.4	Nearest Plane Algorithm and Micciancio-Goldwasser Algorithm	48
3.5	Discrete Gaussian Distribution and Sampling over lattices	54
3.6	Learning With Errors problem	57
3.7	Two Lattice-based Cryptographic Constructions: GGH and CRYSTALS - Dilithium	60
4	Elliptic Curves and Isogenies	68
4.1	Definition and Properties	68
4.2	The Endomorphism Ring	74
4.3	Elliptic Curves over Finite Fields	76
4.4	Elliptic Curves over $\mathbb{C}$	77
4.5	Orders in Imaginary Quadratic Fields and the Ideal Class Group	79
4.6	The Deuring Lifting Theorem and Isogeny graphs	86
4.7	An Isogeny-based Cryptographic Construction: CSIDH	90
5	Overview of a Post-Quantum Lattice-based HIBBE scheme	93
5.1	Additional Lattice Algorithms for the scheme	94
5.2	Construction of the Hash function	98
5.3	Description of the HIBBE scheme	99

6	Post-Quantum Adaptor Signatures from Cryptographic Group Actions	102
6.1	Sigma Protocols from Group Actions and CSI-FiSh	103
6.2	ORCAS	107
6.3	Properties of Base ORCAS for Filtered CSI-FiSh	109

Chapter 1

Introduction

As the applications of the internet continue to grow worldwide and become increasingly integrated into everyday life, protecting sensitive information and ensuring the privacy of digital communications have become essential priorities. Modern society relies heavily on online services for banking, healthcare, education, commerce, social interaction, and governmental processes. As a consequence, big amounts of personal and confidential data are transmitted and stored electronically every day. This widespread dependence on interconnected systems has also increased the opportunities for malicious actors to intercept, manipulate, or exploit information. Therefore, securing digital communication channels against unauthorized access and guaranteeing the integrity and authenticity of exchanged information are fundamental challenges of today.

Cryptography plays a central role in addressing these challenges by providing mathematical tools and protocols designed to protect information. Through encryption, cryptographic schemes allow users to exchange messages while preventing unauthorized parties from learning their contents. Digital signatures enable verification of authenticity and integrity, ensuring that messages originate from legitimate senders and have not been modified. More generally, cryptography aims to preserve confidentiality, integrity, authenticity, and, in some settings, anonymity and accountability, among many other things.

With the rapid expansion of the World Wide Web and the increasing complexity of online applications, cryptography has evolved far beyond its traditional domains of secret sharing, encryption, and signature schemes. Modern cryptographic research encompasses advanced primitives and protocols such as multi-party computation, which allows mutually distrusting parties to jointly compute functions over private inputs, fully homomorphic encryption, enabling computation directly on encrypted data, and hierarchical identity-based broadcast encryption, designed to support scalable access control in large systems. Additional developments include zero-knowledge proofs, which permit proving knowledge of information without revealing the information itself, as well as cryptographic constructions used in decentralized technologies, including atomic swaps and other protocols supporting blockchain ecosystems.

The field of cryptography is traditionally divided into two main categories: private-key cryptography, also known as symmetric cryptography, and public-key cryptography, or asymmetric cryptography. In the former, secure communication between two users relies on the existence of a common secret key shared in advance between the communicating parties. This key is used both to encrypt and decrypt messages, allowing confidentiality to be preserved. While symmetric schemes are generally efficient and practical, their main limitation lies in the secure distribution and management of shared keys, especially in systems involving many users. In contrast, public-key cryptography enables secure communication without requiring a common shared secret beforehand. Instead, each participant possesses

a pair of keys consisting of a public key, which can be openly distributed, and a secret key, which must remain private. The security of such schemes is typically based on the computational hardness of solving specific mathematical problems. So the difficulty faced by an adversary is reduced to breaking these underlying assumptions, which is believed to be computationally infeasible. In addition to secure communication, public-key cryptography provides the foundation for numerous advanced cryptographic primitives, including digital signatures and key exchange protocols. Public-key cryptography and some of its associated constructions will be the main focus of this thesis.

1.1 Hard Problems in Classical Cryptography

The two main hard problems on which current-day public-key cryptography is based on are the problems of integer factorization and the Discrete Logarithm Problem. Integer factorization can be modeled as follows.

Definition 1.1.1 (Integer Factorization). The Integer Factorization problem asks, given $n = p \cdot q$ product of the two distinct (large) primes p and q , to find p and q .

The best known classical algorithm that solves this problem is the number field sieve, whose original idea can be found in [41], which runs in expected time

$$e^{[c+o(1)](\log n)^{\frac{1}{3}}(\log(\log(n)))^{\frac{2}{3}}}, \text{ with } c = 2 \cdot \left(\frac{2}{3}\right)^{\frac{2}{3}} \approx 1.526,$$

but no polynomial-time (classical) algorithm is known. The hardness of integer factorization is the foundation, for example, for the RSA cryptosystem [54]. For what concerns the Discrete Logarithm Problem, here is its formulation.

Definition 1.1.2 (Discrete Logarithm Problem (DLP)). The Discrete Logarithm Problem (DLP) asks, given a finite cyclic group (together with a generator) $G = \langle g \rangle$ (in multiplicative notation) and given $h = g^a$, to find a .

Clearly, for cryptographic applications, in order to avoid brute force attacks, the finite cyclic group in the discrete logarithm problem (DLP) has to be of large order.

Solving the discrete logarithm problem (DLP), namely recovering the exponent from a group element, is at least as hard as solving the Computational Diffie–Hellman (CDH) problem, which consists of computing the shared secret in the Diffie–Hellman key exchange protocol [23]. Indeed, an algorithm for the DLP immediately yields an algorithm for CDH by recovering the secret exponents and computing the corresponding shared key. Conversely, whether an efficient solution to the CDH problem implies an efficient solution to the DLP remains unknown in general.

The stress given on the classical composition of the hardware on which we are running the algorithms is crucial: in fact, there exists a polynomial-time quantum algorithm, Shor’s algorithm [59] [60], that allows a malicious user running on a quantum computer to break the Integer Factorization problem and the DLP in polynomial time.

The existence of Shor’s algorithm poses a big threat to those cryptosystems whose hardness is based on the Integer Factorization problem or on the DLP (and those systems are ubiquitous). Even if large scale quantum computers are not readily available as of today (2026), (see, for example IBM’s Quantum Computation Roadmap [36]) new technological advances would render our communication insecure. Also, the store-now decrypt-later paradigm will greatly impact forward secrecy once a large

scale quantum computer is built. To solve this, cryptographers have come up with alternative systems that shall substitute the endangered ones in the future. Those systems constitute the branch of cryptography that has been called Post-Quantum Cryptography. The goal of Post-Quantum cryptography is to replace classical systems whose security is affected by Shor's algorithm with other classical systems, whose hardness shall be based on mathematical problems whose difficulty is not affected by any quantum algorithm (including Shor's). In this thesis I will treat in detail various aspects of Post-Quantum Cryptography, from an implementation point of view, but also from a mathematical and theoretical perspective.

1.2 NIST Post-Quantum Cryptography Standardization and Post-Quantum Areas

Post-Quantum Cryptography can be divided into five main areas, depending on the mathematical problems underlying the cryptographical construction. Those areas were delimited after a 2017 call [47] for new Post-Quantum cryptosystems from the American Department of Commerce (DoC) agency NIST (National Institute of Standards and Technology). Since in this call only a small number of digital signatures emerged and passed the first rounds, in 2022 there has been an additional call by NIST for digital signatures [48]. This additional call was also made to diversify the mathematical tools that are used in the signatures.

In the following we give a list with the macro-areas, together with the most important schemes that have been created in it.

- Lattice-based cryptography (NTRUEncrypt, CRYSTALS - Dilithium (standardized), CRYSTALS - Kyber (standardized), Falcon (standardized), HAWK) : the hardness of the cryptosystems that are in this area is based on the fact that, given a n -dimensional integer lattice with a "bad" (i.e. very non-orthogonal, basis), it is hard for an attacker to find the shortest non-zero vector in it (SVP). There are also other computational problems that we will see in detail.
- Code-based cryptography (BIKE, Classic McEliece, QC-MDPC KEM, HQC (standardized), CROSS, LESS): the hardness of these cryptosystems is based, among other problems, on the fact that it is very hard for an attacker to say that two linear codes are equivalent or there exists a monomial transformation between the two.
- Hash-based cryptography (Gravity - SPHINCS, SPHINCS+ (standardized)): the hardness of these cryptosystems relies on the peculiar structure of Merkle trees of hashed strings and on the properties of hash functions.
- Isogeny-based cryptography (SIDH (broken), CSIDH, SQIsign): the hardness of these cryptosystems is based on the fact that it is hard for an attacker to find and write down isogenies between two elliptic curves.
- Multivariate cryptography (Rainbow (broken), UOV, MAYO, SNOVA): the hardness of these cryptosystems is based on the fact that it is computationally hard for an attacker to solve random non-linear (usually quadratic) systems of equations.

The previous list does not have the ambition to cover all the systems and all the hard problems behind them, but it is just a slice of the most important constructions that have arisen from the two NIST calls.

1.3 Outline of the Thesis

In this PhD Thesis we will address some aspects of post-quantum cryptography related to lattices and isogenies, and we will also talk about a special type of advanced signature that is useful for blockchain atomic swaps: the adaptor signature. In particular, the following is the outline of the thesis.

In Chapter 2 we will talk about cryptographic preliminaries and definitions that we will need in the thesis. We will see the definitions of cryptographic primitives: schemes, digital signatures, identification schemes, the Fiat-Shamir transformation, key encapsulation mechanisms, the Diffie-Hellman key exchange protocol, identity-based broadcast encryption schemes (IBBE) and their hierarchical counterparts (HIBBE). Moreover we will talk about the theory of adaptor signatures and their practical and security definitions.

In Chapter 3 we will talk extensively about lattices: we will give the basic definitions and properties, and then we will switch to the description of the most important lattice algorithms (Babai's algorithm, Gauss algorithm, the LLL algorithm) and hard problems (the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP), among others). After that we will talk about the Nearest Plane Algorithm and the Micciancio-Goldwasser Algorithm (for which we present the pseudo-code and extended proof of correctness for the first time in Theorem 3.4.2), both of which are key building blocks for the lattice-based post-quantum HIBBE scheme that we will describe in Chapter 5. We will discuss also about discrete Gaussian sampling over lattices and the Learning With Errors (LWE) problem (and some of its variants). Then we will describe two lattice-based cryptosystems: GGH and CRYSTALS - Dilithium.

Chapter 4 is entirely devoted to elliptic curves and isogenies. We discuss the basic definitions and properties; then, we talk about the endomorphism ring of an elliptic curve and of elliptic curves over finite fields and over the complex field $\mathbb{C}$. Then we will see the orders in imaginary quadratic fields and the ideal class group. We describe the action of the ideal class group on elliptic curves (which is the basis for many cryptosystems) and the Deuring Lifting Theorem. We will briefly talk about isogeny graphs and we will focus on an isogeny based cryptosystem: CSIDH [16].

In Chapter 5 we discuss the construction and we give an in-depth description of a lattice-based post quantum HIBBE cryptosystem by C. Yang, Zheng, Wang, Lu and Y. Yang [67]. In particular, we present the work we did on writing the pseudo-code of all the algorithms in the scheme (most of which do not appear elsewhere), using the procedures of Chapter 3. I also made an implementation in Python of this HIBBE scheme, with help from M. Trombini.

Finally, in Chapter 6, we present the work I did with R. Cartor, N. Daly, G. Gaggero, J. LeGrow and S. Sconza [12], which describes how one can build post-quantum adaptor signatures from cryptographic group actions with strong security. Our motivation was to construct an adaptor signature without having to modify the existing underlying digital signature scheme: we managed to build one from CSI-FiSh [11], which uses the group action on elliptic curves (with isogenies) described in CSIDH.

1.4 Principal results

The first main result of the thesis is the analysis of the hierarchical identity-based broadcast encryption scheme proposed by C. Yang, Zheng, Wang, Lu and Y. Yang in [67], done in Chapters 2, 3 and 5.

The interest in this scheme is due to the fact that it is one of the few of its kind to be post-quantum, and relies on the hardness of the LWE, i.e. on lattice-based problems. For the first time we

give a detailed description of the scheme and we present a full introduction to the mathematical and cryptographic background needed to describe it.

In particular, in Sections 2.4 and 2.5 we introduce the cryptographic notions of identity-based broadcast encryption scheme (IBBE) and its hierarchical form (HIBBE). Then, in Chapter 3 we deal with lattices and algorithms concerning them. We give the first definitions and the main lattice problems with cryptographic relevance; then, we discuss Babai's Algorithm, the Gauss Reduction Algorithm and the LLL Algorithm. Then we describe the Nearest Plane Algorithm and we give a detailed discussion and proof of correctness of an Algorithm by Micciancio and Goldwasser, which can be found in [44, Lemma 7.1]. Then, we move on to discussing how to sample discrete Gaussian distributions over lattices and we introduce the Learning With Errors (LWE) problem and some of its variants. In the last section of the chapter we present two lattice-based cryptographic constructions: GGH and CRYSTALS - Dilithium. Finally, in Chapter 5, we present the main construction.

The second main result of the thesis is the construction of an adaptor signature for CSI-FiSh [11] that we called ORCAS [12]. This is work we did with Ryann Cartor, Nathan Daly, Giulia Gaggero, Jason T. LeGrow and Silvia Sconza. We present it in Chapters 2, 4 and 6.

Cryptographic group actions are a promising technique that underlie several post-quantum digital signatures, including CSI-FiSh. Despite this, there are only two group action-based adaptor signature constructions in the literature: Isogeny Adaptor Signatures (IAS) [66] and adaptor modified CSI-FiSh [37]. Each of these constructions has notable disadvantages. IAS requires a large, expensive non-interactive zero-knowledge proof, making pre-signatures between 10 and 70 times as large as signatures, and taking between 15 and 20 times as long to produce as signatures. Adaptor modified CSI-FiSh produces signatures with a slightly different format from CSI-FiSh, meaning they will not pass CSI-FiSh verification, meaning that the scheme can only be used on a blockchain which has standardized both the original and the modified signatures. Our proposal, ORCAS, which stands for One-Round "Cheating" Adaptor Signatures (so named because its pre-signing algorithm resembles how cheating signer might try to forge a Fiat-Shamir-based signature), is directly compatible with CSI-FiSh. While the signatures are produced in a different manner and follow a slightly different distribution, they can still be verified by a blockchain which supports only CSI-FiSh.

In Section 2.6 we define adaptor signatures and their properties. Then, in Chapter 4 we discuss elliptic curves and isogenies. More precisely, we talk about the endomorphism ring of elliptic curves, elliptic curves over finite fields and over the complex numbers. Then, we discuss orders in imaginary quadratic fields, ideal class groups, the Deuring Lifting Theorem and isogeny graphs. Then we present an isogeny-based cryptographic construction: CSIDH [16]. Finally, in Chapter 6 we present the basic version of ORCAS, namely Base ORCAS, and we prove that it is a correct and secure adaptor signature for a slightly modified version of CSI-FiSh, which we call Filtered CSI-FiSh. Several other technical results about filtering allow to deduce from this that Base ORCAS is correct and secure so for the standard version of CSI-FiSh.

1.5 Notations

We collect here the notation used in the thesis for algorithms, functions and rings.

- If an algorithm A is probabilistic on inputs $(a_1, \dots, a_n)$, we will write the result of its execution as

$$b \leftarrow A(a_1, \dots, a_n).$$

Otherwise, if A is deterministic, we will write $b = A(a_1, \dots, a_n)$.

- With a similar notation as before, if an element x is sampled (uniformly) randomly from a finite set X we write $x \leftarrow X$. Otherwise, if x is sampled from a set X (possibly infinite) according to a probability distribution $\mathcal{D}$ on X , we write $x \xleftarrow{\mathcal{D}} X$.
- The standard $\mathcal{O}, o, \Omega, \omega$ notation for functions $f : \mathbb{N} \rightarrow \mathbb{R}$ is used. Moreover, if $f(n) = g(n) \cdot \log^k(n)$ we use $\tilde{\mathcal{O}}(f(n))$ to denote $\mathcal{O}(g(n))$, i.e. we use $\tilde{\mathcal{O}}$ to ignore logarithmic factors.
- A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is negligible if $f = o(1/n^t)$ for every $t \in \mathbb{N}$. Sometimes, to indicate a negligible function in n we write $\text{negl}(n)$.
- We denote with $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ the standard dot product in $\mathbb{R}^n$ and we denote with $\|\cdot\|_2 = \sqrt{\langle \cdot, \cdot \rangle}$ the standard Euclidean norm on $\mathbb{R}^n$.
- Given a modulus $q \in \mathbb{Z}_{>1}$, we indicate the integers modulo q with $\mathbb{Z}/q\mathbb{Z}$. If $q = p$ is a prime number, $\mathbb{Z}/p\mathbb{Z}$ will also be denoted by $\mathbb{F}_p$. If $q = p^r$ is a power of a prime p , we indicate the finite field with $q = p^r$ elements as $\mathbb{F}_q$.

Chapter 2

Cryptographic Preliminaries

In this chapter we will introduce some basic concepts in cryptography, focusing on encryption schemes, digital signatures, identification schemes, key encapsulation mechanisms, (hierarchical) identity-based broadcast encryption schemes and adaptor signatures.

2.1 Cryptosystems and Digital Signatures

The first definition we give is the one of encryption scheme, or cryptosystem, which is the basic theoretical construction that allows two parties to unilaterally exchange messages in a safe way.

Definition 2.1.1 (Encryption scheme/Cryptosystem). An encryption scheme or cryptosystem CS is a tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \tilde{\mathcal{K}}, \text{Enc}, \text{Dec})$, where $\mathcal{P}$ is the space of the messages (plaintexts), $\mathcal{C}$ is the space of the ciphertexts (encrypted plaintexts), $\mathcal{K}$ is the space of the encryption keys, $\tilde{\mathcal{K}}$ is the space of decryption keys and

$$\text{Enc} : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}, \text{Dec} : \mathcal{C} \times \tilde{\mathcal{K}} \rightarrow \mathcal{M}$$

are the encryption and decryption functions, such that for all $e_k \in \mathcal{K}$ there exists $d_k \in \tilde{\mathcal{K}}$ such that $\text{Dec}(\text{Enc}(m, e_k), d_k) = m$ for all $m \in \mathcal{M}$ (i.e. the system is correct). Moreover, e_k is called encryption key, while d_k is the decryption key. Analogously, one can write that, for all $e_k \in \mathcal{K}$ there exist $d_k \in \tilde{\mathcal{K}}$ such that for all $m \in \mathcal{M}$

$$\mathbb{P}[\text{Dec}(c, d_k) = m | c = \text{Enc}(e_k, m)] = 1.$$

For security reasons it is useful to model the encryption function not as a regular deterministic function but as a (efficient) probabilistic algorithm. Now, even if the results of the probabilistic encryption on a same message $m \in \mathcal{M}$ may be different, they should all decrypt to the same message m .

This means that, in an encryption scheme, we want that if $c \leftarrow E(m, e_k)$ and $m' = D(c, d_k)$ then it must be that $m = m'$.

Example 2.1.2 (One-time pad). Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \tilde{\mathcal{K}} = \{0, 1\}^\ell$, with $\ell \in \mathbb{N}_{\geq 1}$. The encryption and decryption functions are the same and are defined in the following way:

$$\begin{aligned} \text{Enc} = \text{Dec} : (\mathbb{Z}/2\mathbb{Z})^\ell \times (\mathbb{Z}/2\mathbb{Z})^\ell &\rightarrow (\mathbb{Z}/2\mathbb{Z})^\ell \\ (s = (s_1, \dots, s_\ell), k = (k_1, \dots, k_\ell)) &\mapsto (s_1 + k_1, \dots, s_\ell + k_\ell). \end{aligned}$$

It is fairly straightforward to see that if we use the same key k in encryption and decryption the system is correct. In fact, for all $m \in \mathcal{M}$ and for all $k \in \mathcal{K}$ we have that

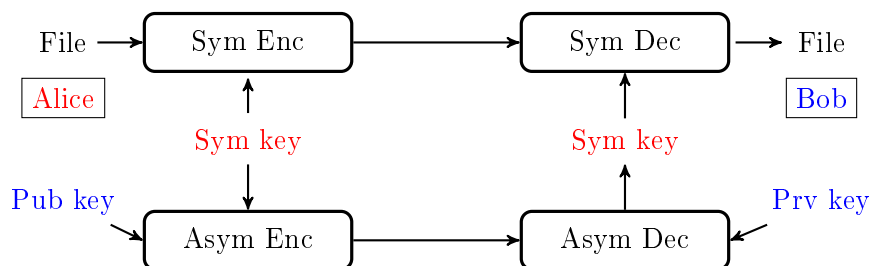
$$\text{Dec}(\text{Enc}(m, k), k)_i = (m_i + k_i) + k_i = m_i \text{ for all } i = 1, \dots, \ell,$$

since $2k_i = 0$ in $\mathbb{Z}/2\mathbb{Z}$. This cryptosystem is called one-time pad, because the (noisy) key we are adding to our message in the encryption phase and removing it in the decryption step must be changed after every use.

In this example we have seen that we use the same key to encrypt and decrypt a message, and this key must be exchanged between two parties beforehand in order to be able to communicate. Therefore this scheme falls into the category of secret-key cryptography schemes.

Definition 2.1.3 (Secret-key vs. Public-key cryptosystems). Secret-key cryptography (or symmetric cryptography) is the branch of cryptography that deals with cryptosystems for which there exist two efficient algorithms that allow, given an encryption key, to recover the corresponding decryption key and vice versa. Often times decryption and encryption keys are the same (thus the name symmetric). The keys, in practice, represent a shared secret between two parties (thus the name secret-key). Public-key cryptography (or asymmetric cryptography), on the other hand, deals with cryptosystems for which the encryption key is public (thus the name public-key) and the decryption key is maintained private. This means that there should not be an efficient way to recover the decryption key from the encryption key. This fact also breaks the symmetry between the two parties: the party that does not own the private key cannot decrypt the messages, but now everyone that has access to the public key can encrypt.

While secret-key cryptosystems have been around since the dawn of civilization (Caesar's cipher is an example, which is an ancestor of the one-time pad), public-key cryptography is a fairly modern innovation, because the first functioning public-key system, RSA, created by Ron Rivest, Adi Shamir, and Leonard Adleman, was first introduced in 1977 [54]. Their use spiked with the introduction of Internet and the World Wide Web in the 1990s, because of the need of a common shared secret between users. Naturally, secret-key and public-key systems can be utilized together: two parties first exchange a secret key using public-key means, and then they share a message using a secret-key system with the key they previously agreed upon.



Alice wants to send to Bob a File. Bob knows the decryption key **Prv key** of a public-key cryptosystem, and sends to Alice (and to everyone) the encryption key **Pub key**. Alice then chooses a secret symmetric key **Sym key**, and with the public-key system she is able to securely send it to Bob. So now both Alice and Bob know the key **Sym key**. Now, using a symmetric cryptosystem with this common secret key Alice is able to send the **Sym key**-encrypted File to Bob and Bob, using again **Sym key**, can safely decrypt it.

The main disadvantage of public-key cryptosystems is the fact that they tend to be significantly slower than the secret-key ones. So this combination of the two ends up to be the best compromise between the two, and it coincides with what is really used in practice.

Since public-key cryptosystems will appear a lot in this thesis, we present a more formal definition of them in the following.

Definition 2.1.4 (Public-key cryptosystem). A public-key cryptosystem CS consists of three polynomial-time algorithms (in the security parameter λ) ($\text{KeyGen}, \text{Enc}, \text{Dec}$) such that they perform as follows:

KeyGen: on input a security parameter 1^λ , it is a probabilistic polynomial-time algorithm that outputs a pair secret key/public key (sk, pk) .

$$(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda).$$

Enc: on input a public key pk and a message m it is a (possibly) probabilistic polynomial-time algorithm that outputs a ciphertext $c \leftarrow \text{Enc}(\text{pk}, m)$.

$$c \leftarrow \text{Enc}(\text{pk}, m).$$

Dec: on input a secret key sk and a ciphertext c it is a deterministic polynomial-time algorithm that outputs a message m' .

$$m' = \text{Dec}(\text{sk}, c).$$

If the ciphertext c is not decryptable, the algorithm Dec returns failure $\perp$.

Clearly, for correctness, we require that an honest sender gets his message decrypted, i.e. for every message m

$$\mathbb{P} \left[m = m' \mid \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ c \leftarrow \text{Enc}(\text{pk}, m) \\ m' = \text{Dec}(\text{sk}, c) \end{array} \right] = 1.$$

Public-key cryptosystem

$A, \text{KeyGen}(1^\lambda), \text{Dec}(\text{sk}, c)$	$B, \text{Enc}(\text{pk}, m)$
---	-------------------------------

$$(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$$

$$\begin{array}{ccc}
 & \xrightarrow{\text{pk}} & \\
 & & c \leftarrow \text{Enc}(\text{pk}, m) \\
 & \xleftarrow{c} & \\
 m', \perp \leftarrow \text{Dec}(\text{sk}, c) & &
 \end{array}$$

This refined definition for public-key cryptosystems allows us to precisely define security notions. The one we want to present here is indistinguishability under adaptive chosen ciphertext attack: IND-CCA security.

Definition 2.1.5 (IND-CCA Security). A public-key cryptosystem CS is IND-CCA if for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$ the advantage of $\mathcal{A}$ of winning the following game is $\leq \text{negl}(\lambda)$. The adversary $\mathcal{A}$ has access to a decryption oracle which decrypts arbitrary ciphertexts at the adversary's request, returning the plaintext. The challenger generates a key pair

$$(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$$

and publishes pk (so that the adversary can have it; the challenger retains sk). The adversary may perform a polynomial number of encryptions, calls to the decryption oracle based on arbitrary ciphertexts, or other operations. Eventually, the adversary submits two distinct chosen plaintexts m_0, m_1 to

the challenger. The challenger selects a bit $b \leftarrow \{0, 1\}$ uniformly at random, and sends the challenge ciphertext $c \leftarrow \text{Enc}(\text{pk}, m_b)$ back to the adversary. The adversary $\mathcal{A}$ is free to perform a polynomial number of additional computations or encryptions. Moreover the adversary may make further calls to the decryption oracle, but may not submit the challenge ciphertext c . In the end the adversary $\mathcal{A}$ will try to guess the value of b . Suppose that the event of the adversary $\mathcal{A}$ winning the game we have just described is called $\text{Exp}_{\mathcal{A}, CS}^{\text{IND-CCA}}(1^\lambda)$. Then we want that

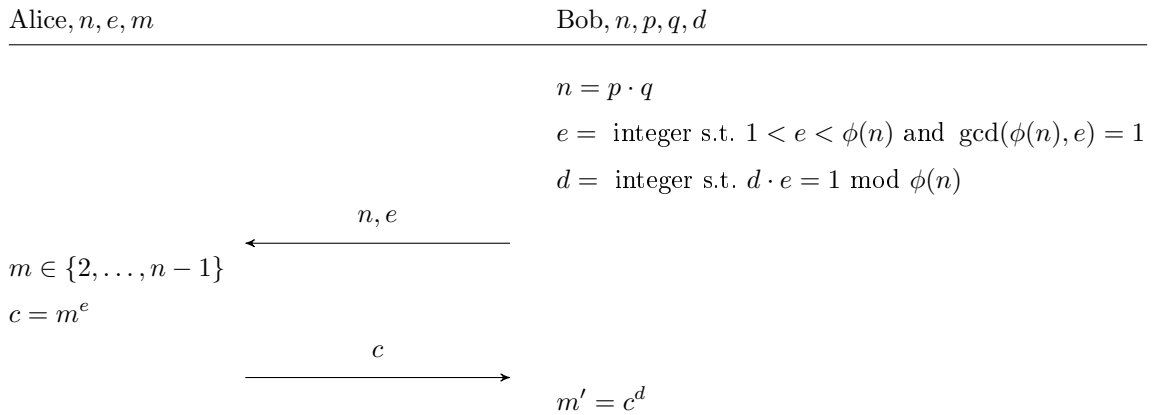
$$\left| \mathbb{P}[\text{Exp}_{\mathcal{A}, CS}^{\text{IND-CCA}}(1^\lambda)] - \frac{1}{2} \right| \leq \text{negl}(\lambda).$$

The first practical public-key cryptosystem which was proven to be IND-CCA secure in the standard model is the Cramer-Shoup cryptosystem [19].

Before we have presented an example of symmetric-key cryptosystem, here we present RSA, the prototypical public-key system.

Example 2.1.6 ("Textbook") RSA). In this example we explain the "textbook" version of the RSA cryptosystem. Bob computes $n = p \cdot q$ where p and q are two distinct large primes. Bob then computes $\phi(n) = (p - 1)(q - 1)$ (Euler totient function) and chooses an integer e such that $1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$. Using Euclid's algorithm, he computes an integer d such that $d \cdot e = 1 \bmod \phi(n)$. He publishes (e, n) as public key, and keeps (d, p, q) as his private key. Suppose that Alice wants to send a message m such that $1 < m < n$ to Bob. She computes $c = m^e \bmod n$ and she sends it to Bob. Now, Bob can decrypt m' as $m' = c^d = (m^e)^d$ and, thanks to Euler's Theorem and the Chinese Remainder Theorem, we have that $m' = m$.

RSA cryptosystem



The version of RSA used in practice is slightly different from the one above, due to security and execution-time reasons, but the underlying mathematical idea (that renders the system correct) is the same.

Cryptosystems are the first construction of cryptography, both from an historical point of view and as a concept. However, in modern cryptography there are many more tools that allow users to do much difficult interactions. The main new concept is the one of digital signatures.

Digital signatures are the analogous but improved version of the classical handwritten signatures. They keep the standard features which are authenticity and integrity, but they also provide something new, namely the non-repudiation of a signature, that is if someone digitally signs a document, then they cannot claim not to have signed it.

A digital signature scheme is a protocol between a signer S and a verifier V . The goal of the signer S is to sign a message m and send the message and its signature to the verifier V ; V must be able

to verify the identity of the signer (namely, S), without getting to know any secret (the secret key of S) that could allow V to sign other messages as S (thus we don't want V to be able to forge the signatures of S).

Definition 2.1.7. A digital signature scheme Σ is a tuple $(\text{KeyGen}, \text{Sign}, \text{Verify})$ of probabilistic algorithms defined as follows:

KeyGen: on input a security parameter 1^λ , outputs a pair secret key/public key (sk, pk) .

$$(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda).$$

Sign: on input a secret key sk and a message m , returns a signature σ .

$$\sigma \leftarrow \text{Sign}(\text{sk}, m).$$

Verify: on input a public key pk a message m and a signature σ , returns a bit $b \in \{0, 1\}$; 0 means that verification is not passed, while 1 means that the signature is valid.

$$\{0, 1\} \ni b = \text{Verify}(\text{pk}, m, \sigma).$$

In order for a digital signature to be correct, meaning that all the honestly-generated signatures pass verification, one has to assume that for all messages m and all $\lambda \in \mathbb{N}$, we have that

$$\mathbb{P} \left[\text{Verify}(\text{pk}, m, \sigma) = 1 \mid \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \sigma \leftarrow \text{Sign}(\text{sk}, m) \end{array} \right] = 1.$$

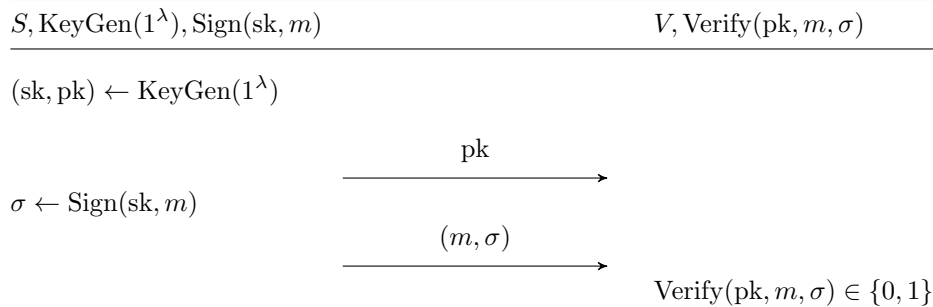
Here is how a signature scheme works:

1. The signer S generates a couple $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$ and publishes pk (which, then, is available to the verifier V).
2. The signer S signs the message m using his secret key, and gets $\sigma \leftarrow \text{Sign}(\text{sk}, m)$. Then he publishes the signature σ .
3. The verifier V computes $b = \text{Verify}(\text{pk}, m, \sigma)$. If $b = 1$ the signature is valid, otherwise the signature cannot be trusted.

Under the assumption at the end of the definition, the signature of a honest signer is always accepted as valid by the verifier.

The security of a signature scheme is based on the difficulty of forging a valid signature without knowing the secret key sk .

Digital signature scheme



There are many definitions of security for digital signatures. We report here the standard one, that addresses existential unforgeability under adaptive chosen message attack (EUF-CMA).

Definition 2.1.8 (EUF-CMA Security). A digital signature scheme Σ is EUF-CMA if for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $\mathbb{P}[\text{SigForge}_{\mathcal{A},\Sigma}(\lambda) = 1] \leq \text{negl}(\lambda)$ where the **SigForge** game is defined as follows.

$\text{SigForge}_{\mathcal{A},\Sigma}(\lambda)$	$\mathcal{O}_S(m)$
101 : $\mathcal{Q} \leftarrow \emptyset$	201 : $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
102 : $\mathcal{S} \leftarrow \emptyset$	202 : $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
103 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	203 : $\mathcal{S} \leftarrow \mathcal{S} \cup \{(m, \sigma)\}$
104 : $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_S}(\text{pk})$	204 : return σ
105 : assert $m \notin \mathcal{Q}$	
106 : return $((m, \sigma) \notin \mathcal{S} \wedge \text{Verify}(\text{pk}, m, \sigma))$	

This means that an adversary $\mathcal{A}$ who can use probabilistic algorithms, runs in polynomial-time and can call the oracle $\mathcal{O}_S(\cdot)$ a polynomial (in the security parameter λ) number of times, has negligible probability of winning the $\text{SigForge}_{\mathcal{A},\Sigma}(\lambda)$ game, i.e. $\mathcal{A}$ has negligible probability of forging a signature σ on a message m without knowing the secret key sk but with the possibility of asking a polynomial (in λ) number of times the oracle $\mathcal{O}_S(\cdot)$ for a signature σ' on a given message m' (with $m' \neq m$ for every m' chosen by $\mathcal{A}$).

In other words, the previous security definition tells us the following. The adversary $\mathcal{A}$ chooses a message m . The adversary $\mathcal{A}$ uses probabilistic algorithms, runs in polynomial time and can ask an oracle $\mathcal{O}_S(\cdot)$ to sign (a polynomial number in the security parameter λ of) messages (all different from m) with the secret key sk (the adversary $\mathcal{A}$ only knows the public key pk but the oracle $\mathcal{O}_S(\cdot)$ has access to the secret key sk). Then the probability that the adversary $\mathcal{A}$ forges a signature on m is negligible in λ .

Signatures can be built from scratch, but often they are constructed adapting in a very peculiar manner another cryptographic primitive: identification schemes.

2.2 Identification schemes and the Fiat-Shamir Transformation

An identification scheme (ID scheme) is an interactive protocol between a prover P and a verifier V . The goal of the prover P is to prove his identity to the verifier V , without sharing any secret (namely, his secret key) that could allow V to misidentify himself as P .

The prover P is equipped with a function

$$\text{Resp} : K \times T \times C \longrightarrow Z,$$

where K is the space of the secret keys, T is the space of the tests, C is the space of the challenges and Z is the space of the answers.

The verifier V is also equipped with a function

$$\text{Verify} : \tilde{K} \times Z \times T \times C \longrightarrow \{0, 1\},$$

where $\tilde{K}$ is the space of the public keys.

The functions Resp and Verify are related to each other, because we want the existence of a non-empty subset $L \subseteq K \times \tilde{K}$ such that $\forall(\text{sk}, \text{pk}) \in L$

$$\text{Verify}(\text{pk}, z, t, c) = 1 \iff z = \text{Resp}(\text{sk}, t, c)$$

$\forall(t, c, z) \in T \times C \times Z$. This means that

$$\mathbb{P} \left[\text{Verify}(\text{pk}, z, t, c) = 1 \mid \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow L \\ z = \text{Resp}(\text{sk}, t, c) \end{array} \right] = 1.$$

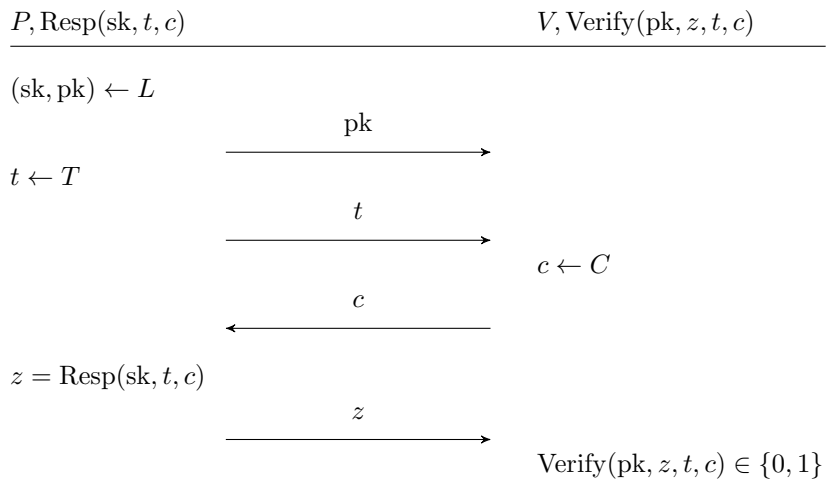
Now we explain how an ID scheme works:

1. The prover P picks a random couple $(\text{sk}, \text{pk}) \leftarrow L$ and publishes pk (which, then, is available to the verifier V).
2. The prover P chooses randomly $t \leftarrow T$ in the space of the tests T . Then P sends it to the verifier V .
3. The verifier V picks randomly a challenge $c \leftarrow C$ in the space of the challenges C . Then V sends it to P .
4. Using his function and his secret key, the prover P calculates $z = \text{Resp}(\text{sk}, t, c)$ (which lies in the space of the responses Z) and sends it to V .
5. Using his function and the public key, the verifier V computes $\text{Verify}(\text{pk}, z, t, c)$. Then, depending on the outcome, he accepts or rejects P 's proof of identity.

Thanks to the assumptions on L and to the relation between the functions Resp and Verify , we have that an honest proof is always accepted by the verifier, and so the scheme is correct.

The security of the ID scheme is based on the difficulty of giving a correct response z to the challenge c without knowing the function Resp .

ID scheme



Let us see an example of an ID scheme proposed in 1986 by Amos Fiat and Adi Shamir in [26].

Example 2.2.1. Let n be a public modulus which is the product of two secret distinct big primes l and q , such that it is computationally easy to compute square roots modulo n (for example, we can

take $n = l \cdot q$ with $l, q \equiv 3 \pmod{4}$). Let f be a pseudo-random function which maps arbitrary strings to the range $\{0, \dots, n-1\}$. The function f should be indistinguishable from a truly random function by any polynomially-bounded computation.

The prover P prepares a string I which contains all the relevant information about himself (name, address, ...) and then performs the following steps:

1. Compute the values $a_i = f(I, i)$ for small values of i .
2. Pick k distinct values of i for which a_i is invertible $\pmod{n}$ and is a quadratic residue $\pmod{n}$, then compute the smallest (in $0, \dots, n-1$) square root s_i of $a_i^{-1} \pmod{n}$.

The private key is comprised of the s_i values, while the public key is I . To simplify the notation, we assume that the first k indices $i = 1, \dots, k$ are used.

Now there is the interactive part of the protocol between the prover P and the verifier V .

1. P sends I to V .
2. V generates $a_i = f(I, i)$ for $i = 1, \dots, k$
3. P picks a random $r \in \{0, \dots, n-1\}$ and sends $t = r^2 \pmod{n}$ to V (the test).
4. V sends a random binary vector $(c_1, \dots, c_k) \in \{0, 1\}^k$ to P (the challenge).
5. P sends to V

$$z = r \prod_{i=1}^k s_i^{c_i} \pmod{n}$$

(the response).

6. V checks if

$$t = z^2 \prod_{i=1}^k a_i^{c_i} \pmod{n}.$$

The verifier V accepts P 's proof of identity only if the check is successful.

Naturally, if P and V follow the protocol, V always accept the proof of identity as valid, because

$$z^2 \prod_{i=1}^k a_i^{c_i} = r^2 \prod_{i=1}^k (s_i^2 a_i)^{c_i} = r^2 = t \pmod{n}.$$

Now let us pass to signature schemes. Our next goal is to produce a signature scheme using the format of the ID scheme, so the signer S will be equipped with the previous function Resp , while the verifier V will need the same function Verify as before. We call M the space of all the possible messages.

First, let us go the naive way.

The naive transformation works as follows:

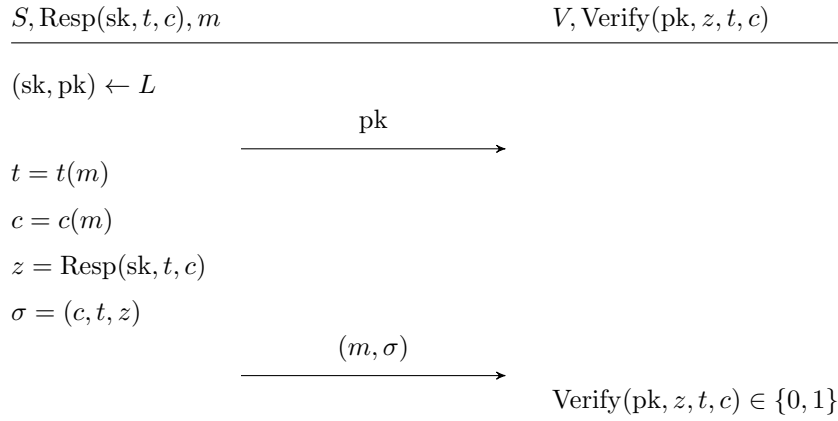
The signer S wants to sign a message $m \in M$ and sends it to the verifier.

1. S picks a random couple $(\text{sk}, \text{pk}) \leftarrow L$ and publishes pk .
2. The signer S produces a test $t = t(m) \in T$ in the space of the tests T (the notation $t(m)$ stands for the fact that t is somewhat related to m ; it is not important to focus on that).

3. The signer S produces a challenge $c = c(m) \in C$ (c is somewhat related to m ; it is not important to focus on that).
4. Using the function Resp and his private key sk the signer S computes $z = \text{Resp}(\text{sk}, t, c)$ and sends the message m and the signature $\sigma = (c, t, z)$ to V .
5. Using the signature $\sigma = (c, t, z)$ and his function Verify , the verifier V computes $\text{Verify}(\text{pk}, z, t, c)$. Then, depending on the outcome, he accepts or rejects the signature of S .

As it was the case of the original ID scheme, for the assumptions on L and on the functions Resp and Verify , an honestly produced signature is always accepted by the verifier.

Naive transformation of an ID scheme



This naive system constitutes the basic way to transform an ID scheme into a signature one. As one can tell by its construction, though, it has many flaws. In particular, in order for the verifier V to be able to verify the signature σ , σ has to contain the test, the challenge and the response. In addition to that, we would like to have a standard way to pick a test and to create a challenge.

An elaboration of an idea of Fiat and Shamir tells us how to improve the previous construction.

First of all we can observe that the pseudo-random function f of the ID scheme of the Example 2.2.1 can be chosen as an hash function $\mathcal{H}$ (or iterations of hash functions). This function can be exploited to transform any ID scheme into a signature one. Instead of asking the verifier V for a challenge, the prover/signer S creates a challenge for himself by hashing the concatenation of the message he wants to send and the test; then he sends to the verifier V the test and the response to the challenge. In this way S also avoids having to interact with the verifier several times (and if he wants to have stricter conditions on the response he can simply restart the signing procedure).

As we have just remarked, in the case of the Fiat-Shamir transformation, signer and verifier also need a public hash function $\mathcal{H} : M \times T \rightarrow C$ which maps couples (message, test) into challenges. The transformation works as follows:

The signer S wants to sign a message $m \in M$ and send it to the verifier.

1. S picks a random couple $(\text{sk}, \text{pk}) \leftarrow L$ and publishes pk .
2. The signer S chooses randomly $t \leftarrow T$ in the space of the tests T .
3. Using the public hash function $\mathcal{H}$, the signer S creates a hash of m and t : $c = \mathcal{H}(m, t)$.

4. Using the function Resp and his private key sk the signer S computes $z = \text{Resp}(\text{sk}, t, c)$ and sends the signature $\sigma = (t, z)$ to V .
5. Using the public hash function $\mathcal{H}$ and the first component of the signature $\sigma = (t, z)$, the verifier V computes $d = \mathcal{H}(m, t)$.
6. Using the signature $\sigma = (t, z)$ and his function Verify , the verifier V computes $\text{Verify}(\text{pk}, z, t, d)$. Then, depending on the outcome, he accepts or rejects the signature of S .

We still have that an honest signer is always accepted by the verifier.

Fiat-Shamir transformation of an ID scheme

$\mathcal{H} : M \times T \rightarrow C$ hash function

$S, \text{Resp}(\text{sk}, t, c), m$

$V, \text{Verify}(\text{pk}, z, t, c)$

$(\text{sk}, \text{pk}) \leftarrow L$

pk

$t \leftarrow T$

$c = \mathcal{H}(m, t)$

$z = \text{Resp}(\text{sk}, t, c)$

$\sigma = (t, z)$

(m, σ)

$d = \mathcal{H}(m, t)$

$\text{Verify}(\text{pk}, z, t, d) \in \{0, 1\}$

Remark 2.2.2. We have kept for exposure reasons the interactive description of the algorithm above, even though clearly the Fiat-Shamir transformation of an ID scheme is inherently non-interactive.

Now we see the Fiat-Shamir transformation of the ID scheme of the Example 2.2.1 into a signature scheme.

Example 2.2.3. All the notation is imported from the previous example, with the exception of P , who is replaced by S .

To sign a message m (seen as a binary string):

1. S picks at random $r \in \{0, \dots, n-1\}$ and computes $t = r^2 \pmod n$.
2. S computes $f(m, t)$ and uses its first k bits as $(c_1, \dots, c_k)$.
3. S computes

$$z = r \prod_{i=1}^k s_i^{c_i} \pmod n$$

and sends I, m, t and z to V .

To verify the signature of S on m :

1. V computes $a_j = f(I, j)$ for $j = 1, \dots, k$.
2. V computes $f(m, t)$ and extracts the first k bits as $(d_1, \dots, d_k)$.

3. V checks if

$$t = z^2 \prod_{i=1}^k a_i^{d_i} \pmod{n}.$$

With the same argument as before, one can see that if S and V follow the protocol, V always accept the signature as valid.

Very often we don't want to send the test t as part of the signature, so the signer S only sends the challenge c and the response z ($\sigma = (c, z) \in C \times Z$). In this way, if we still want to have a functioning signature scheme, the verifier V must have an algorithm that can extract the test (or a sufficient part of it) from the response. This is what Fiat and Shamir did in 1986:

Example 2.2.4. All the notation is imported from the previous examples.

To sign a message m (seen as a binary string):

1. S picks at random $r \in \{0, \dots, n-1\}$ and computes $t = r^2 \pmod{n}$.
2. S computes $f(m, t)$ and uses its first k bits as $(c_1, \dots, c_k)$.
3. S computes

$$z = r \prod_{i=1}^k s_i^{c_i} \pmod{n}$$

and sends I, m , the $(c_1, \dots, c_k)$ vector and z to V .

To verify the signature of S on m :

1. V computes $a_j = f(I, j)$ for $j = 1, \dots, k$.
2. V computes

$$w = z^2 \prod_{i=1}^k a_i^{c_i} \pmod{n}.$$

3. V verifies that the first k bits of $f(m, w)$ are $(c_1, \dots, c_k)$.

With the same argument as before, one can see that if S and V follow the protocol, V always accept the signature as valid.

The Fiat-Shamir transformation is a very useful way to turn an ID scheme into a digital signature.

In the rest of the thesis we will deal with more advanced cryptographic constructions, and our proposal for a special adaptation of digital signatures relies on this Fiat-Shamir heuristic.

2.3 Key Encapsulation Mechanisms and Diffie-Hellman Key Exchange

Another very important cryptographic construction is the one of key encapsulation mechanisms. Key Encapsulation Mechanisms (KEMs) are a particular kind public-key cryptosystems that allow Bob to generate a short secret key and transmit it to Alice confidentially, in a public channel, in spite of eavesdropping and intercepting adversaries.

Definition 2.3.1 (KEM). A key encapsulation mechanism (KEM) Δ is a tuple of probabilistic algorithms (KeyGen, Encap, Decap) defined as follows:

KeyGen: on input a security parameter 1^λ outputs a pair secret key/public key (sk, pk) :

$$(sk, pk) \leftarrow \text{KeyGen}(1^\lambda).$$

Encap: on input a public key pk it randomly chooses a private key k (whose length is a function of the security parameter 1^λ) and returns k along with its encapsulation c :

$$(k, c) \leftarrow \text{Encap}(pk).$$

Decap: on input a secret key sk and an encapsulation c' it either returns the encapsulated private key k' or fails:

$$k'/\perp \leftarrow \text{Decap}(sk, c').$$

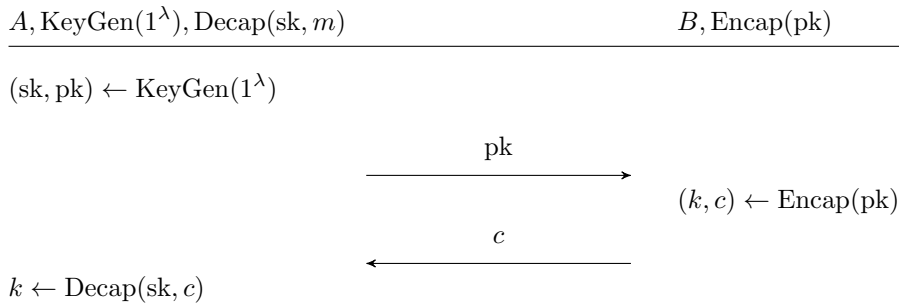
The scheme should be correct, i.e.

$$\mathbb{P} \left[k \neq k' \mid \begin{array}{l} (sk, pk) \leftarrow \text{KeyGen}(1^\lambda) \\ (k, c) \leftarrow \text{Encap}(pk) \\ k' \leftarrow \text{Decap}(sk, c) \end{array} \right] \leq \text{negl}(\lambda).$$

Here we explain how a key encapsulation mechanism can be used by Alice and Bob to obtain a common shared private key.

1. Alice A generates a couple $(sk, pk) \leftarrow \text{KeyGen}(1^\lambda)$ and publishes pk (which, then, is available to Bob B).
2. Bob B uses Encap with the public key pk and generates $(k, c) \leftarrow \text{Encap}(pk)$. Bob B keeps k and publishes c .
3. Alice A , with her secret key sk recovers k as $k \leftarrow \text{Decap}(sk, c)$ (with very high probability).

Key encapsulation mechanism



An example of KEM can be the RSA cryptosystem (Example 2.1.6), where, instead of a message m Alice and Bob get to share a private key k . Security notions for KEMs are related to security notions for public-key cryptosystems, such as IND-CCA security (see Definition 2.1.5).

Similar in nature to KEMs are key agreement protocols. They are systems in which two (or more) parties generate a cryptographic key as a function of information provided by each honest party so that no party can predetermine the resulting value. Moreover, all honest participants influence the outcome.

The most famous key agreement protocol is the Diffie-Hellman key exchange [23]. We present it here.

Setup: A probabilistic setup algorithm for the manager M that, given k bits of security, size of the identity string l and the maximum size n of receiver sets, outputs a public key pk and a master key mk . Here, pk includes n, l, k and descriptions of the shared key space $\mathcal{K}$ and of the ciphertext space $\mathcal{C}$; pk is published, and mk is given only to M .

$$(\text{pk}, \text{mk}) \leftarrow \text{Setup}(k, l, n).$$

We note that n determines only the maximum size of receiver sets to each of which ciphertexts are sent. On the other hand, l determines the number of potential receivers, which is superpolynomial if the scheme is identity-based.

KeyExt: A probabilistic key extraction algorithm for M that, given (mk, pk) as generated by the Setup and an identity string $\text{id} \in \{0, 1\}^l$ of a receiver, outputs a private key sk_{id} for the receiver with the identity id .

$$\text{sk}_{\text{id}} \leftarrow \text{KeyExt}(\text{pk}, \text{mk}, \text{id}).$$

Enc: A probabilistic encryption algorithm for senders that, given a receiver set S and a public key pk , outputs a shared key $\text{key} \in \mathcal{K}$ and a header $\text{Hdr} \in \mathcal{C}$. Here, each element of S is in $\{0, 1\}^l$ and $|S| \leq n$.

$$(\text{key}, \text{Hdr}) \leftarrow \text{Enc}(S, \text{pk}).$$

Dec: A probabilistic decryption algorithm for receivers that, given a receiver set S , a header Hdr for a receiver set S , $\text{id} \in S$, sk_{id} and the public key pk , outputs key .

$$\text{key} \leftarrow \text{Dec}(S, \text{id}, \text{sk}_{\text{id}}, \text{Hdr}, \text{pk}).$$

The correctness of an IBBE scheme is guaranteed by the following condition. For each $(\text{pk}, \text{mk}) \leftarrow \text{Setup}(k, l, n)$, for every $\text{id} \in \{0, 1\}^l$ such that $\text{sk}_{\text{id}} \leftarrow \text{KeyExt}(\text{pk}, \text{mk}, \text{id})$, if S is a receiver set, $\text{id}' \in S$, $(\text{key}, \text{Hdr}) \leftarrow \text{Enc}(S, \text{pk})$ and $\text{key}' \leftarrow \text{Dec}(S, \text{id}', \text{sk}_{\text{id}'}, \text{Hdr}, \text{pk})$, then $\text{key} = \text{key}'$.

We are ready to give the first example of this thesis of an IBBE scheme.

Before that, though, we briefly talk about bilinear pairings in groups.

Let $\mathcal{G}$ and $\mathcal{G}_T$ be groups of prime order p (in multiplicative notation). A bilinear pairing is an efficient map

$$e : \mathcal{G} \times \mathcal{G} \longrightarrow \mathcal{G}_T$$

such that:

- e is bilinear : for all $u, v \in \mathcal{G}$ and for all $\alpha, \beta \in \mathbb{Z}/p\mathbb{Z}$ we have $e(u^\alpha, v^\beta) = e(u, v)^{\alpha\beta}$.
- e is non-degenerate: if $g \in \mathcal{G}$ is a generator of $\mathcal{G}$, i.e. $\mathcal{G} = \langle g \rangle$, then $e(g, g)$ is a generator of $\mathcal{G}_T$, i.e. $\mathcal{G}_T = \langle e(g, g) \rangle$.

The next IBBE scheme was proposed by Sakai and Furukawa in [55].

Example 2.4.1 (Sakai-Furukawa IBBE scheme). Let $\mathcal{H} : \{0, 1\}^l \longrightarrow \mathbb{Z}/p\mathbb{Z}$ be a cryptographic hash function.

Setup: Given $n, l, k \in \mathbb{N}$, Setup first chooses $(\mathcal{G}, \mathcal{G}_T)$ cyclic groups of order p (in multiplicative notation) of size polynomial in k and an efficient bilinear pairing $e : \mathcal{G} \times \mathcal{G} \longrightarrow \mathcal{G}_T$. The Setup sets $\text{param} = (n, l, p, \mathcal{G}, \mathcal{G}_T, e)$ and it randomly chooses

$$P, Q \in \mathcal{G}, \quad \omega, \sigma \in \mathbb{Z}/p\mathbb{Z}.$$

Next, it generates

$$\begin{aligned}\tilde{P} &= P^\omega, \quad \tilde{Q} = Q^\omega, \\ (P_i)_{i=1,\dots,n} &= (P^{\sigma^i}), \quad (Q_i)_{i=1,\dots,n} = (Q^{\sigma^i}), \\ y &= e(\tilde{P}, Q).\end{aligned}$$

Finally, it outputs

$$\begin{aligned}\text{pk} &= \left(\text{param}, y, \begin{pmatrix} \tilde{Q} & Q & Q_1 & \cdots & Q_n \end{pmatrix} \right) \\ \text{mk} &= (\sigma, \tilde{P}).\end{aligned}$$

KeyExt: Given $\text{id} \in \{0, 1\}^l$, pk and mk, KeyExt generates

$$\text{sk}_{\text{id}} = \tilde{P}^{(\sigma + \mathcal{H}(\text{id}))^{-1}},$$

where $(\sigma + \mathcal{H}(\text{id}))^{-1}$ is the multiplicative inverse in $\mathbb{Z}/p\mathbb{Z}$. Then, it outputs sk_{id} .

Enc: Given a set of identities $S \subseteq \{0, 1\}^l$ such that $|S| \leq n$, Enc randomly chooses $\rho \in \mathbb{Z}/p\mathbb{Z}$ and generates

$$\begin{aligned}\text{key} &= y^\rho \\ \text{Hdr} &= (H_1, H_2, S) = \left(Q^{\rho \cdot \prod_{\text{id}' \in S} (\sigma + \mathcal{H}(\text{id}'))}, \tilde{Q}^\rho, S \right),\end{aligned}$$

where the product $\prod_{\text{id}' \in S} (\sigma + \mathcal{H}(\text{id}'))$ is in $\mathbb{Z}/p\mathbb{Z}$. Finally, it outputs (key, Hdr) .

Dec: Given pk, Hdr = (H_1, H_2, S) and sk_{id} such that $\text{id} \in S$, it generates

$$\begin{aligned}v &= e(\text{sk}_{\text{id}}, H_1) \cdot e \left(P^{\left(\prod_{\text{id}' \in S, \text{id}' \neq \text{id}} \mathcal{H}(\text{id}') \right)} \cdot P^{-\left(\prod_{\text{id}' \in S, \text{id}' \neq \text{id}} (\sigma + \mathcal{H}(\text{id}')) \right)}, H_2 \right) \\ \text{expectedkey} &= v^{\left(\prod_{\text{id}' \in S, \text{id}' \neq \text{id}} \mathcal{H}(\text{id}')^{-1} \right)},\end{aligned}$$

where the products and the inverses at the exponents are all in $\mathbb{Z}/p\mathbb{Z}$.

Now, the important thing is that $\text{expectedkey} = \text{key}$. The proof is a simple calculation.

Let $A = \prod_{\text{id}' \in S, \text{id}' \neq \text{id}} \mathcal{H}(\text{id}')$ (then $A^{-1} = \prod_{\text{id}' \in S, \text{id}' \neq \text{id}} \mathcal{H}(\text{id}')^{-1}$), $B = \prod_{\text{id}' \in S, \text{id}' \neq \text{id}} (\sigma + \mathcal{H}(\text{id}'))$ and $C = \prod_{\text{id}' \in S} (\sigma + \mathcal{H}(\text{id}'))$. Then :

$$\begin{aligned}\text{expectedkey} &= v^{A^{-1}} = \\ &= e(\text{sk}_{\text{id}}, H_1)^{A^{-1}} \cdot e \left(P^A \cdot P^{-B}, H_2 \right)^{A^{-1}} = \\ &= e(\tilde{P}^{(\sigma + \mathcal{H}(\text{id}))^{-1}}, Q^{\rho \cdot C})^{A^{-1}} \cdot e(P^{A-B}, H_2)^{A^{-1}} = \\ &= e(P^{\omega \cdot (\sigma + \mathcal{H}(\text{id}))^{-1}}, Q^{\rho C})^{A^{-1}} \cdot e(P^{A-B}, H_2)^{A^{-1}} = \\ &= e(P^{(\sigma + \mathcal{H}(\text{id}))^{-1} \cdot C}, Q^{\rho \omega})^{A^{-1}} \cdot e(P^{A-B}, H_2)^{A^{-1}} =\end{aligned}$$

$$\begin{aligned}
&= e(P^{(\sigma + \mathcal{H}(\text{id}))^{-1} \cdot C \cdot A^{-1}}, H_2) \cdot e(P^{1-B \cdot A^{-1}}, H_2) = \\
&= e(P^{(\sigma + \mathcal{H}(\text{id}))^{-1} \cdot C \cdot A^{-1} + 1 - B \cdot A^{-1}}, H_2) = \\
&= e(P \cdot P^{((\sigma + \mathcal{H}(\text{id}))^{-1} \cdot C - B) \cdot A^{-1}}, H_2) = \\
&= e(P, H_2) = e(P, \tilde{Q}^\rho) = e(P^\rho, \tilde{Q}) = \\
&= e(P^\rho, Q^\omega) = e(P^{\omega\rho}, Q) = \\
&= e(\tilde{P}^\rho, Q) = e(\tilde{P}, Q)^\rho = y^\rho = \\
&= \text{key},
\end{aligned}$$

where

$$(\sigma + \mathcal{H}(\text{id}))^{-1}C - B = (\sigma + \mathcal{H}(\text{id}))^{-1} \cdot \prod_{\text{id}' \in S} (\sigma + \mathcal{H}(\text{id}')) - \prod_{\text{id}' \in S, \text{id}' \neq \text{id}} (\sigma + \mathcal{H}(\text{id}')) = 0.$$

It is also worth mentioning that the public key pk also contains $P_1, \dots, P_n$ and $Q_1, \dots, Q_n$, in order to allow the sender to calculate H_1 and to allow the receivers to calculate

$$P^{-\left(\prod_{\text{id}' \in S, \text{id}' \neq \text{id}} (\sigma + \mathcal{H}(\text{id}'))\right)}.$$

Example 2.4.2. Let us look at an example, where the groups $\mathcal{G}$ and $\mathcal{G}_T$ are now in additive notation (multiplication becomes addition and exponentiation becomes multiplication by a scalar).

Let $n = 5, k = 10, l = 10$.

For Setup we take $p = 99991$ a prime, $\mathcal{G} = \mathcal{G}_T = (\mathbb{Z}/p\mathbb{Z}, +)$ and the bilinear pairing

$$\begin{aligned}
e : \mathcal{G} \times \mathcal{G} &\longrightarrow \mathcal{G}_T \\
(a, b) &\longmapsto a \cdot b.
\end{aligned}$$

Then, we choose $P = 23, Q = 99 \in \mathcal{G}$ and $\omega = 12, \sigma = 88 \in \mathbb{Z}/p\mathbb{Z}$.

Next, we generate $\tilde{P} = \omega \cdot P = 276$ and $\tilde{Q} = \omega \cdot Q = 1188$.

Then, Setup calculates $y = e(\tilde{P}, Q) = \tilde{P} \cdot Q = 27324$.

It outputs all the parameters, $y, \tilde{Q}$ and Q as public parameters in pk and σ and $\tilde{P}$ as mk .

Now, as hash function $\mathcal{H}$ we select the function that maps an element of $\{0, 1\}^{10}$ in its decimal representation in $\mathbb{Z}$ and then in its class in $\mathbb{Z}/p\mathbb{Z}$.

Suppose the identity of the sender is $\text{id}_{\text{send}} = 0000100000$, who wants to exchange a key with $\text{id}_{\text{rec1}} = 0000011101$ and $\text{id}_{\text{rec2}} = 0011100101$. Then $\mathcal{H}(\text{id}_{\text{send}}) = 32, \mathcal{H}(\text{id}_{\text{rec1}}) = 29, \mathcal{H}(\text{id}_{\text{rec2}}) = 229 \in \mathbb{Z}/p\mathbb{Z}$, and

$$\text{sk}_{\text{id}_{\text{rec1}}} = 87174, \text{sk}_{\text{id}_{\text{rec2}}} = 90529.$$

The sender selects randomly $\rho = 21 \in \mathbb{Z}/p\mathbb{Z}$ and computes $\text{key} = \rho \cdot y = 73849$ and the header

$$\begin{aligned}
\text{Hdr} &= (H_1, H_2, \{\text{id}_{\text{rec1}}, \text{id}_{\text{rec2}}\}) = ((\rho \cdot (\sigma + \mathcal{H}(\text{id}_{\text{rec1}})) \cdot (\sigma + \mathcal{H}(\text{id}_{\text{rec2}}))) \cdot Q, \rho \cdot \tilde{Q}, \{\text{id}_{\text{rec1}}, \text{id}_{\text{rec2}}\}) = \\
&= (14970, 24948, \{0000011101, 0011100101\}).
\end{aligned}$$

The sender, then, outputs to the public the header $\text{Hdr} = (H_1, H_2, S)$.

Now, the first receiver $\text{id}_{\text{rec1}} = 0000011101$ wants to get key, and so he computes

$$\begin{aligned}
v_1 &= e(\text{sk}_{\text{id}_{\text{rec1}}}, H_1) + e(\mathcal{H}(\text{id}_{\text{rec2}}) \cdot P - (\sigma + \mathcal{H}(\text{id}_{\text{rec2}})) \cdot P, H_2) = \\
&= \text{sk}_{\text{id}_{\text{rec1}}} \cdot H_1 + \mathcal{H}(\text{id}_{\text{rec2}}) \cdot P - (\sigma + \mathcal{H}(\text{id}_{\text{rec2}})) \cdot P \cdot H_2 =
\end{aligned}$$

$$= 12942.$$

Then, he calculates

$$\text{expectedkey}_1 = \mathcal{H}(\text{id}_{\text{rec2}})^{-1} \cdot v_1 = 73849,$$

and we see that it is equal to key.

The second receiver $\text{id}_{\text{rec2}} = 0011100101$ does the analogue thing, and gets

$$\begin{aligned} v_2 &= e(\text{sk}_{\text{id}_{\text{rec2}}}, H_1) + e(\mathcal{H}(\text{id}_{\text{rec1}}) \cdot P - (\sigma + \mathcal{H}(\text{id}_{\text{rec1}})) \cdot P, H_2) = \\ &= \text{sk}_{\text{id}_{\text{rec2}}} \cdot H_1 + \mathcal{H}(\text{id}_{\text{rec1}}) \cdot P - (\sigma + \mathcal{H}(\text{id}_{\text{rec1}})) \cdot P \cdot H_2 = \\ &= 41810, \end{aligned}$$

and $\text{expectedkey}_2 = \mathcal{H}(\text{id}_{\text{rec1}})^{-1} \cdot v_2 = 73849 = \text{key}$.

An extension of IBBE schemes on which we will focus involves the addition of a hierarchical structure, thus creating the hierarchical identity-based broadcast encryption schemes (HIBBE).

2.5 Hierarchical Identity-based Broadcast Encryption schemes (HIBBE)

The hierarchical structure adds some complexity to the construction compared to the IBBEs, but the formal definition of HIBBEs is strongly related to their non-hierarchical counterpart.

An HIBBE algorithm, in a similar way to its non-hierarchical construction, consists of five algorithms: Setup, KeyExt, Delegate, Enc and Dec. Now, the identities below the manager M are not all direct children of M , but they belong to a tree, with root M . To denote an identity id_i at depth level l we put a superscript l to it, making it id_i^l . Whenever it is necessary to keep track of the list of parents a certain identity has, we will write id_i^l as $[\text{id}_{i_1}^1, \text{id}_{i_2}^2, \dots, \text{id}_{i_{l-1}}^{l-1}, \text{id}_i^l]$, where this list represents all the parents of id_i^l from the first level below M ($\text{id}_{i_1}^1$) to the direct parent $\text{id}_{i_{l-1}}^{l-1}$. When we write id_i without an apex we mean that the identity can be at any level of the hierarchy.

Setup : A probabilistic setup algorithm that takes as inputs the security parameter n , the maximal hierarchy depth d , the maximal size h of the receiver set and it outputs a list of public parameters pp and the master key mk .

$$(\text{pp}, \text{mk}) \leftarrow \text{Setup}(n, d, h).$$

KeyExt: This algorithm takes as inputs the master key mk and an identity id_j^l of depth $l \leq d$ and outputs a secret key $B_{\text{id}_j^l}$ for the identity id_j^l .

$$B_{\text{id}_j^l} \leftarrow \text{KeyExt}(\text{mk}, \text{id}_j^l).$$

Delegate: This algorithm takes as inputs public parameters pp , a secret key $B_{\text{id}_j^l}$ of a parent node identity id_j^l ($l < d$), a child node identity id_i^s at depth s ($s \leq d$) and outputs a secret key $B_{\text{id}_i^s}$ for the identity id_i^s .

$$B_{\text{id}_i^s} \leftarrow \text{Delegate}(\text{pp}, B_{\text{id}_j^l}, \text{id}_i^s),$$

asserted that id_i^s is a child node of id_j^l .

Enc: A probabilistic polynomial-time algorithm that takes as inputs the public parameters pp , a receiver identity set $\mathcal{R} = [\text{id}_1, \dots, \text{id}_k]$ with $k \leq h$ and outputs a pair (Hdr, K) , where Hdr is the ciphertext header and K is a symmetric encryption key used to encrypt messages.

$$(\text{Hdr}, K) \leftarrow \text{Enc}(\text{pp}, \mathcal{R}).$$

Dec: This algorithm takes as inputs the public parameters pp , a receiver identity set $S = [\text{id}_1, \dots, \text{id}_k]$ with $k \leq h$, an identity id_i , the corresponding secret key B_{id_i} , a header Hdr , and, if $\text{id}_i \in S$ it outputs the symmetric encryption key K , which is used to recover messages.

$$\begin{cases} K & \text{if } \text{id}_i \in S \\ \perp & \text{if } \text{id}_i \notin S \end{cases} \leftarrow \text{Dec}(\text{pp}, S, \text{id}_i, B_{\text{id}_i}, \text{Hdr}).$$

For correctness, the HIBBE system must satisfy the following. For all S subset of identities such that $|S| \leq h$, for all $\text{id}_i \in S$, if $(\text{pp}, \text{mk}) \leftarrow \text{Setup}(n, d, h)$, $B_{\text{id}_i} \leftarrow \text{KeyExt}(\text{mk}, \text{id}_i)$ and $(\text{Hdr}, K) \leftarrow \text{Enc}(\text{pp}, S)$, then

$$\text{Dec}(\text{pp}, S, \text{id}_i, B_{\text{id}_i}, \text{Hdr}) = K.$$

One can prove that in the hierarchy, the manager M can decrypt every message, and every parent of a node which is sending the message (i.e. every node higher in the hierarchy and ancestor) can decrypt the message being sent.

2.6 Adaptor Signatures

The last cryptographic construction that we need for this thesis is the one of adaptor signatures. As the name suggests adaptor signatures are an integration for signatures. This integration extends a digital signature scheme by incorporating a pre-signing step, in which parties irrevocably commit to signing messages when specified conditions are met. In particular, Alice and Bob can agree on messages m_A, m_B and construct pre-signatures (intuitively, commitments to signatures) on them. Alice will have the ability to adapt Bob's pre-signature into a signature on m_B , but by doing so she will reveal enough information for Bob to adapt Alice's pre-signature into a signature on m_A . In this way, each party is assured that if their message becomes signed, they can produce a signature on their partner's message.

To deal effectively with adaptor signatures we have to introduce the concept of hard relations.

Definition 2.6.1 (Hard Relation). A hard relation $\mathcal{R}$ with underlying language $\mathcal{L}$ is a NP relation such that there is a probabilistic polynomial-time sampling algorithm GenR , that, on input the security parameter 1^λ , outputs a statement/witness pair (Y, y) , such that, for all probabilistic polynomial-time adversaries $\mathcal{A}$,

$$\mathbb{P} \left[(Y, y') \in \mathcal{R} \mid \begin{array}{l} (Y, y) \leftarrow \text{GenR}(1^\lambda) \\ y' \leftarrow \mathcal{A}(Y) \end{array} \right] = \text{negl}(\lambda),$$

where negl is a negligible function in the security parameter.

The definition before tells us that a hard relation is a binary relation $\mathcal{R}$ for which, given a statement/witness pair $(Y, y) \in \mathcal{R}$, for every probabilistic polynomial-time adversary $\mathcal{A}$ who knows the statement Y it is almost impossible to recover the witness y .

Hard relations can derive from many mathematical constructions, but one of the main settings in which they appear are group actions.

We are now ready to define adaptor signatures.

Definition 2.6.2 (Adaptor Signature scheme). An adaptor signature scheme with respect to a digital signature scheme $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify})$ and a hard relation $\mathcal{R}$ with underlying language $\mathcal{L}$ is a tuple $\mathcal{AS} = (\text{PreSign}, \text{PreVerify}, \text{Adapt}, \text{Ext})$ of algorithms defined as follows:

PreSign: On input a secret key sk , a message m and a statement $Y \in \mathcal{L}$, the pre-signing algorithm outputs a pre-signature $\hat{\sigma}$.

$$\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y).$$

PreVerify: On input a public key pk , a message m , a statement $Y \in \mathcal{L}$ and a pre-signature $\hat{\sigma}$, the pre-verification algorithm outputs a bit $b \in \{0, 1\}$.

$$\{0, 1\} \ni b \leftarrow \text{PreVerify}(\text{pk}, m, \hat{\sigma}).$$

Adapt: On input a pre-signature $\hat{\sigma}$ and a witness y for the relation $\mathcal{R}$, the adapting algorithm outputs a signature σ .

$$\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y).$$

Ext: On input a signature σ , a pre-signature $\hat{\sigma}$ and a statement $Y \in \mathcal{L}$, the extraction algorithm outputs either a witness y such that $(Y, y) \in \mathcal{R}$, or $\perp$.

$$y/\perp \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y).$$

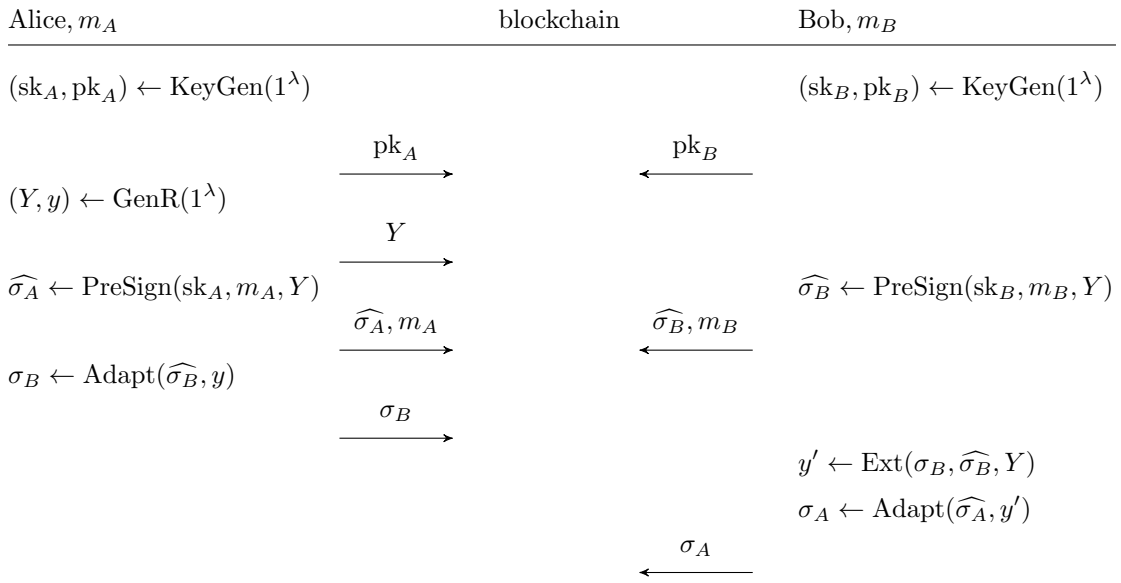
If the scheme is used honestly, it must abide the following correctness notion. $\mathcal{AS}$ is (pre-signature) correct if for all $\lambda \in \mathbb{N}$, for all $(Y, y) \in \mathcal{R}$ and for any message m , we have

$$\mathbb{P} \left[\begin{array}{l} \text{PreVerify}(\text{pk}, m, Y, \hat{\sigma}) = 1 \\ \text{Verify}(\text{pk}, m, \sigma) = 1 \\ (Y, y') \in \mathcal{R} \end{array} \middle| \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ \hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y) \\ \sigma \leftarrow \text{Adapt}(\hat{\sigma}, y) \\ y' \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y) \end{array} \right] = 1.$$

Alice and Bob can use an adaptor signature scheme to exchange digital goods securely across blockchains, as follow: first, Alice and Bob agree to exchange Alice's good A for Bob's good B. Let Alice's message m_A encode "Send good A to Bob", and Bob's message B encode "Send good B to Alice". Then:

1. Alice pre-signs m_A and publishes the pre-signature on the blockchain.
2. Bob pre-signs m_B and publishes the pre-signature.
3. Alice adapts Bob's pre-signature into a signature, and records this signature on the blockchain. This authorizes sending Bob's good B to Alice.
4. Bob uses his pre-signature and signature (obtained from the blockchain) to adapt Alice's pre-signature into a signature, which he records on the blockchain. This authorizes sending Alice's good A to Bob.

Digital goods swap on the blockchain



Remark 2.6.3. Since we assume in the protocol description that both Alice and Bob are honest, we have that

$$\text{PreVerify}(pk_A, m_A, Y, \widehat{\sigma}_B) = 1 \text{ and } \text{PreVerify}(pk_B, m_B, Y, \widehat{\sigma}_B) = 1.$$

There are some additional security conditions on adaptor signatures that guarantee that a dishonest actor in the exchange can be caught.

Dai, Okamoto and Yamamoto [20] and Gerhart, Schröder, Soni and Thyagarajan [30, 29] are the proposers of the following security definitions for adaptor signatures, which replace older ones that have proven not to be secure enough in many real-world applications.

The first property we are going to address is pre-signature adaptability. This property guarantees that any valid pre-signature with respect to a statement in the language of the hard relation can be adapted into a valid signature. This prevents a cheating user from providing a pre-signature that pre-verifies but cannot be adapted, which would break the security of the atomic swap protocol.

Definition 2.6.4 (Pre-signature Adaptability). An adaptor signature scheme $\mathcal{AS}$ satisfies pre-signature adaptability if for all $\lambda \in \mathbb{N}$, for all $(sk, pk) \leftarrow \text{KeyGen}(1^\lambda)$, for all $(Y, y) \in \mathcal{R}$ and for all $m \in \{0, 1\}^* \left(= \bigcup_{n \geq 1} \{0, 1\}^n \right)$, if $\hat{\sigma}$ satisfies $\text{PreVerify}(pk, m, Y, \hat{\sigma}) = 1$, then $\text{Verify}(pk, m, \text{Adapt}(\hat{\sigma}, y)) = 1$.

To prevent differential attacks two other interesting concepts are unlinkability and canonical signing.

Unlinkability guarantees that ordinary signatures (produced using the signing algorithm of the underlying signature scheme) cannot be distinguished from adapted pre-signatures. Thus, a third-party observer who sees the signatures coming from the atomic swap cannot identify those signatures as having arisen from an atomic swap.

Definition 2.6.5 (Unlinkability). Let the Unlinkability game be defined as follows.

Unlinkability $_{\mathcal{A}, \mathcal{AS}}(\lambda)$	Challenge $_b(m, (Y, y))$	$\mathcal{O}_{\text{pS}}(m, Y)$
101 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	201 : assert $(Y, y) \in \mathcal{R}$	401 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
102 : $b \leftarrow \{0, 1\}$	202 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$	402 : return $\hat{\sigma}$
103 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{S}}, \mathcal{O}_{\text{pS}}, \text{Challenge}_b}(\text{pk})$	203 : $\sigma_0 \leftarrow \text{Adapt}(\hat{\sigma}, y)$	
104 : return $[b = b']$	204 : $\sigma_1 \leftarrow \text{Sign}(\text{sk}, m)$	$\mathcal{O}_{\text{S}}(m)$
	205 : return σ_b	301 : $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
		302 : return σ

For an adversary $\mathcal{A}$, we define the adversary's advantage as $\text{Adv}(\mathcal{A}) = 2\mathbb{P}[\text{Unlinkability}_{\mathcal{A}, \mathcal{AS}}(\lambda) = 1] - 1$. We say that an adaptor signature scheme $\mathcal{AS}$ is unlinkable if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $\text{Adv}(\mathcal{A}) \leq \text{negl}(\lambda)$.

We explain now in greater detail how the Unlinkability game between a challenger and an adversary $\mathcal{A}$ works. The challenger generates a couple (sk, pk) with the KeyGen algorithm on input a security parameter λ . He selects randomly a bit $b \in \{0, 1\}$ and he runs the Challenge_b algorithm: after asserting that $(Y, y) \in \mathcal{R}$, the challenger computes a pre-signature $\hat{\sigma}$ on m and adapts it with y to a signature σ_0 on m . At the same time, the challenger computes a "regular" signature σ_1 on m (using the Sign algorithm of the digital signature scheme). Now, we have two signatures on m : one coming from a pre-signature (σ_0) and one from a "regular" signature (σ_1). According to the random choice of b , the challenger outputs σ_b . He then gives pk and σ_b to the adversary $\mathcal{A}$. The adversary $\mathcal{A}$ has to output $b' = 0$ if he thinks that the signature comes from an adapted pre-signature and $b' = 1$ if, instead, he thinks that the signature is an output of the signing algorithm Sign . The adversary $\mathcal{A}$ can ask oracles $\mathcal{O}_{\text{pS}}(\cdot, \cdot)$ and $\mathcal{O}_{\text{S}}(\cdot)$ for (a polynomial number in λ of) pre-signatures and signatures on any message, respectively. The adversary $\mathcal{A}$ can also ask the Challenge_b oracle a polynomial in λ number of times. Then, to achieve unlinkability, the probability that the adversary $\mathcal{A}$ succeeds in distinguishing the output of Challenge_b (i.e the probability that the adversary's output b' is equal to b) has to be $\frac{1}{2} + \text{negl}(\lambda)$, i.e. the probability of winning a coin toss plus a negligible function in the security parameter λ .

A related notion to unlinkability is canonical signing.

Definition 2.6.6 (Canonical Signing, Strongly Canonical Signing). Canonical signing requires that for any message m and secret key sk , ordinary signatures are distributed identically to signatures coming from the canonical signing algorithm CanonicalSign , defined as follows. This means that, given any message m and secret key sk , the output distributions of $\text{Sign}(\text{sk}, m)$ and $\text{CanonicalSign}(\text{sk}, m)$ inside the space of signatures are the same. We also define strongly canonical signing: the output of Sign is distributed identically to that of $\text{CanonicalSign}_{(Y, y)}$, for every pair $(Y, y) \in \mathcal{R}$.

CanonicalSign(sk, m)	CanonicalSign $_{(Y, y)}(\text{sk}, m)$
101 : $(Y, y) \leftarrow \text{GenR}(1^\lambda)$	201 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
102 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$	202 : $\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y)$
103 : $\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y)$	203 : return σ
104 : return σ	

Remark 2.6.7. Strongly canonical signing implies canonical signing.

In addition to unlinkability, we define a security property called weak unlinkability, which guarantees that, even if an observer may be able to distinguish an adapted pre-signature from a standard one, it is still hard to link an adapted pre-signature to the statement with respect to which it was generated.

In practice, this means that even if a user publishing a signature cannot hide the fact that they participated in an atomic swap, any information beyond that remains private.

Definition 2.6.8 (Weak Unlinkability). An adaptor signature scheme $\mathcal{AS}$ with respect to the signature Σ and hard relation $\mathcal{R}$ is weakly unlinkable if, for every probabilistic polynomial-time adversary $\mathcal{A}$ and for all $\lambda \in \mathbb{N}$

$$\mathbb{P}[\text{WeakUnlinkability}_{\mathcal{A}, \mathcal{AS}}(\lambda) = 1] \leq \frac{1}{2} + \text{negl}(\lambda),$$

where negl is a negligible function and the WeakUnlinkability game is defined as follows.

$\text{WeakUnlinkability}_{\mathcal{A}, \mathcal{AS}}(1^\lambda)$	$\text{Challenge}_b(m, (Y_0, y_0))$	$\mathcal{O}_S(m)$
101 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	201 : assert $(Y_0, y_0) \in \mathcal{R}$	301 : $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
102 : $b \leftarrow \{0, 1\}$	202 : $(Y_1, y_1) \leftarrow \text{GenR}()$	302 : return σ
103 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_S, \mathcal{O}_{\text{pS}}, \text{Challenge}_b}(\text{pk})$	203 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y_b)$	$\mathcal{O}_{\text{pS}}(m, Y)$
104 : return $[b = b']$	204 : $\sigma \leftarrow \text{Adapt}(\hat{\sigma}, y_b)$	401 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
	205 : return σ	402 : return $\hat{\sigma}$

The WeakUnlinkability game between a challenger and an adversary $\mathcal{A}$ works as follows. We generate a couple (sk, pk) with the KeyGen algorithm on input a security parameter λ . We select randomly a bit $b \in \{0, 1\}$ and we run the Challenge_b algorithm: after asserting that $(Y_0, y_0) \in \mathcal{R}$, we compute a fresh statement/witness pair $(Y_1, y_1) \leftarrow \text{GenR}()$. Then we compute a pre-signature $\hat{\sigma}$ on m using Y_b and we adapt it with y_b to a signature σ on m . Now, we output σ . We then give pk and σ to the adversary $\mathcal{A}$. The adversary $\mathcal{A}$ has to output $b' = 0$ if he thinks that the adapted pre-signature σ was generated using (Y_0, y_0) and $b' = 1$ if, instead, he thinks that the adapted pre-signature was generated using the fresh statement/witness pair (Y_1, y_1) . The adversary $\mathcal{A}$ can ask oracles $\mathcal{O}_{\text{pS}}(\cdot, \cdot)$ and $\mathcal{O}_S(\cdot)$ for (a polynomial number in λ of) pre-signatures and signatures on any message, respectively. Then, to get weak unlinkability, the probability that the adversary $\mathcal{A}$ succeeds in distinguishing the output of Challenge_b has to be $\frac{1}{2} + \text{negl}(\lambda)$.

Remark 2.6.9. Unlinkability implies weak unlinkability, and the converse holds true if $\mathcal{AS}$ has canonical signing for the digital signature scheme Σ [12, Section 2.1].

Dai, Okamoto and Yamamoto [20] also introduced the notions of strong full extractability and full extractability. These notions guarantee that it is infeasible for any adversary to produce a valid signature (on a new message for full extractability, on any message for strong full extractability) except by adapting a pre-signature given with respect to a statement that has a known witness.

Definition 2.6.10 (Full Extractability, Strong Full Extractability). An adaptor signature scheme $\mathcal{AS}$ satisfies full extractability [resp. strong full extractability] if, for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$

$$\mathbb{P}[\text{FullExtractability}_{\mathcal{A}, \mathcal{AS}}(\lambda) = 1] \leq \text{negl}(\lambda),$$

$$[\text{resp. } \mathbb{P}[\text{StrongFullExtractability}_{\mathcal{A}, \mathcal{AS}}(\lambda) = 1] \leq \text{negl}(\lambda),]$$

where the $\text{FullExtractability}$ [resp. $\text{StrongFullExtractability}$] game is defined below. (The lines in gray appear only in the "non-strong" version).

FullExtractability(λ) StrongFullExtractability(λ)		
101 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$		
102 : $C, T, S, U \leftarrow \emptyset$		
103 : $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{NewY}}, \mathcal{O}_S, \mathcal{O}_{\text{pS}}(\cdot, \cdot)}(\text{pk})$		
104 : assert $\text{Verify}(\text{pk}, m^*, \sigma^*) = 1$		
105 : assert $m^* \notin S$		
106 : assert $(m^*, \sigma^*) \notin U$		
107 : return $((Y, \text{Ext}(\sigma^*, \hat{\sigma}, Y)) \notin \mathcal{R} \ \forall (Y, \hat{\sigma}) \in T[m^*] \text{ such that } Y \notin C)$		
$\mathcal{O}_S(m)$	$\mathcal{O}_{\text{pS}}(m, Y)$	$\mathcal{O}_{\text{NewY}}$
201 : $\sigma \leftarrow \text{Sign}(\text{sk}, m)$	301 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$	401 : $(Y, y) \leftarrow \text{GenR}()$
202 : $S \leftarrow S \cup \{m\}$	302 : $T[m] \leftarrow T[m] \cup \{(Y, \hat{\sigma})\}$	402 : $C \leftarrow C \cup \{Y\}$
203 : $U \leftarrow U \cup \{(m, \sigma)\}$	303 : return $\hat{\sigma}$	403 : return Y
204 : return σ		

We now explain how the **FullExtractability** game works. We generate a secret key/public key couple (sk, pk) using **KeyGen** and the security parameter λ . The adversary $\mathcal{A}$ can ask the three oracles $\mathcal{O}_S(m)$, $\mathcal{O}_{\text{pS}}(m, Y)$ and $\mathcal{O}_{\text{NewY}}$ for a polynomial (in λ) number of times: the oracle $\mathcal{O}_S(m)$ can sign messages using the secret key sk , the oracle $\mathcal{O}_{\text{pS}}(m, Y)$ can compute pre-signatures given messages and statements and the oracle $\mathcal{O}_{\text{NewY}}$ can generate new statements in the relation $\mathcal{R}$. Given the public key pk to the adversary $\mathcal{A}$, the probability that he is able to generate a couple (m^*, σ^*) of message and signature with the following properties has to be negligible in the security parameter λ :

1. $\text{Verify}(\text{pk}, m^*, \sigma^*) = 1$, i.e. σ^* is a valid signature for m^* .
2. The message m^* never appeared as an input of the adversary's requests to the oracle $\mathcal{O}_S(\cdot)$.
3. The couple (m^*, σ^*) was never obtained as a request from the oracle $\mathcal{O}_S(\cdot)$ (i.e. the adversary $\mathcal{A}$ has never had σ^* as output of $\mathcal{O}_S(m^*)$).
4. For all $(Y, \hat{\sigma})$ such that Y never appeared as a new statement from the oracle $\mathcal{O}_{\text{NewY}}$ and $\hat{\sigma}$ is a pre-signature obtained from $\mathcal{O}_{\text{pS}}(m^*, Y)$, if $y^* = \text{Ext}(\sigma^*, \hat{\sigma}, Y)$, then the couple (Y, y^*) is not a statement/witness pair in $\mathcal{R}$.

The **StrongFullExtractability** game is the same as the **FullExtractability** one except that we remove condition 2. from the previous list, i.e. the adversary $\mathcal{A}$ can ask the oracle $\mathcal{O}_S(\cdot)$ on input m^* .

We also introduce extractability, which is equivalent to full extractability if the relation $\mathcal{R}$ is a hard relation and the scheme has canonical signing.

Definition 2.6.11 (Extractability). An adaptor signature scheme $\mathcal{AS}$ is extractable if for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$

$$\mathbb{P}[\text{Extractability}_{\mathcal{A}, \mathcal{AS}}(\lambda) = 1] \leq \text{negl}(\lambda),$$

where the **Extractability** game is defined as follows.

Extractability $_{\mathcal{A}, \mathcal{AS}}(\lambda)$	$\mathcal{O}_{\text{pS}}(m, Y)$
101 : $T \leftarrow \emptyset$	201 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
102 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	202 : $T[m] \leftarrow T[m] \cup \{(Y, \hat{\sigma})\}$
103 : $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pS}}(\cdot, \cdot)}(\text{pk})$	203 : return $\hat{\sigma}$
104 : assert $\text{Verify}(\text{pk}, m, \sigma) = 1$	
105 : for $(Y, \hat{\sigma}) \in T[m] : \text{assert } (Y, \text{Ext}(\sigma, \hat{\sigma}, Y)) \notin \mathcal{R}$	
106 : return 1	

The Extractability game and condition work as follows. We generate (sk, pk) using $\text{KeyGen}(1^\lambda)$. The adversary $\mathcal{A}$ is given the public key pk and the oracle $\mathcal{O}_{\text{pS}}$, which is the same as the one in the **FullExtractability** game. Then the probability that $\mathcal{A}$ is able to generate a couple (m, σ) , where σ is a valid signature on the message m such that for all couples statement/pre-signature $(Y, \hat{\sigma})$ generated as outputs of $\mathcal{O}_{\text{pS}}(m, Y)$ the couple $(Y, \text{Ext}(\sigma, \hat{\sigma}, Y))$ is not a statement/witness pair in $\mathcal{R}$, has to be negligible in the security parameter λ .

Another security definition is unique extractability. It guarantees that it is infeasible for any efficient (i.e. probabilistic polynomial-time) adversary to generate a pre-signature for some message and statement that corresponds to two distinct valid signatures on the message.

Definition 2.6.12 (Unique Extractability). An adaptor signature scheme $\mathcal{AS}$ is said to be uniquely extractable if for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function negl such that for every $\lambda \in \mathbb{N}$

$$\mathbb{P}[\text{UniqueExtractability}_{\mathcal{A}, \mathcal{AS}}(\lambda) = 1] \leq \text{negl}(\lambda),$$

where the **UniqueExtractability** game is defined as follows.

UniqueExtractability $_{\mathcal{A}, \mathcal{AS}}(\lambda)$	$\mathcal{O}_{\text{S}}(m)$
101 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	201 : $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
102 : $(m, Y, \hat{\sigma}, \sigma, \sigma') \leftarrow \mathcal{A}^{\mathcal{O}_{\text{S}}, \mathcal{O}_{\text{pS}}(\cdot, \cdot)}(\text{pk})$	202 : return σ
103 : assert $(\sigma' \neq \sigma)$	
104 : assert $\text{Verify}(\text{pk}, m, \sigma) = \text{Verify}(\text{pk}, m, \sigma') = 1$	$\mathcal{O}_{\text{pS}}(m, Y)$
105 : assert $\text{PreVerify}(\text{pk}, m, Y, \hat{\sigma}) = 1$	301 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
106 : $y = \text{Ext}(\sigma, \hat{\sigma}, Y)$	302 : return $\hat{\sigma}$
107 : $y' = \text{Ext}(\sigma', \hat{\sigma}, Y)$	
108 : return $(Y, y) \in \mathcal{R} \wedge (Y, y') \in \mathcal{R}$	

The **UniqueExtractability** game and condition work as follows. We generate the secret key/public key couple (sk, pk) using $\text{KeyGen}(1^\lambda)$. The adversary $\mathcal{A}$ can ask the oracles $\mathcal{O}_{\text{S}}$ and $\mathcal{O}_{\text{pS}}$ for signatures on m and pre-signatures on m and Y , respectively (for a polynomial in λ number of times). We have unique extractability if the probability that the adversary $\mathcal{A}$ can generate a tuple $(m, Y, \hat{\sigma}, \sigma, \sigma')$ with the following properties is negligible in λ :

1. m is a message.
2. Y is a statement in the language $\mathcal{L}$ in the relation $\mathcal{R}$.
3. σ and σ' are two valid signatures for m .
4. $\hat{\sigma}$ is a valid pre-signature on m according to the statement Y .
5. If $y = \text{Ext}(\sigma, \hat{\sigma}, Y)$ and $y' = \text{Ext}(\sigma', \hat{\sigma}, Y)$, then both (Y, y) and (Y, y') are valid statement/witness pairs for the relation $\mathcal{R}$.

While pre-signature adaptability ensures that valid pre-signatures with respect to honest statements $Y \in \mathcal{L}$ can be adapted into valid signatures, pre-verify soundness protects against adversaries using malicious statements by ensuring that for $Y \notin \mathcal{L}$ it is difficult to find valid pre-signatures with respect to Y . Gerhart, Schröder, Soni and Thyagarajan [29] define two notions of pre-verify soundness.

Definition 2.6.13 (Computational Pre-verify Soundness). An adaptor signature scheme $\mathcal{AS}$ is computationally pre-verify sound if for every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$ and all (polynomially-bounded) $Y \notin \mathcal{L}$

$$\mathbb{P} \left[\text{PreVerify}(\text{pk}, m, \hat{\sigma}, Y) = 1 \mid \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \\ (m, \hat{\sigma}) \leftarrow \mathcal{A}(\text{sk}) \end{array} \right] \leq \text{negl}(\lambda).$$

Definition 2.6.14 (Statistical Pre-verify Soundness). An adaptor signature scheme $\mathcal{AS}$ is statistically pre-verify sound if for all $\lambda \in \mathbb{N}$, for all (polynomially-bounded) $Y \notin \mathcal{L}$ and for all $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$

$$\mathbb{P}[\exists(m, \hat{\sigma}) \text{ such that } \text{PreVerify}(\text{pk}, m, \hat{\sigma}, Y) = 1] = 0.$$

Chapter 3

Lattices

In this chapter we introduce the concept of lattices and some hard problems related to them. Moreover, we will talk about some cryptosystems whose hardness is based on the difficulty of an adversary to solve those hard lattice problems: GGH and CRYSTALS - Dilithium. Some of the topics explored here follow the exposition of Bertuzzo [10] and Hoffstein, Pipher and Silverman [34].

3.1 Definitions and Properties

Definition 3.1.1 (Lattice). Given n linearly independent vectors $v_1, \dots, v_n \in \mathbb{R}^n$, the n -dimensional lattice $\mathcal{L}$ generated by those vectors is the discrete subgroup of $\mathbb{R}^n$

$$\mathcal{L}(v_1, \dots, v_n) = \left\{ \sum_{i=1}^n \lambda_i v_i \mid \lambda_i \in \mathbb{Z} \ \forall i = 1, \dots, n \right\},$$

i.e. $\mathcal{L}(v_1, \dots, v_n) = \text{span}_{\mathbb{Z}}(v_1, \dots, v_n)$.

In this case, we say that $B = [v_1, \dots, v_n]$ is a basis for the lattice $\mathcal{L}$, and it is very useful to interpret the vectors of the basis as columns of a square matrix, called basis matrix: $B = (v_1 | \dots | v_n) \in M_n(\mathbb{R})$. In this way we have that $\mathcal{L}(v_1, \dots, v_n) = \mathcal{L}(B) = \{Bx \mid x \in \mathbb{Z}^n\}$.

It is easy to see that if $U \in \text{GL}_n(\mathbb{Z})$ (i.e. U is a unimodular matrix), then $\mathcal{L}(B) = \mathcal{L}(BU)$, i.e. B and BU are bases of the same lattice. Indeed, the following result holds true.

Proposition 3.1.2. *Let $B, B' \in M_n(\mathbb{R})$ be two bases for two n -dimensional lattices. Then B, B' are bases for the same lattice $\mathcal{L}$ if and only if there exists a unimodular matrix U such that $B' = BU$.*

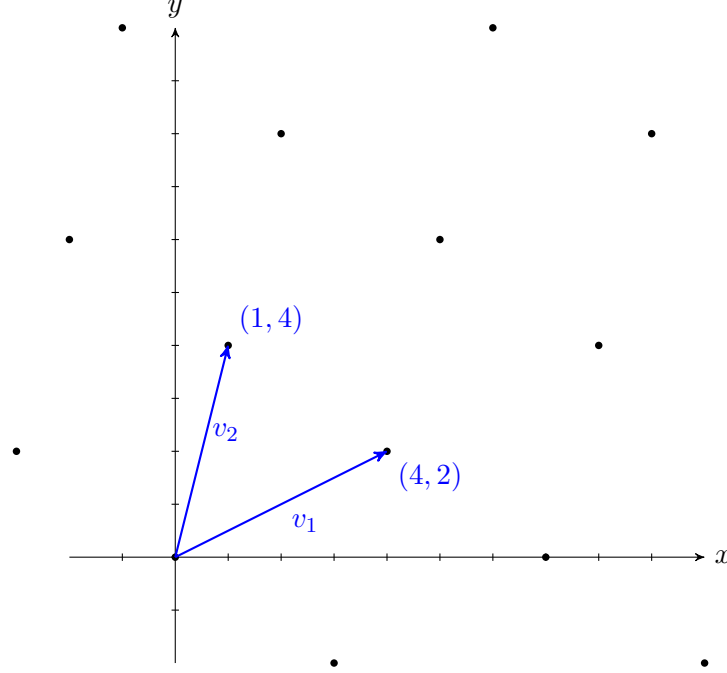
Proof. Suppose that $B, B' \in M_n(\mathbb{R})$ are two bases of the same lattice $\mathcal{L}$. Let us write $B = (v_1, \dots, v_n)$ and $B' = (v'_1, \dots, v'_n)$ with $v_1, \dots, v_n, v'_1, \dots, v'_n \in \mathbb{R}^n$ column vectors. Then, for every $i = 1, \dots, n$ we have that there exists $x_i \in \mathbb{Z}^n$ such that $Bx_i = v'_i$. Let us write U as $U = (x_1, \dots, x_n) \in M_n(\mathbb{Z})$. We have that $BU = B'$. Analogously, for every $i = 1, \dots, n$ we have that there exists $x'_i \in \mathbb{Z}^n$ such that $B'x'_i = v_i$. We put $U' = (x'_1, \dots, x'_n) \in M_n(\mathbb{Z})$. Then $B'U' = B$. Now, we have that $B'U'U = B'$. Since $\det(B') \neq 0$ we have that $\det(U')\det(U) = \det(U'U) = 1$. Now, since $U, U' \in M_n(\mathbb{Z})$, it must be that $\det(U), \det(U') \in \{\pm 1\}$. This means that U is unimodular. Vice versa, suppose that $B' = BU$ for some unimodular matrix U . We claim that $\mathcal{L}(B') = \mathcal{L}(B)$. In fact, if $v \in \mathcal{L}(B')$, then there exists $x \in \mathbb{Z}^n$ such that $v = B'x$. Then $v = BUx$ with $Ux \in \mathbb{Z}^n$, so $v \in \mathcal{L}(B)$. Since U is unimodular, $B = B'U^{-1}$ and we proceed as before (noting that U^{-1} is also unimodular). This concludes the proof of the claim. $\square$

In particular, every lattice admits multiple bases, and this is a key feature for many cryptographic applications.

Remark 3.1.3. A n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ can also be equivalently defined as a discrete, free, full-rank sub-module of $(\mathbb{R}^n, +)$ [15, Section III.4, Theorem VI].

Of particular interests are the integer lattices, that are lattices that live inside $\mathbb{Z}^n$, and so their bases consist of vectors with integer entries. In this thesis we will only focus on this type of lattices.

Example 3.1.4. Let us consider the vectors $v_1 = (4, 2), v_2 = (1, 4)$ in $\mathbb{Z}^2$. Then they generate a bidimensional (integer) lattice, and (being independent) they are a basis for it.

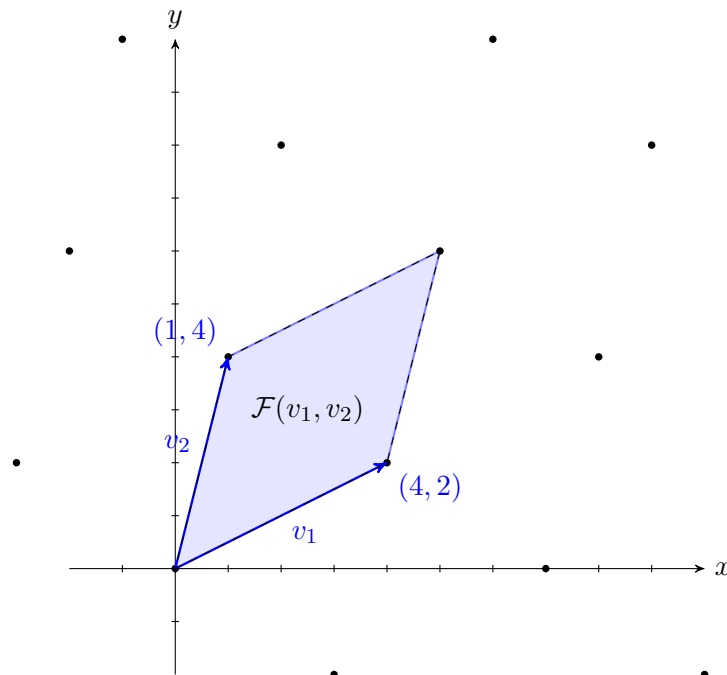


Another important aspect of lattices is their fundamental parallelepiped, also known in literature as fundamental domain.

Definition 3.1.5 (Fundamental Parallelepiped). Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and let $B = [v_1, \dots, v_n]$ be a basis for $\mathcal{L}$. The fundamental parallelepiped of $\mathcal{L}$ with respect to B is the set

$$\mathcal{F}(B) = \mathcal{F}(v_1, \dots, v_n) = \{t_1 v_1 + \dots + t_n v_n \mid 0 \leq t_i < 1, i = 1, \dots, n\}.$$

Below we depict the fundamental parallelepiped associated to the (integer) lattice of Example 3.1.4.



Fundamental parallelepipeds have great importance in the study of lattices because they allow to write a vector $w \in \mathbb{R}^n$ in a unique way. In fact, traslating $\mathcal{F}(v_1, \dots, v_n)$ by each element of the lattice $\mathcal{L}(v_1, \dots, v_n)$ creates a partition of $\mathbb{R}^n$.

Proposition 3.1.6. *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and let $\mathcal{F}$ be a fundamental parallelepiped for $\mathcal{L}$. Then every vector $w \in \mathbb{R}^n$ can be uniquely written as $w = t + w$ for a unique $t \in \mathcal{F}$ and a unique $v \in \mathcal{L}$.*

As one can imagine, the shape of the fundamental parallelepiped depends strongly from the basis we choose. However, the n -volume of this parallelepiped is invariant under change of basis.

Definition 3.1.7 (Determinant (lattice)). Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and let $B = [v_1, \dots, v_n]$ be an arbitrary basis for $\mathcal{L}$. Then the determinant of the lattice $\mathcal{L}$ is defined as $\det(\mathcal{L}) = |\det(B)|$.

To show that this definition is well-posed, we have to show that it is independent of the choice of the basis. Suppose B and B' are two different bases for the same n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$. Then, thanks to Proposition 3.1.2 there exists a unimodular matrix $U \in \text{GL}_n(\mathbb{Z})$, $\det(U) = \pm 1$ such that $B' = BU$. But then

$$|\det(B')| = |\det(B)| |\det(U)| = |\det(B)|.$$

The geometric interpretation of this determinant is exactly the n -volume of the fundamental parallelepiped of the lattice. So we have the following proposition.

Proposition 3.1.8. *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice. The n -volume $\text{Vol}(\mathcal{F})$ of every fundamental parallelepiped $\mathcal{F}$ attached to $\mathcal{L}$ is an invariant under change of basis, and it is equal to the determinant $\det(\mathcal{L})$ of the lattice.*

Remark 3.1.9. If we consider an arbitrary basis $B = [v_1, \dots, v_n]$ of a n -dimensional lattice $\mathcal{L}$ and if we think of this vectors as the sides of the fundamental parallelepiped $\mathcal{F}(v_1, \dots, v_n)$, then the n -volume would be the greatest if the vectors $v_1, \dots, v_n$ were pairwise orthogonal.

This enables us to give an upper bound for the determinant of a lattice. This upper bound is the Hadamard's inequality.

Proposition 3.1.10 (Hadamard's inequality). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice, let $B = [v_1, \dots, v_n]$ be an arbitrary basis for $\mathcal{L}$. Then*

$$\det(\mathcal{L}) \leq \|v_1\|_2 \cdots \|v_n\|_2.$$

An important feature of a lattice basis is how close it is to being orthogonal, i.e. how close are its vectors to be pairwise orthogonal. This property can be measured by the tightness of Hadamard's inequality.

Definition 3.1.11 (Hadamard ratio). Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and let $B = [v_1, \dots, v_n]$ be a basis for $\mathcal{L}$. The Hadamard ratio of the basis B is the quantity

$$\text{Hr}(B) = \left(\frac{\det(\mathcal{L})}{\|v_1\|_2 \cdots \|v_n\|_2} \right)^{\frac{1}{n}}.$$

Hadamard's inequality tells us that $0 < \text{Hr}(B) \leq 1$. In particular, the closer are the vectors of the basis B to being pairwise orthogonal, the closer to 1 is the Hadamard ratio.

A classical lattice problem asks, given a basis of a lattice, to find one of the shortest non-zero vectors inside the lattice (it is not unique because a lattice is a group). Minkowski's theorem helps us give a bound for the length of this vector.

Theorem 3.1.12 (Minkowski). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice, and let $S \subseteq \mathbb{R}^n$ be a convex, symmetric and bounded region, whose n -volume satisfies*

$$\text{Vol}(S) > 2^n \det(\mathcal{L}).$$

Then S contains a non-zero vector belonging to $\mathcal{L}$. Moreover, if S is closed, the hypothesis can be weakened to request only that $\text{Vol}(S) \geq 2^n \det(\mathcal{L})$.

The bound on the length of a shortest non-zero vector is due to Hermite.

Theorem 3.1.13 (Hermite). *Every n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ contains a non-zero vector v such that*

$$\|v\|_2 \leq \sqrt{n} \det(\mathcal{L})^{1/n}$$

Proof. Let us fix an arbitrary n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$. If $B > 0$ we let

$$S = \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid -B \leq x_i \leq B, i = 1, \dots, n\}.$$

The region S is symmetric, closed, convex and bounded, and its n -volume is $\text{Vol}(S) = (2B)^n$. If we let $B = \det(\mathcal{L})^{1/n}$, we have that $\text{Vol}(S) = 2^n \det(\mathcal{L})$. Then, applying Minkowski's Theorem 3.1.12, we get that there exists a non-zero vector $v \in \mathcal{L} \cap S$. But then

$$\|v\|_2 = \sqrt{v_1^2 + \cdots + v_n^2} \leq \sqrt{n}B = \sqrt{n} \det(\mathcal{L})^{1/n}.$$

□

Now, we introduce the Gram-Schmidt orthogonalization process.

Lemma 3.1.14 (Gram-Schmidt Orthogonalization Process). *Let $B = [v_1, \dots, v_n]$ be a basis for $\mathbb{R}^n$. The following algorithm, the Gram-Schmidt Orthogonalization Algorithm, denoted as **GramSchmidt**, creates an orthogonal basis $\tilde{B} = [v_1^*, \dots, v_n^*]$ for $\mathbb{R}^n$.*

GramSchmidt: Gram-Schmidt Orthogonalization Algorithm

Input: a basis $B = [v_1, \dots, v_n]$ of $\mathbb{R}^n$
Output: an orthogonal basis $\tilde{B}$ of $\mathbb{R}^n$

```

1 :  $v_1^* = v_1$ 
2 : for  $i = 2$  to  $n$  do
3 :   for  $j = 1$  to  $i - 1$  do
4 :      $\mu_{i,j} = \frac{\langle v_i, v_j^* \rangle}{\|v_j^*\|^2}$ 
5 :   endfor
6 :    $v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*$ 
7 : endfor
8 : return  $\tilde{B} = [v_1^*, \dots, v_n^*]$ 

```

As is the case for the basis of a lattice B , it is useful to render $\tilde{B}$ as a matrix whose columns are $[v_1^*, \dots, v_n^*]$. Sometimes, we will also talk about Gram-Schmidt orthogonalization of a basis matrix B , which gives as outputs a matrix $\tilde{B}$.

Remark 3.1.15. Given a basis $B = [v_1, \dots, v_n]$ of $\mathbb{R}^n$, its orthogonalization $\tilde{B} = [v_1^*, \dots, v_n^*]$ via the Gram-Schmidt process has the following properties:

1. $\langle v_i^*, v_j^* \rangle = 0$ for all $i, j = 1, \dots, n$ such that $i \neq j$.
2. $\text{span}_{\mathbb{R}}(v_1, \dots, v_i) = \text{span}_{\mathbb{R}}(v_1^*, \dots, v_i^*)$ for all $i = 1, \dots, n$

The Gram-Schmidt orthogonalization process will be a key sub-routine of other algorithms presented in this thesis.

Given a lattice $\mathcal{L} \subseteq \mathbb{R}^n$ we can define its dual, the dual lattice.

Definition 3.1.16 (Dual lattice). Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice. Then, the dual lattice $\mathcal{L}^*$ is the set

$$\mathcal{L}^* = \{w \in \mathbb{R}^n \mid \langle w, v \rangle \in \mathbb{Z} \text{ for all } v \in \mathcal{L}\}.$$

The following proposition tells us that the dual lattice of a given lattice is indeed a lattice itself, and we can deduce its basis from the starting one.

Proposition 3.1.17. *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice with basis $B = [v_1, \dots, v_n] \in \text{GL}_n(\mathbb{R})$. Then the dual lattice $\mathcal{L}^* \subseteq \mathbb{R}^n$ is a n -dimensional lattice with basis $(B^T)^{-1} = [w_1, \dots, w_n]$.*

Proof. Let $w \in \mathcal{L}^*$. This means that $\langle w, v \rangle \in \mathbb{Z}$ for all $v \in \mathcal{L}$. But B is a basis for $\mathcal{L}$, so this is equivalent to say that $\langle w, v_i \rangle \in \mathbb{Z}$ for all $i = 1, \dots, n$, i.e. (if we interpret vectors as column vectors) $v_i^T w \in \mathbb{Z}$ for all $i = 1, \dots, n$. But then, $w \in \mathcal{L}^*$ is equivalent to $B^T w \in \mathbb{Z}^n$. So, since B is invertible, $w = (B^T)^{-1} z$ for some $z \in \mathbb{Z}^n$. This means that $w \in \mathcal{L}((B^T)^{-1})$. So $\mathcal{L}^* = \mathcal{L}((B^T)^{-1})$. This concludes the proof. $\square$

Lattices and their duals are strictly connected to one another. We report here some properties which involve their relation.

Proposition 3.1.18. *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and let $\mathcal{L}^*$ be its dual lattice. Then, we have the following.*

$$(1) (\mathcal{L}^*)^* = \mathcal{L}.$$

$$(2) \det(\mathcal{L}^*) = \frac{1}{\det(\mathcal{L})}.$$

Proof. (1) Let us consider a basis $B \in \text{GL}_n(\mathbb{R})$ of $\mathcal{L}$. Since $((B^T)^{-1})^T = B$ we have the claim.

(2) As for (1), consider a basis $B \in \text{GL}_n(\mathbb{R})$ of $\mathcal{L}$. Since $(B^T)^{-1}$ is a basis for $\mathcal{L}^*$, then

$$\det(\mathcal{L}^*) = |\det((B^T)^{-1})| = \frac{1}{|\det(B^T)|} = \frac{1}{|\det(B)|} = \frac{1}{\det(\mathcal{L})}.$$

□

Other important invariants of a lattice are the successive minima.

Given $r \in \mathbb{R}_{>0}$ and $c \in \mathbb{R}^n$, let $\mathcal{B}_n(c, r) = \{x \in \mathbb{R}^n \mid \|x - c\|_2 < r\}$ the n -dimensional open ball or radius r centered in c . The successive minima $\lambda_1, \dots, \lambda_n$ of a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ are fundamental constants defined in the following way (in the case of the standard Euclidean norm $\|\cdot\|_2$).

Definition 3.1.19 (i -th minimum). Given a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$, for every $i = 1, \dots, n$ the i -th minimum $\lambda_i(\mathcal{L})$ of the lattice $\mathcal{L}$ is the radius of the smallest n -sphere centered in the origin containing i linearly independent lattice vectors:

$$\lambda_i(\mathcal{L}) = \inf\{r \in \mathbb{R}_{>0} \mid \dim_{\mathbb{R}}(\text{span}_{\mathbb{R}}(\mathcal{L} \cap \mathcal{B}_n(0, r))) \geq i\}. \quad (3.1)$$

Since Remark 3.1.3 tells us that a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is a discrete, free, full-rank, submodule of $\mathbb{R}^n$, we know that there always exist vectors achieving the successive minima. So there are linearly independent vectors $s_1, \dots, s_n \in \mathcal{L}$ such that $\|s_i\|_2 = \lambda_i(\mathcal{L})$ for all $i = 1, \dots, n$. This means, in particular, that the infimum in (3.1) is actually a minimum if $\mathcal{B}_n(0, r)$ is replaced with the closed ball $\overline{\mathcal{B}_n(0, r)}$.

This can be summarized in the following result, which is [44, Theorem 1.2].

Theorem 3.1.20. *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice with successive minima $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Then there exist n linearly independent lattice vectors $s_1, \dots, s_n \in \mathcal{L}$ such that $\|s_i\|_2 = \lambda_i$ for all $i = 1, \dots, n$.*

In particular, $\lambda_1(\mathcal{L})$ is the length of a shortest non-zero vector in the lattice, and equals the minimum distance between any two distinct lattice points:

$$\lambda_1(\mathcal{L}) = \min_{0 \neq v \in \mathcal{L}} \|v\|_2 = \min \|v - w\|_2 \mid v, w \in \mathcal{L}, v \neq w.$$

3.2 Principal Lattice Problems and Babai's Algorithm

The Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) are the two most famous and important hard problems in the context of lattices, and the hardness of a large number of lattice-based cryptosystem is based on them or on slight variations of them.

Definition 3.2.1 (Shortest Vector Problem (SVP)). Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice with basis $B = [v_1, \dots, v_n]$. The Shortest Vector Problem (SVP) asks to find a non-zero vector $v \in \mathcal{L}$ who minimizes the euclidean norm $\|\cdot\|_2$, i.e.

$$v = \arg \min\{\|w\|_2 \mid 0 \neq w \in \mathcal{L}\}.$$

Definition 3.2.2 (Closest Vector Problem (CVP)). Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice with basis $B = [v_1, \dots, v_n]$ and let $z \in \mathbb{R}^n \setminus \mathcal{L}$. The Closest Vector Problem (CVP) asks to find a vector $v \in \mathcal{L}$ which is the closest possible to z , i.e. that minimizes the quantity $\|z - v\|_2$. This means that v has to be

$$v = \arg \min\{\|w - z\|_2 \mid w \in \mathcal{L}\}.$$

In lattice-based cryptography, one typically considers the approximation variant of these problems, which are denoted by an additional subscript $\gamma > 0$ indicating the approximation factor. For instance, in SVP_γ the goal is to find a vector whose norm is at most γ times that of a shortest nonzero vector.

Remark 3.2.3. General instances of the CVP problem have been proven to be NP-hard, and that is why they are very useful to cryptographers.

We have to be careful, though, because if the basis we are given is "too orthogonal" there exists an algorithm that can efficiently solve those problems: Babai's algorithm.

If a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ admits a basis $B = [v_1, \dots, v_n]$ where the v_i s are pairwise orthogonal, it becomes very easy to solve both SVP and CVP. For SVP, if $v \in \mathcal{L}$, we can write it in the basis B as

$$v = a_1 v_1 + \dots + a_n v_n$$

with $a_i \in \mathbb{Z}$ for all $i = 1, \dots, n$. So

$$\|v\|_2^2 = a_1^2 \|v_1\|_2^2 + \dots + a_n^2 \|v_n\|_2^2,$$

so the shortest non-zero vectors of $\mathcal{L}$ are the shortest vectors in the set

$$\{\pm v_1, \dots, \pm v_n\}.$$

Similarly, for CVP, suppose that we want the lattice vector(s) closest to a fixed target $z \in \mathbb{R}^n$. Since the vectors in B are linearly independent they are a $\mathbb{R}$ -basis for $\mathbb{R}^n$ and so there exist $t_1, \dots, t_n \in \mathbb{R}$ such that

$$z = t_1 v_1 + \dots + t_n v_n.$$

So, if

$$\mathcal{L} \ni v = a_1 v_1 + \dots + a_n v_n$$

with $a_i \in \mathbb{Z}$ for all $i = 1, \dots, n$ we have that

$$\|v - z\|_2^2 = (a_1 - t_1)^2 \|v_1\|_2^2 + \dots + (a_n - t_n)^2 \|v_n\|_2^2,$$

which is minimized by taking a_i the integer closest to t_i for all $i = 1, \dots, n$.

Babai's algorithm, whose success really depends on the "orthogonality" of the lattice bases, builds up from the previous idea.

Theorem 3.2.4 (Babai's Algorithm). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice with basis $B = [v_1, \dots, v_n]$. If the vectors of B are close to being pairwise orthogonal, the following Algorithm solves CVP.*

Babai's Algorithm

Input: a basis B of the n -dimensional lattice $\mathcal{L}$ and a target vector $z \in \mathbb{R}^n$

Output: the closest lattice vector to z

```

1 : write  $z = t_1 v_1 + \dots + t_n v_n$  with  $t_i \in \mathbb{R} \forall i = 1, \dots, n$ 
2 : for  $i = 1$  to  $n$  do
3 :    $a_i = \lfloor t_i \rceil$   $\#\#\# \lfloor \cdot \rceil$  is notation for closest integer
4 : endfor
5 : return  $v = a_1 v_1 + \dots + a_n v_n$ 
```

In general, if the vectors in the basis B are close to being pairwise orthogonal, Babai's Algorithm manages to find a valid solution to CVP_γ for some $\gamma > 0$. However, if the vectors in the basis are highly non-orthogonal, the algorithm is bound to fail. As we pointed out before, a measure for this "reasonable" orthogonality is the Hadamard ratio of the basis: if it is close to 1 Babai's algorithm will succeed, if it is close to 0 it will give us the wrong result.

There are many other lattice problems related to SVP and CVP. In the last part of this section we will define them.

Definition 3.2.5 (GapSVP). Given a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ with basis $B = [v_1, \dots, v_n]$ and a function $\psi(n)$ which varies with n , GapSVP (which is the promise version of SVP) asks, given $v \in \mathcal{L}$ a shortest non-zero vector in the lattice $\mathcal{L}$, to decide if

$$\|v\|_2 \leq 1 \text{ or } \|v\|_2 \geq \psi(n).$$

Definition 3.2.6 (GapCVP). Given a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ with basis $B = [v_1, \dots, v_n]$, a function $\psi(n)$ which varies with n , and a vector $z \in \mathbb{R}^n \setminus \mathcal{L}$, GapCVP (which is the promise version of CVP) asks, given $v \in \mathcal{L}$ a closest vector in the lattice $\mathcal{L}$ to z , to decide if

$$\|v - z\|_2 \leq 1 \text{ or } \|v - z\|_2 \geq \psi(n).$$

Definition 3.2.7 (SBP). Given a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$, the Shortest Basis Problem (SBP) asks to find a basis $B = [v_1, \dots, v_n]$ of $\mathcal{L}$ such that it is the shortest possible, meaning that it minimizes the values

$$\max_{i=1, \dots, n} \|v_i\|_2 \text{ or } \sum_{i=1}^n \|v_i\|_2^2.$$

There exists many variants of the SBP, according to the method in which one decides to measure the "length" of the basis.

The following problem is based on successive minima of a lattice.

Definition 3.2.8 (SIVP). The Shortest Independent Vectors Problem (SIVP) asks, given a basis $B = [v_1, \dots, v_n]$ of a n -dimensional lattice $\mathcal{L}(B) \subseteq \mathbb{R}^n$, to find n linearly independent lattice vectors $w_1, \dots, w_n \in \mathcal{L}$ such that $\|w_i\|_2 \leq \lambda_n(\mathcal{L}(B))$ for all $i = 1, \dots, n$.

Since a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is a discrete, free, full-rank sub-module of $\mathbb{R}^n$ (Remark 3.1.3), we can effectively define the distance between a vector $z \in \mathbb{R}^n$ and the lattice as

$$\text{dist}(z, \mathcal{L}) = \min\{\|v - z\|_2 \mid v \in \mathcal{L}\}.$$

Definition 3.2.9 (BDD). Given a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ with basis $B = [v_1, \dots, v_n]$ and a vector $z \in \mathbb{R}^n$ such that

$$\text{dist}(z, \mathcal{L}) < \frac{\lambda_1(\mathcal{L})}{2}$$

(i.e. the distance between w and the lattice $\mathcal{L}$ is strictly smaller than $\frac{\|v\|_2}{2}$ where v is a shortest non-zero vector in $\mathcal{L}$), the Bounded Distance Decoding (BDD) problem asks to find the vector $w \in \mathcal{L}$ which is the closest to z .

The BDD problem is very similar to the CVP, but in BDD the solution w is unique because have an effective bound a priori on the distance between z and the lattice $\mathcal{L}$.

In their book [9] Bernstein, Buchmann and Dahmen state a conjecture (Conjecture 1) in which they address the hardness of solving exactly and solving approximately (with a polynomial bound in

n). Some cryptographic constructions admit security proofs that are based on the hardness of solving some worst-case problems on lattices. By worst-case problems on lattices we mean that breaking a cryptographic system (also with some small but non-negligible probability) is at least as hard as solve some problems on lattices (approximately, with a polynomial bound) in their worst-case. In other words, breaking a lattice-based cryptosystem means having an efficient algorithm that allows us to solve every instance of a underlying lattice problem (in most cases the underlying problem is SVP or SVP_γ with $\gamma = p(n)$, p a polynomial in n).

3.3 Gauss Reduction Algorithm and LLL Reduction Algorithm

The algorithm to find an optimal, short and quasi-orthogonal basis for a 2-dimensional lattice is due to Gauss. The key idea consists in subtracting alternatively a multiple of a basis vector to the other one, thus reducing at every step the length of one of the vectors, until no further improvement is possible.

Suppose that $\mathcal{L} \subseteq \mathbb{R}^2$ is a 2-dimensional lattice with basis $B = [v_1, v_2]$ and, up to relabeling, suppose that $\|v_1\|_2 < \|v_2\|_2$. We want to shorten v_2 by subtracting from it a certain multiple of v_1 . If we could subtract an arbitrary multiple of v_1 we would replace v_2 with

$$v_2^* = v_2 - \frac{\langle v_1, v_2 \rangle}{\|v_1\|_2^2} v_1,$$

as it is the case in the Gram-Schmidt orthogonalization process. However, in general, the number $\frac{\langle v_1, v_2 \rangle}{\|v_1\|_2^2}$ is not an integer, and thus the vector $\frac{\langle v_1, v_2 \rangle}{\|v_1\|_2^2} v_1$ is not, in general, a vector inside $\mathcal{L}$. The best possible compromise consists in substituting v_2 with the vector $v_2 - mv_1$, where

$$m = \left\lfloor \frac{\langle v_1, v_2 \rangle}{\|v_1\|_2^2} \right\rfloor,$$

This new couple of vectors is again a basis for $\mathcal{L}$, but the norm of $v_2 - mv_1$ is smaller than the one of v_2 . After this step, if $\|v_2 - mv_1\|_2 > \|v_1\|_2$, then the algorithm stops. If that is not the case the two vectors are swapped and the reduction step is repeated. Gauss proved that this algorithm returns a good basis for $\mathcal{L}$, as the following result shows. More details can be found in [44, Theorem 2.5].

Proposition 3.3.1 (Gauss Reduction for 2-dimensional lattices). *Let $\mathcal{L} \subseteq \mathbb{R}^2$ be a 2-dimensional lattice with basis $B = [v_1, v_2]$. The following algorithm returns a (good) basis for $\mathcal{L}$.*

Gauss Reduction Algorithm

Input: a basis $B = [v_1, v_2]$ of the 2-dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^2$

Output: a basis B' of $\mathcal{L} \subseteq \mathbb{R}^2$

```

1 : while  $\|v_2\|_2 < \|v_1\|_2$  do
2 :   swap  $v_1$  and  $v_2$ 
3 :    $m = \left\lfloor \frac{\langle v_1, v_2 \rangle}{\|v_1\|_2^2} \right\rfloor$ 
4 :   if  $m = 0$  then
5 :     return  $B' = [v_1, v_2]$ 
6 :   else
7 :      $v_2 = v_2 - mv_1$ 
8 :   endif
9 : endwhile
10 : return  $B' = [v_1, v_2]$ 

```

In particular, when the algorithm stops, the vector v_1 is a shortest non-zero vector in $\mathcal{L}$, thus the algorithm solves SVP. Moreover, the angle θ between v_1 and v_2 in B' verifies the inequality $|\cos(\theta)| \leq \frac{\|v_1\|_2}{2\|v_2\|_2}$, i.e. $\frac{\pi}{3} \leq \theta \leq \frac{2\pi}{3}$.

The previous proposition, i.e. Gauss Reduction Algorithm, shows how to explicitly find a shortest non-zero vector in a dimension $n = 2$ lattice. However, when the dimension n grows, SVP and CVP become much harder to solve. A substantial progress in this area is due to the LLL Reduction Algorithm, created by Lenstra, Lenstra and Lovász [40] in 1982.

Given an arbitrary basis $B = [v_1, \dots, v_n]$ of a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$, the goal of the LLL Reduction Algorithm is to transform it into a better basis, using the same idea of the Gauss Reduction Algorithm, i.e. creating a basis whose vectors are ordered according to their length (in particular, v_1 will solve an approximate version of SVP) but also pairwise quasi-orthogonal.

We remind the reader that, thanks to Hadamard's inequality (Proposition 3.1.10),

$$\det(\mathcal{L}) = \text{Vol}(\mathcal{F}) \leq \|v_1\|_2 \cdots \|v_n\|_2,$$

where $\text{Vol}(\mathcal{F})$ is the n -volume of a fundamental parallelepiped $\mathcal{F}$ for $\mathcal{L}$. If the pairwise quasi-orthogonality of these vectors gets bigger, then the difference between the RHS and the LHS of the inequality becomes smaller.

In order to create this new basis, we have to compute another one. From $B = [v_1, \dots, v_n]$ we perform the Gram-Schmidt orthogonalization, as described in Lemma 3.1.14. We start putting $v_1^* = v_1$. Then, for $i = 2, \dots, n$ we have

$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*, \text{ where } \mu_{i,j} = \frac{\langle v_i, v_j^* \rangle}{\|v_j^*\|_2^2} \text{ for } 1 \leq j \leq i-1. \quad (3.2)$$

The basis $\tilde{B} = [v_1^*, \dots, v_n^*]$ is a basis for the $\mathbb{R}$ -vector space generated by $B = [v_1, \dots, v_n]$, but in general $\tilde{B}$ is not a lattice basis for $\mathcal{L}(B)$, since the Gram-Schmidt process involves linear combinations of the v_i s with coefficients $\mu_{i,j}$ which, in general, are not all in $\mathbb{Z}$. However, the next result shows that the determinant is an invariant for basis constructed in this way.

Proposition 3.3.2. Let $B = [v_1, \dots, v_n]$ be a basis for the n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and let $\tilde{B} = [v_1^*, \dots, v_n^*]$ the associated Gram-Schmidt orthogonalized basis. Then

$$\det(\mathcal{L}) = \prod_{i=1}^n \|v_i^*\|_2.$$

Proof. Let B be seen as the matrix $[v_1, \dots, v_n]$ (with the usual convention of putting the vectors v_i as columns). We know that $\det(\mathcal{L}) = |\det(B)|$. Analogously, we interpret the basis $\tilde{B}$ as the Gram-Schmidt orthogonalized vectors in column. From the equation (3.2) we get that $B = \tilde{B}M$, where M is the change of basis matrix

$$M = \begin{pmatrix} 1 & \mu_{21} & \mu_{31} & \cdots & \mu_{n-1,1} & \mu_{n,1} \\ 0 & 1 & \mu_{3,2} & \cdots & \mu_{n-1,2} & \mu_{n,2} \\ 0 & 0 & 1 & \cdots & \mu_{n-1,3} & \mu_{n,3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \mu_{n,n-1} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Then $\det(M) = 1$ and

$$\det(\mathcal{L}) = |\det(B)| = |\det(\tilde{B})| |\det(M)| = |\det(\tilde{B})|.$$

Since the column vectors of $\tilde{B}$ are pairwise orthogonal, then $|\det(\tilde{B})| = \prod_{i=1}^n \|v_i^*\|_2$, and the proof is concluded. $\square$

We can interpret the vectors v_i^* using the concept of orthogonal complement of a vector subspace. Intuitively, the Gram-Schmidt process can be described in the following way. For $i = 1, \dots, n$

$$v_i^* = \text{Projection of } v_i \text{ onto } \text{span}_{\mathbb{R}}(v_1, \dots, v_{i-1})^\perp.$$

Even if $\tilde{B} = [v_1^*, \dots, v_n^*]$ is not a basis for the lattice $\mathcal{L}$, we use this list of vectors to define a new concept of basis, which is very important for the LLL Algorithm.

Definition 3.3.3 (LLL-reduced Basis). Let $B = [v_1, \dots, v_n]$ be a basis for a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and let $\tilde{B} = [v_1^*, \dots, v_n^*]$ be the associated Gram-Schmidt orthogonalized basis. We say that the basis B is LLL-reduced if the following conditions are met:

$$|\mu_{i,j}| = \frac{|\langle v_i, v_j^* \rangle|}{\|v_j^*\|_2^2} \leq \frac{1}{2} \text{ for all } 1 \leq j < i \leq n, \quad (3.3)$$

$$\|v_i^*\|_2^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2 \right) \|v_{i-1}^*\|_2^2 \text{ for all } 1 < i \leq n. \quad (3.4)$$

Equation (3.3) is known as size condition, while equation (3.4) is known as Lovász condition.

There are different equivalent ways to define condition (3.4). It is equivalent to the inequality

$$\|v_i^* + \mu_{i,i-1} v_{i-1}^*\|_2^2 \geq \frac{3}{4} \|v_{i-1}^*\|_2^2,$$

and it is also equivalent to the request that

$$\left\| \text{Projection of } v_i \text{ onto } \text{span}_{\mathbb{R}}(v_1, \dots, v_{i-2})^\perp \right\| \geq \frac{3}{4} \left\| \text{Projection of } v_{i-1} \text{ onto } \text{span}_{\mathbb{R}}(v_1, \dots, v_{i-2})^\perp \right\|.$$

The main result of Lenstra, Lenstra and Lovász in [40] tells that a LLL-reduced basis is a good basis, and it is possible to compute it in polynomial time.

Theorem 3.3.4. Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice. An arbitrary LLL-reduced basis $B = [v_1, \dots, v_n]$ for $\mathcal{L}$ satisfies the following conditions.

- (1) $\prod_{i=1}^n \|v_i\|_2 \leq 2^{\frac{n(n-1)}{4}} \det(\mathcal{L})$.
- (2) $\|v_j\|_2 \leq 2^{\frac{j-1}{2}} \|v_i^*\|_2$ for all $1 \leq j \leq i \leq n$.

Moreover, for the first vector v_1 of a LLL-reduced basis we have that:

- (3) $\|v_1\|_2 \leq 2^{\frac{n-1}{4}} |\det(\mathcal{L})|^{\frac{1}{n}}$.
- (4) $\|v_1\|_2 \leq 2^{\frac{n-1}{2}} \min_{0 \neq v \in \mathcal{L}} \|v\|_2$.

Then, in particular, a LLL-reduced basis solves SVP_γ with $\gamma = 2^{\frac{n-1}{2}}$.

Proof. (1) The Lovász condition 3.4 and the fact that $|\mu_{i,j}| \leq \frac{1}{2}$ imply that

$$\|v_i^*\|_2^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|v_{i-1}^*\|_2^2 \geq \frac{1}{2} \|v_{i-1}^*\|_2^2.$$

Applying this inequality multiple times we get that

$$\|v_j^*\|_2^2 \leq 2^{i-j} \|v_i^*\|_2^2. \quad (3.5)$$

Thus we have a bound for the norm of v_i :

$$\begin{aligned} \|v_i\|_2^2 &= \left\| v_i^* + \sum_{j=1}^{i-1} \mu_{i,j} v_j^* \right\|_2^2 = \text{since } v_1^*, \dots, v_n^* \text{ are pairwise orthogonal} \\ &= \|v_i^*\|_2^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|v_j^*\|_2^2 \leq \text{from (3.3)} \\ &\leq \|v_i^*\|_2^2 + \sum_{j=1}^{i-1} \frac{1}{4} \|v_j^*\|_2^2 \leq \text{from (3.5)} \\ &\leq \|v_i^*\|_2^2 + \sum_{j=1}^{i-1} 2^{i-j-2} \|v_i^*\|_2^2 = \left(1 + 2^{i-2} \sum_{j=1}^{i-1} \frac{1}{2^j}\right) \|v_i^*\|_2^2 = \frac{1 + 2^{i-1}}{2} \|v_i^*\|_2^2 \leq \\ &\leq 2^{i-1} \|v_i^*\|_2^2, \end{aligned}$$

where the first equality descends from (3.2). Then

$$\|v_i\|_2^2 \leq 2^{i-1} \|v_i^*\|_2^2. \quad (3.6)$$

Multiplying equation (3.6) over $i = 1, \dots, n$ we get

$$\prod_{i=1}^n \|v_i\|_2^2 \leq \prod_{i=1}^n 2^{i-1} \|v_i^*\|_2^2 = 2^{\frac{n(n-1)}{2}} \prod_{i=1}^n \|v_i^*\|_2^2 = 2^{\frac{n(n-1)}{2}} \det(\mathcal{L})^2,$$

where, for the last equality, we used Proposition 3.3.2. Extracting the square roots from both sides of the inequality we get the thesis.

(2) For every $j \leq i$, we use equation (3.6) (with $i = j$) and equation (3.5) to get

$$\|v_j\|_2^2 \leq 2^{j-1} \|v_j^*\|_2^2 \leq 2^{j-1} \cdot 2^{i-j} \|v_i^*\|_2^2 = 2^{i-1} \|v_i^*\|_2^2.$$

Extracting the square roots, we get the thesis.

(3) We start from (2), putting $j = 1$, we multiply over $i = 1, \dots, n$ and we use Proposition 3.3.2 to get

$$\|v_1\|_2^n \leq \prod_{i=1}^n 2^{\frac{i-1}{2}} \|v_i^*\|_2 = 2^{\frac{n(n-1)}{4}} \prod_{i=1}^n \|v_i^*\|_2 = 2^{\frac{n(n-1)}{4}} \det(\mathcal{L}).$$

We conclude extracting the n -th roots.

(4) Let $0 \neq v \in \mathcal{L}$ a non-zero vector in the lattice. Then

$$v = \sum_{j=1}^i a_j v_j = \sum_{j=1}^i b_j v_j^*,$$

with $a_1, \dots, a_i \in \mathbb{Z}$, $a_i \neq 0$ and $b_1, \dots, b_i \in \mathbb{R}$. In particular $|a_i| \geq 1$. By construction, we know that for every k the vectors $v_1^*, \dots, v_k^*$ are pairwise orthogonal and $\text{span}_{\mathbb{R}}(v_1, \dots, v_k) = \text{span}_{\mathbb{R}}(v_1^*, \dots, v_k^*)$ (Remark 3.1.15). Then

$$\langle v, v_i^* \rangle = a_i \langle v_i, v_i^* \rangle = b_i \langle v_i^*, v_i^* \rangle \text{ and } \langle v_i, v_i^* \rangle = \langle v_i^*, v_i^* \rangle,$$

from which we get that $a_i = b_i$. Then, $|b_i| = |a_i| \geq 1$. Using this fact and (2) (with $j = 1$) we get that

$$\|v\|_2^2 = \sum_{j=1}^i b_j \|v_j^*\|_2^2 \geq b_i^2 \|v_i^*\|_2^2 \geq \|v_i^*\|_2^2 \geq 2^{-(i-1)} \|v_1\|_2^2 \geq 2^{-(n-1)} \|v_1\|_2^2.$$

Extracting the square roots we have the thesis. □

We can now write the pseudo-code for the LLL Reduction Algorithm.

LLL: LLL Reduction Algorithm

Input: a basis $B = [v_1, \dots, v_n] (\in \text{GL}_n(\mathbb{R}))$ of the n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$
Output: a LLL-reduced basis $\tilde{B}$ of $\mathcal{L} \subseteq \mathbb{R}^n$

```
1 : for  $i = 1$  to  $n$  do
2 :    $v = v_i$ 
3 :   for  $j = i - 1$  down to  $1$  do
4 :      $\mu_{i,j} = \frac{\langle v_i, v_j^* \rangle}{sn_i^*}$ 
5 :      $v = v - \mu_{i,j} v_j^*$ 
6 :   endfor
7 :    $v_i^* = v$ 
8 :    $sn_i^* = \|v_i^*\|_2^2$ 
9 : endfor
10 :  $k = 2$ 
11 : while  $k \leq n$  do
12 :   for  $j = k - 1$  down to  $1$  do
13 :      $v_k = v_k - \lfloor \mu_{k,j} \rfloor v_j$ 
14 :     for  $\ell = 1$  to  $j - 1$  do
15 :        $\mu_{k,\ell} = \mu_{k,\ell} - \lfloor \mu_{k,j} \rfloor \mu_{j,\ell}$ 
16 :     endfor
17 :      $\mu_{k,j} = \mu_{k,j} - \lfloor \mu_{k,j} \rfloor$ 
18 :   endfor
19 :   if  $sn_k^* \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right) sn_{k-1}^*$  then
20 :      $k = k + 1$ 
21 :   else
22 :     swap  $v_k$  and  $v_{k-1}$ 
23 :     update  $v_k^*, v_{k-1}^*, sn_k^*, sn_{k-1}^*, \mu_{k,k-1}$ 
24 :     for  $j = 1$  to  $k - 2$  do
25 :       update  $\mu_{k-1,j}, \mu_{k,j}$ 
26 :     endfor
27 :     for  $i = k + 1$  to  $n$  do
28 :       update  $\mu_{i,k}, \mu_{i,k-1}$ 
29 :     endfor
30 :      $k = \max\{2, k - 1\}$ 
31 :   endif
32 : endwhile
33 : return  $B' = [v_1, \dots, v_n]$ 
```

Arguments to prove the following theorem, which goes beyond the purpose of this thesis, can be found (for integer lattices) in [44, Section 2.2].

Theorem 3.3.5 (LLL Reduction Algorithm). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and $B = [v_1, \dots, v_n] \in \text{GL}_n(\mathbb{R})$ a basis for $\mathcal{L}$. The LLL Reduction Algorithm, denoted with **LLL**, terminates*

after a finite number of steps and return a LLL-reduced basis for $\mathcal{L}$. More precisely, let $MB = \max\{\|v_1\|_2, \dots, \|v_n\|_2\}$. Then the LLL Reduction Algorithm performs the main loop (lines 11-32) at most $\mathcal{O}(n^2 \log(n) + n^2 \log(MB))$ times. In particular, the LLL Reduction Algorithm is polynomial-time.

Summing up, the LLL Reduction Algorithm is polynomial-time and it can be used to solve SVP_γ instances where $\gamma = 2^{\frac{n-1}{2}} \mathcal{O}(2^n)$ is exponential in the size n of the lattice.

3.4 Nearest Plane Algorithm and Micciancio-Goldwasser Algorithm

We have just seen that the LLL Reduction Algorithm enables us to solve in polynomial time the SVP_γ with $\gamma = \mathcal{O}(2^n)$. A natural question now is the following: is there an algorithm that allows us to solve an approximate version of the CVP? The answer is positive and the algorithm relies on the power of the LLL construction.

The Nearest Plane Algorithm is the algorithm that allows us to solve CVP_γ , and is due to Babai [5].

NearestPlane: Nearest Plane Algorithm

Input: a basis $B = [v_1, \dots, v_n] (\in \text{GL}_n(\mathbb{R}))$ of the n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and a target vector $z \in \mathbb{R}^n$

Output: a vector $v \in \mathcal{L}$ such that $\|v - z\|_2 \leq 2^{\frac{n}{2}} \text{dist}(z, \mathcal{L})$

```

1:  $B' = [v'_1, \dots, v'_n] = \mathbf{LLL}(B)$ 
2:  $\tilde{B}' = [v'^*_1, \dots, v'^*_n] = \mathbf{GramSchmidt}(B')$ 
3:  $t = z$ 
4: for  $i = n$  down to 1 do
5:    $t = t - \left\lfloor \frac{\langle t, v'^*_i \rangle}{\|v'^*_i\|_2^2} \right\rfloor v'_i$ 
6: endfor
7: return  $v = z - t$ 
```

In Regev's lecture notes [51] one can find the proof of the following theorem (in the case of an integer lattice and an integer target).

Theorem 3.4.1 (Nearest Plane Algorithm). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a n -dimensional lattice and $B = [v_1, \dots, v_n]$ a basis for $\mathcal{L}$. The Nearest Plane Algorithm is polynomial-time and solves CVP_γ with $\gamma = 2^{\frac{n-1}{2}}$.*

So, Babai's Nearest Plane Algorithm is in some way a compendium to the LLL Reduction Algorithm. While the latter allows us to solve approximate instances of SVP, the former solves the approximate version of CVP.

These two algorithms are the most known and useful ones in the mathematical area related to computations on lattices. We will need them both later in the thesis.

The last algorithm we will need does not have an accepted widespread name, but its importance and ubiquitousness are undeniable. The algorithm we are talking about is the one described by Micciancio and Goldwasser in [44, Lemma 7.1]. In honor of them we decided to call it Micciancio-Goldwasser Algorithm.

The Micciancio-Goldwasser Algorithm allows, given a basis of an integer lattice and a full rank set of linearly independent vectors inside the lattice, to retrieve a basis of the lattice whose vectors have length upper bounded by the one of the independent vectors.

In the algorithm, we will need to compute the Hermite normal form of a matrix $Q \in M_n(\mathbb{Z})$. This procedure is analogous to the row echelon form reduction, only that this time we want to remain with coefficients in $\mathbb{Z}$. This means that in the row reduction we only take integer multiple of rows: we obtain an upper triangular matrix $T \in M_n(\mathbb{Z})$ and an invertible matrix $U \in GL_n(\mathbb{Z})$ such that $T = U \cdot Q$.

Theorem 3.4.2 (Micciancio-Goldwasser Algorithm). *There is a polynomial-time algorithm, denoted with $\mathbf{MG}$, that, on input a lattice basis $B = [b_1, \dots, b_n] = B \in M_n(\mathbb{Z})$ (as columns) and linearly independent vectors $S = [s_1, \dots, s_n] = S \subseteq \mathcal{L}(B)$ (as columns, so $S \in M_n(\mathbb{Z})$) such that (up to reordering) $\|s_1\|_2 \leq \|s_2\|_2 \leq \dots \leq \|s_n\|_2$, outputs a basis $R' = [r'_1, \dots, r'_n] \in M_n(\mathbb{Z})$ (as columns) equivalent to B (i.e. $\mathcal{L}(R') = \mathcal{L}(B)$, i.e. $|\det(R')| = |\det(B)|$) such that*

$$\|r'_k\|_2 \leq \max \left\{ \frac{\sqrt{k}}{2} \|s_k\|_2, \|s_k\|_2 \right\}$$

for all $k = 1, \dots, n$. Moreover, the new basis R' satisfies

$$\text{span}_{\mathbb{R}}(r'_1, \dots, r'_k) = \text{span}_{\mathbb{R}}(s_1, \dots, s_k) \text{ and } \|r'_k\|_2 \leq \|s_k^*\|_2$$

for all $k = 1, \dots, n$.

Proof. We remind that, for column vectors $C = [c_1, \dots, c_n] \in M_n(\mathbb{Z})$, we have that $C' = [c_1^*, \dots, c_n^*]$ are the Gram-Schmidt orthogonalized vectors. It is worth to note that the vectors we obtain are in general not normalized.

Since the vectors s_i , $i = 1, \dots, n$, belong to the lattice $\mathcal{L}(B)$, for $i = 1, \dots, n$ there exists $q_i \in \mathbb{Z}^n$ such that $s_i = Bq_i$. Let $Q = (q_1, \dots, q_n) \in M_n(\mathbb{Z})$. Because $s_1, \dots, s_n$ are linearly independent and have integer entries, we have that $\det(S) \in \mathbb{Z} \setminus \{0\}$. Now, we transform Q into an upper triangular integer matrix performing a sequence of special elementary row operations (i.e. adding an integer multiple of a row to some other row, multiplying a row by -1 or exchanging the order of two rows), and perform the corresponding sequence of column operations to B . Equivalently, we find (via the Hermite normal form), a matrix $U \in GL_n(\mathbb{Z})$ (i.e. $U \in M_n(\mathbb{Z})$, $\det(U) \in \{\pm 1\}$) such that $T = UQ$ is upper triangular. Then, we compute $R = BU^{-1}$. Since $|\det(U)| = 1$ and U has integer coordinates, we have that $\mathcal{L}(R) = \mathcal{L}(B)$, that is R is a basis of $\mathcal{L}(B)$. Moreover

$$S = BU^{-1} \cdot UQ = R \cdot T, \quad (3.7)$$

so, because T is upper triangular, $\text{span}_{\mathbb{R}}(s_1, \dots, s_k) = \text{span}_{\mathbb{R}}(r_1, \dots, r_k)$ for all $k = 1, \dots, n$. In fact, from (3.7), we get that, for all $k = 1, \dots, n$

$$s_k = \sum_{i=1}^k t_{i,k} r_i, \quad (3.8)$$

where $t_{i,j} = (T)_{i,j}$ for all $1 \leq i, j \leq n$.

Now, we claim that $s_k^* = t_{k,k} r_k^*$ for all $k = 1, \dots, n$. We prove it by strong induction on k .

(Base case $k = 1$). Thanks to (3.8), we have that

$$s_1^* = s_1 = t_{1,1} r_1 = t_{1,1} r_1^*.$$

(Induction step). We suppose that the claim is true for all $i = 1, \dots, k$. We prove it for $k+1$. Remember that, from (3.8), we have that $s_{k+1} = s_k + t_{k+1,k+1} r_{k+1}$. So

$$s_{k+1}^* = s_{k+1} - \sum_{i=1}^k \frac{\langle s_{k+1}, s_i^* \rangle}{\langle s_i^*, s_i^* \rangle} s_i^* =$$

$$\begin{aligned}
&= s_k + t_{k+1,k+1}r_{k+1} - \sum_{i=1}^k \frac{\langle s_k + t_{k+1,k+1}r_{k+1}, t_{i,i}r_i^* \rangle}{\langle t_{i,i}r_i^*, t_{i,i}r_i^* \rangle} t_{i,i}r_i^* = \\
&= s_k + t_{k+1,k+1}r_{k+1} - \sum_{i=1}^k \frac{\langle s_k + t_{k+1,k+1}r_{k+1}, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* = \\
&= t_{k+1,k+1}r_{k+1} - t_{k+1,k+1} \sum_{i=1}^k \frac{\langle r_{k+1}, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* + s_k - \sum_{i=1}^k \frac{\langle s_k, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* = \\
&= t_{k+1,k+1} \left(r_{k+1} - \sum_{i=1}^k \frac{\langle r_{k+1}, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* \right) + s_k - \sum_{i=1}^k \frac{\langle s_k, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* = \\
&= t_{k+1,k+1}r_{k+1}^* + s_k - \sum_{i=1}^k \frac{\langle s_k, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* = \\
&= t_{k+1,k+1}r_{k+1}^* + s_k - s_k = \\
&= t_{k+1,k+1}r_{k+1}^*,
\end{aligned}$$

where the penultimate equality follows from the fact that $s_k \in \text{span}_{\mathbb{R}}(r_1, \dots, r_k) = \text{span}_{\mathbb{R}}(r_1^*, \dots, r_k^*)$. So this concludes the induction step and proves the claim.

From this fact, we get that

$$\|r_k^*\|_2 = \frac{\|s_k^*\|_2}{|t_{k,k}|} \leq \|s_k^*\|_2 \text{ for all } k = 1, \dots, n,$$

because $t_{k,k}$ is a non-zero integer for all $k = 1, \dots, n$. We still need to check that

$$\|r_k\|_2 \leq \max \left\{ \frac{\sqrt{k}}{2} \|s_k\|_2, \|s_k\|_2 \right\}$$

for all $k = 1, \dots, n$.

We modify the vectors $r_1, \dots, r_n$ sequentially as follows.

We begin with $r'_1 = r_1$. We have that $\|r'_1\|_2 \leq \|s_1^*\|_2$, $\|r'_1\|_2 = \|r_1\|_2 \leq \|s_1^*\|_2 = \|s_1\|_2$ and $\text{span}_{\mathbb{R}}(r'_1) = \text{span}_{\mathbb{R}}(s_1)$ (and $\mathcal{L}(r'_1) = \mathcal{L}(r_1)$).

Suppose that we have defined $r'_1, \dots, r'_{k-1}$ for $k \in \{2, \dots, n\}$ such that

$$(a_1, \dots, a_n) = (r'_1, \dots, r'_{k-1}, r_k, \dots, r_n)$$

is a basis for $\mathcal{L}(B)$ such that

$$\text{span}_{\mathbb{R}}(a_1, \dots, a_v) = \text{span}_{\mathbb{R}}(s_1, \dots, s_v) = \text{span}_{\mathbb{R}}(r_1, \dots, r_v)$$

for all $v = 1, \dots, n$. Suppose also that we have defined them in a way that $\|r'_i\|_2 \leq \|s_i^*\|_2$ for all $i = 1, \dots, k-1$. Please note that $\mathcal{L}(r'_1, \dots, r'_{k-1}) = \mathcal{L}(r_1, \dots, r_{k-1})$.

Let us define r'_k .

If $r_k^* = s_k^*$ or $r_k^* = -s_k^*$, then we define $r'_k = s_k$, and $(r'_1, \dots, r'_k, r_{k+1}, \dots, r_n)$ is still a basis for $\mathcal{L}(B)$. In fact, from the previous claim, i.e. $s_k^* = t_{k,k}r_k^*$, and our hypothesis, we get that $t_{k,k} = \pm 1$. So, thanks to (3.8), we have

$$r'_k = s_k = t_{1,k}r_1 + \dots + t_{k-1,k}r_{k-1} + t_{k,k}r_k = t_{1,k}r_1 + \dots + t_{k-1,k}r_{k-1} \pm r_k.$$

So $r'_k \pm r_k \in \mathcal{L}(r_1, \dots, r_{k-1}) = \mathcal{L}(r'_1, \dots, r'_{k-1})$. This tells us that

$$(a'_1, \dots, a'_n) = (r'_1, \dots, r'_{k-1}, r'_k, r_{k+1}, \dots, r_n)$$

is still a basis for $\mathcal{L}(B)$ such that

$$\text{span}_{\mathbb{R}}(a'_1, \dots, a'_v) = \text{span}_{\mathbb{R}}(s_1, \dots, s_v) = \text{span}_{\mathbb{R}}(r_1, \dots, r_v)$$

for all $v = 1, \dots, n$. Moreover, r'_k clearly satisfies $\|r'_k\|_2 \leq \|s_k\|_2$ and $\|r'_k\|_2 \leq \|s_k^*\|_2$.

Conversely, if $r_k^* \neq \pm s_k^*$, then $s_k^* = t_{k,k} r_k^*$ with $t_{k,k}$ integer such that $|t_{k,k}| \geq 2$. Now, we consider the projection $r_k - r_k^*$ of r_k onto the $\text{span}_{\mathbb{R}}(r_1, \dots, r_{k-1}) = \text{span}_{\mathbb{R}}(r'_1, \dots, r'_{k-1})$, and we use Babai's Nearest Plane Algorithm (Theorem 3.4.1) to find a lattice point $v \in \mathcal{L}(r'_1, \dots, r'_{k-1}) = \mathcal{L}(r_1, \dots, r_{k-1})$ within distance $\frac{\sqrt{\sum_{i=1}^{k-1} \|r'_i\|_2^2}}{2}$ from $r_k - r_k^*$.

We claim that $\|r_k - v\|_2$ is at most $\frac{\sqrt{k}}{2} \|s_k\|_2$. The vector $r_k - v$ can be written as the sum of the two components r_k^* and $(r_k - r_k^*) - v$. These components are orthogonal to one another because

$$\langle r_k^*, (r_k - r_k^*) - v \rangle = \langle r_k^*, r_k - r_k^* \rangle - \langle r_k^*, v \rangle = 0 - 0 = 0,$$

where $\langle r_k^*, r_k - r_k^* \rangle = 0$ by definition of projection, and $\langle r_k^*, v \rangle = 0$ because $v \in \text{span}_{\mathbb{R}}(r_1, \dots, r_{k-1})$ and $r_k^* \perp \text{span}_{\mathbb{R}}(r_1, \dots, r_{k-1})$ by definition.

The first component r_k^* has length at most

$$\|r_k^*\|_2 = \frac{\|s_k^*\|_2}{|t_{k,k}|} \leq \frac{\|s_k^*\|_2}{2} \leq \frac{\|s_k\|_2}{2},$$

where we use the fact that Gram-Schmidt orthogonalization always reduces or maintains equal the norm of vectors. The second component $(r_k - r_k^*) - v$, by the choice of v has length at most

$$\|(r_k - r_k^*) - v\|_2 \leq \frac{\sqrt{\sum_{i=1}^{k-1} \|r'_i\|_2^2}}{2} \leq \frac{\sqrt{\sum_{i=1}^{k-1} \|s_i^*\|_2^2}}{2} \leq \frac{\sqrt{\sum_{i=1}^{k-1} \|s_i\|_2^2}}{2} \leq \frac{\sqrt{k-1} \max_{i=1, \dots, k-1} \|s_i\|_2}{2} \leq \frac{\sqrt{k-1} \cdot \|s_k\|_2}{2},$$

where we used the fact that $\|r'_i\|_2 \leq \|s_i^*\|_2$ for all $i = 1, \dots, k-1$, the fact that Gram-Schmidt orthogonalization always reduces or maintains equal the norm of vectors and the fact that by hypothesis the vectors $s_i, i = 1, \dots, n$, are ordered by ascending norm. It follows that

$$\begin{aligned} \|r_k - v\|_2 &= \|r_k^* + (r_k - r_k^*) - v\|_2 = \sqrt{\|r_k^*\|_2^2 + \|(r_k - r_k^*) - v\|_2^2} \leq \\ &\leq \sqrt{\frac{\|s_k\|_2^2}{4} + \frac{(k-1)\|s_k\|_2^2}{4}} = \frac{\sqrt{k}}{2} \|s_k\|_2. \end{aligned}$$

So, if we put $r'_k = r_k - v$, $(a''_1, \dots, a''_n) = (r'_1, \dots, r'_{k-1}, r'_k, r_{k+1}, \dots, r_n)$ we obtain a basis that satisfies all the properties in the statement (up to k). In fact:

- $(a''_1, \dots, a''_n)$ is still a basis for $\mathcal{L}(B)$: we have that $r'_k = r_k - v$ and $-v \in \mathcal{L}(r'_1, \dots, r'_{k-1})$ (because v does). So $r'_k - r_k = -v \in \mathcal{L}(r'_1, \dots, r'_{k-1}) = \mathcal{L}(r_1, \dots, r_{k-1})$.
- $\|r'_i\|_2 \leq \max \left\{ \frac{\sqrt{i}}{2} \|s_i\|_2, \|s_i\|_2 \right\}$ for all $i = 1, \dots, k$: the ones before are fine and we have just proved that $\|r'_k\|_2 \leq \frac{\sqrt{k}}{2} \|s_k\|_2$.
- $\text{span}_{\mathbb{R}}(r'_1, \dots, r'_i) = \text{span}_{\mathbb{R}}(s_1, \dots, s_i)$ for all $i = 1, \dots, k$: at each step we have that

$$\text{span}_{\mathbb{R}}(r'_1, \dots, r'_i) = \text{span}_{\mathbb{R}}(r_1, \dots, r_i) = \text{span}_{\mathbb{R}}(s_1, \dots, s_i).$$

- $\|r'_i\|_2 \leq \|s_i^*\|_2$ for all $i = 1, \dots, k$: we have that $\|r'_i\|_2 \leq \|s_i^*\|_2$ for all $i = 1, \dots, k-1$, and $r'_k = r_k - v$ with $v \in \mathcal{L}(r'_1, \dots, r'_{k-1})$. This, in particular, tells us that $v \in \text{span}_{\mathbb{R}}(r'_1, \dots, r'_{k-1}) = \text{span}_{\mathbb{R}}(r_1^*, \dots, r_{k-1}^*)$. So, we have that

$$\begin{aligned}
\|r'_k\|_2 &= \|(r_k - v)^*\|_2 = \\
&= \left\| r_k - v - \sum_{i=1}^{k-1} \frac{\langle r_k - v, r_i'^* \rangle}{\langle r_i'^*, r_i'^* \rangle} r_i'^* \right\|_2 = \\
&= \left\| \left(r_k - \sum_{i=1}^{k-1} \frac{\langle r_k, r_i'^* \rangle}{\langle r_i'^*, r_i'^* \rangle} r_i'^* \right) - \left(v - \sum_{i=1}^{k-1} \frac{\langle v, r_i'^* \rangle}{\langle r_i'^*, r_i'^* \rangle} r_i'^* \right) \right\|_2 = \\
&= \left\| \left(r_k - \sum_{i=1}^{k-1} \frac{\langle r_k, r_i'^* \rangle}{\langle r_i'^*, r_i'^* \rangle} r_i'^* \right) - (v - v) \right\|_2 = \\
&= \left\| \left(r_k - \sum_{i=1}^{k-1} \frac{\langle r_k, r_i'^* \rangle}{\langle r_i'^*, r_i'^* \rangle} r_i'^* \right) \right\|_2 = \left\| \left(r_k - \sum_{i=1}^{k-1} \frac{\langle r_k, r_i^* \rangle}{\langle r_i^*, r_i^* \rangle} r_i^* \right) \right\|_2 = \\
&= \|r_k^*\|_2 \leq \|s_k^*\|_2,
\end{aligned}$$

where in the equality in the penultimate line we used the fact that

$$\text{span}_{\mathbb{R}}(r_1'^*, \dots, r_{k-1}'^*) = \text{span}_{\mathbb{R}}(r'_1, \dots, r'_{k-1}) = \text{span}_{\mathbb{R}}(r_1, \dots, r_{k-1}) = \text{span}_{\mathbb{R}}(r_1^*, \dots, r_{k-1}^*).$$

So, if we define $[r'_1, \dots, r'_n] = R' \in M_n(\mathbb{Z})$ with the above procedure, we get a basis R' for $\mathcal{L}(B)$ with the desired properties. $\square$

So, the Micciancio-Goldwasser algorithm can be written in pseudo-code as follows, where all the vectors are treated as column vectors.

MG: Micciancio-Goldwasser Algorithm

Input : $B = [b_1, \dots, b_n] \in M_n(\mathbb{Z})$ basis for $\mathcal{L}(B)$,

$S = [s_1, \dots, s_n] \in M_n(\mathbb{Z})$ linearly independent vectors in $\mathcal{L}(B)$ s.t. (up to reordering)

$\|s_1\|_2 \leq \|s_2\|_2 \leq \dots \leq \|s_n\|_2$.

Output : $R' = [r'_1, \dots, r'_n] \in M_n(\mathbb{Z})$ basis for $\mathcal{L}(B)$ s.t.

$\|r'_k\|_2 \leq \max \left\{ \frac{\sqrt{k}}{2} \|s_k\|_2, \|s_k\|_2 \right\}, \text{ span}_{\mathbb{R}}(r'_1, \dots, r'_k) = \text{span}_{\mathbb{R}}(s_1, \dots, s_k), \|r'_k\|_2 \leq \|s_k^*\|_2$

for all $k = 1, \dots, n$.

```

1 : for  $i = 1$  to  $n$  do
2 :    $q_i = \text{Solve}(Bx = s_i)$ 
3 : endfor ;
4 :  $Q = [q_1, \dots, q_n]$ 
5 :  $T, U = \text{HermiteNormalForm}(Q) \### T = U \cdot Q, T \in M_n(\mathbb{Z})$  upper triangular ,  $U \in \text{GL}_n(\mathbb{Z})$ 
6 :  $R = [r_1, \dots, r_n] = B \cdot U^{-1}$ 
7 :  $R^* = [r_1^*, \dots, r_n^*] = \text{GramSchmidt}(R) \###$  for columns
8 :  $S^* = [s_1^*, \dots, s_n^*] = \text{GramSchmidt}(S) \###$  for columns
9 :  $r'_1 = r_1$ 
10 : for  $i = 2$  to  $n$  do
11 :   if  $r_i^* == s_i^*$  then
12 :      $r'_i = s_i$ 
13 :   else if  $r_i^* == -s_i^*$  then
14 :      $r'_i = s_i$ 
15 :   else
16 :      $v = \text{NearestPlane}(r_i - r_i^*, \mathcal{L}(r'_1, \dots, r'_{i-1}))$ 
17 :      $r'_i = r_i - v$ 
18 :   endif
19 : endfor
20 :  $R' = [r'_1, \dots, r'_n] \### R' \in M_n(\mathbb{Z})$ 
21 : return  $R'$ .
```

Let us have a look at an example of the Micciancio-Goldwasser Algorithm.

Example 3.4.3. Consider the 4-dimensional integer lattice $\mathcal{L} \subseteq \mathbb{Z}^4$ with basis matrix

$$B = [b_1, b_2, b_3, b_4] = \begin{pmatrix} 1346 & 260 & -3210 & 3958 \\ 594 & 116 & -1450 & 1815 \\ -2412 & -423 & 4623 & -4737 \\ 1336 & 268 & -3448 & 4472 \end{pmatrix}.$$

Suppose that we know that the linearly independent vectors

$$s_1 = (4, 0, 0, 0)^T, s_2 = (4, 2, 0, 0)^T, s_3 = (0, -6, 6, 0)^T, s_4 = (0, 0, 16, -32)^T$$

are in $\mathcal{L}$. If we apply the Micciancio-Goldwasser Algorithm to B and $[s_1, s_2, s_3, s_4]$ we get the basis

matrix

$$R' = [r'_1, r'_2, r'_3, r'_4] = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix},$$

where we can see that

$$\|r'_k\|_2 \leq \max \left\{ \frac{\sqrt{k}}{2} \|s_k\|_2, \|s_k\|_2 \right\}$$

for all $k = 1, 2, 3, 4$. In particular the norm of the basis vectors has reduced drastically.

In the future sections we will see the hard lattice problem on whose hardness rely most of the lattice-based cryptosystems, included the ones in this thesis: the the Learning With Errors (LWE) problem.

To introduce the LWE problem, we have to first talk about sampling distributions on lattices.

3.5 Discrete Gaussian Distribution and Sampling over lattices

The discrete Gaussian distribution is the extension of the continuous Gaussian distribution over $\mathbb{R}^n$ to the context of lattices. It was introduced in 2007 by Micciancio and Regev in [45].

Definition 3.5.1. A Gaussian function (over $\mathbb{R}$) is a continuous function of the form

$$f(x) = a \cdot e^{-\frac{(x-c)^2}{2\sigma^2}},$$

where $a, c, \sigma \in \mathbb{R}$ and $a > 0$.

The best known Gaussian function is the density function of the Gaussian probability distribution.

Definition 3.5.2. The Gaussian measure over $\mathbb{R}$ is the function

$$\rho_{\sigma,c}(x) = e^{-\frac{(x-c)^2}{\sigma^2}}.$$

Another possible realization can be made using the scaling factor $s = \sqrt{2\pi}\sigma$. Then, we can generalize the previous measure to a measure on $\mathbb{R}^n$:

$$\rho_{s,c}(x) = e^{-\frac{\pi\|x-c\|^2}{s^2}}.$$

Remark 3.5.3. If we integrate the previous measure over $\mathbb{R}^n$ we get that

$$\int_{x \in \mathbb{R}^n} \rho_{s,c}(x) dx = s^n,$$

so we can define the n -dimensional Gaussian probability distribution function (over $\mathbb{R}^n$) as

$$D_{s,c}(x) = \frac{\rho_{s,c}(x)}{s^n}.$$

Clearly, the expressions for $\rho_{s,c}(x)$ and $D_{s,c}(x)$ are still well-defined if x is a vector in a lattice. Because a lattice $\mathcal{L}$ is a discrete and countable set, the Gaussian measure on $\mathcal{L} \subseteq \mathbb{R}^n$ and its discretized density function are, respectively

$$\rho_{s,c}(\mathcal{L}) = \sum_{x \in \mathcal{L}} \rho_{s,c}(x) \text{ and } D_{s,c}(\mathcal{L}) = \frac{\rho_{s,c}(\mathcal{L})}{s^n}.$$

Then, we can give the following definition.

Definition 3.5.4 (Discrete Gaussian Distribution). The discrete Gaussian distribution over a lattice $\mathcal{L} \subseteq \mathbb{R}^n$, centered in $c \in \mathbb{R}^n$, with scaling factor $s > 0$, is defined, for all $x \in \mathcal{L}$ as

$$D_{\mathcal{L},s,c}(x) = \frac{D_{s,c}(x)}{D_{s,c}(\mathcal{L})} = \frac{\rho_{s,c}(x)}{\rho_{s,c}(\mathcal{L})}.$$

To denote the distribution centered in 0, we write $D_{\mathcal{L},s,0}$ simply as $D_{\mathcal{L},s}$.

The discrete Gaussian distribution is the main tool to insert randomness in the construction of lattice-based cryptosystems. Contrary to a uniform distribution, it does not have well-defined edges, but the larger the scaling factor s is the closer the distribution resembles a uniform one.

In the following we will see how to sample from the discrete Gaussian distribution over a lattice. In order to sample from those distributions one has to use Monte Carlo methods.

In their work [28], Gentry, Peikert and Vaikuntanathan give a nice overview of the tools and algorithms that are needed to do discrete Gaussian sampling on lattices.

To meaningfully talk about sampling on lattices we first need the definition of smoothing parameter, given by Micciancio and Regev in [45].

Definition 3.5.5 (Smoothing parameter). For a n -dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and a positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\mathcal{L})$ is the smallest s such that

$$\rho_{\frac{1}{s},0}(\mathcal{L}^* \setminus \{0\}) \leq \epsilon,$$

where $\rho_{\frac{1}{s},0}(\mathcal{L}^* \setminus \{0\}) = \sum_{x \in \mathcal{L}^* \setminus \{0\}} \rho_{\frac{1}{s},0}(x)$.

The function $s \mapsto \rho_{\frac{1}{s},0}(\mathcal{L}^* \setminus \{0\})$ is a continuous and strictly decreasing function such that

$$\lim_{s \rightarrow 0} \rho_{\frac{1}{s},0}(\mathcal{L}^* \setminus \{0\}) = +\infty \text{ and } \lim_{s \rightarrow +\infty} \rho_{\frac{1}{s},0}(\mathcal{L}^* \setminus \{0\}) = 0.$$

This means that the smoothing parameter $\eta_\epsilon(\mathcal{L})$ is well-defined for every $\epsilon > 0$, and $\epsilon \mapsto \eta_\epsilon(\mathcal{L})$ is the inverse function of $s \mapsto \rho_{\frac{1}{s},0}(\mathcal{L}^* \setminus \{0\})$. This means that also $\epsilon \mapsto \eta_\epsilon(\mathcal{L})$ is a continuous decreasing function of ϵ .

The name smoothing parameter is given because if we start from a uniformly random lattice point in $\mathcal{L}$ and perturb it by a Gaussian of radius $\eta_\epsilon(\mathcal{L})$, then the resulting distribution is $\frac{\epsilon}{2}$ close to being uniform on the entire space [45, Lemma 4.1].

We are now ready to investigate Gaussian sampling on lattices. We start with sampling on the most basic lattice possible, $\mathbb{Z}$, with the algorithm **SampleZ**.

SampleZ

Input: $s > 0$, $c \in \mathbb{R}$, $t(n) \geq \omega(\log(n))$ a fixed function and $n \in \mathbb{Z}_{\geq 0}$

Output: an integer z

```

1 :  $z \leftarrow \mathbb{Z} \cap [c - s \cdot t(n), c + s \cdot t(n)]$ 
2 :  $x \leftarrow [0, 1] (\subseteq \mathbb{R})$  ###  $x$  is sampled uniformly in  $[0, 1]$ 
3 : if  $x < \rho_{s,c}(z)$  then
4 :   return  $z$ 
5 : else
6 :   goto line 1.
```

The lines 2. - 6. in the code for **SampleZ** are where the rejection sampling is performed: with probability $\rho_{s,c}(z) \in (0, 1]$ output z , otherwise repeat.

The correctness of **SampleZ** is given by the following tail inequality on the distribution $D_{\mathbb{Z},s,c}$, which can be found in [27, Lemma 4.2].

Lemma 3.5.6. *For every $\epsilon > 0$, every $s \geq \eta_\epsilon(\mathbb{Z})$ and every $t > 0$*

$$\mathbb{P} \left[|z - c| \geq t \cdot s \mid z \stackrel{D_{\mathbb{Z},s,c}}{\longleftarrow} \mathbb{Z} \right] \leq 2e^{-\pi t^2} \cdot \frac{1 + \epsilon}{1 - \epsilon}.$$

In particular, for $\epsilon \in (0, \frac{1}{2})$ and $t \geq \omega(\sqrt{\log n})$, the probability that $|z - c| \geq t \cdot s$ is negligible.

The following theorem ([27, Lemma 4.3]) guarantees that **SampleZ** terminates and that the distribution on $\mathbb{Z}$ it samples from is statistically close (thus computationally indistinguishable) to the distribution $D_{\mathbb{Z},s,c}$.

Theorem 3.5.7 (SampleZ). *For every $0 < \epsilon < e^{-\pi}$, every $s \geq \eta_\epsilon(\mathbb{Z})$ and $c \in \mathbb{R}$, and every function $t(n) = \omega(\log n)$, **SampleZ** terminates within $t(n) \cdot \omega(\log n)$ iterations with overwhelming probability, and its output distribution is statistically close, thus computationally indistinguishable, to $D_{\mathbb{Z},s,c}$.*

We are now able to describe an algorithm, **SampleD**, that allows us to sample from a discrete Gaussian distribution $D_{\mathcal{L},s,c}$ over any lattice $\mathcal{L} \subseteq \mathbb{R}^n$.

This algorithm is a randomization of Babai's Nearest Plane Algorithm 3.4.1. It chooses a plane at random by sampling from an appropriate discrete Gaussian over $\mathbb{Z}$.

SampleD

Input: $B = [v_1, \dots, v_n]$ a basis for a lattice $\mathcal{L} \subseteq \mathbb{R}^n$, $s > 0$ and $c \in \mathbb{R}^n$
Output: a vector $v \in \mathcal{L}$

```

1 :  $\tilde{B} = [v_1^*, \dots, v_n^*] = \mathbf{GramSchmidt}(B)$ 
2 :  $w_n = 0$ 
3 :  $c_n = c$ 
4 : for  $i = n$  down to 1 do
5 :    $c'_i = \frac{\langle c_i, v_i^* \rangle}{\|v_i^*\|_2^2} \quad \#\#\# \quad c'_i \in \mathbb{R}$ 
6 :    $s'_i = \frac{s}{\|v_i^*\|_2} \quad \#\#\# \quad s'_i > 0$ 
7 :    $z_i \leftarrow \mathbf{SampleZ}(s'_i, c'_i)$ 
8 :    $c_{i-1} = c_i - z_i v_i$ 
9 :    $w_{i-1} = w_i + z_i v_i$ 
10 : endfor
11 :  $v = w_0$ 
12 : return  $v$ 
```

The following result ([27, Theorem 4.1]) guarantees that, under some condition on s , the **SampleD** algorithm is correct.

Theorem 3.5.8 (SampleD). ***SampleD** is a probabilistic polynomial-time algorithm that, given a basis $B = [v_1, \dots, v_n]$ of a n -dimensional lattice $\mathcal{L}$, a parameter $s \geq \max_{i=1, \dots, n} \|v_i^*\|_2 \cdot \omega(\sqrt{\log n})$, and a center $c \in \mathbb{R}^n$, outputs a sample from a distribution that is statistically close to $D_{\mathcal{L},s,c}$, where $\tilde{B} = [v_1^*, \dots, v_n^*]$ is the Gram-Schmidt orthogonalization of B .*

We are now ready to describe the LWE problem.

3.6 Learning With Errors problem

Other than SVP, CVP and other computational problems over lattices, the security of lattice-based most cryptosystems relies on the hardness of the Learning With Errors (LWE) problem. The key idea behind LWE relies in representing secret information through equations with errors: the secrets are hidden behind some noise. Regev introduced this problem in 2005 in [53], and he proved that LWE is as hard to solve as some worst-case problems on lattices. In this way the hardness of the cryptographic constructions based on LWE is based on the hardness of those worst-case lattice problems.

The origins of the LWE problem can be traced back to Ajtai and Dwork seminal work [2], where they describe the first public-key cryptosystem whose hardness is based on some worst-case problems on lattices. Then, some simplifications and improvements were made by Goldreich, Goldwasser and Halevi in [32] and Regev in [52]. Although the LWE problem does not appear in any of the mentioned papers, in Regev's work [52] the explanation of the construction done by Ajtai and Dwork shows implicitly a result on the hardness of LWE.

To talk about LWE we need to define the LWE distribution.

Definition 3.6.1 (LWE distribution). Let $n \geq 1$ and $q \geq 2$ (the modulus) be integers. We fix an error distribution χ over $\mathbb{Z}/q\mathbb{Z}$ and a vector $s \in (\mathbb{Z}/q\mathbb{Z})^n$. The LWE distribution $A_{s,\chi}$ over $(\mathbb{Z}/q\mathbb{Z})^n \times \mathbb{Z}/q\mathbb{Z}$ is obtained performing the following steps:

- (1) Choose a vector $a \leftarrow (\mathbb{Z}/q\mathbb{Z})^n$ uniformly at random.
- (2) Choose an error term $e \leftarrow^{\chi} \mathbb{Z}/q\mathbb{Z}$ in $\mathbb{Z}/q\mathbb{Z}$ according to the distribution χ .
- (3) Compute $b = \langle a, s \rangle + e \bmod q$.
- (4) Output the pair $(a, b) \in (\mathbb{Z}/q\mathbb{Z})^n \times \mathbb{Z}/q\mathbb{Z}$.

Remark 3.6.2. Concerning parameter selection, in all applications the error distribution χ is a Gaussian distribution rounded to the nearest integer reduced modulo q , centered in 0 with standard deviation αq , where α is the error parameter and in general is of the form $\frac{1}{p(n)}$, where $p(n)$ is a polynomial in n . This type of distribution fits well the LWE construction, because when we sample the error terms we obtain small values. The modulus q is often polynomial in n , for efficiency reasons. This distribution is denoted with $\overline{\Psi}_{\alpha}$.

Definition 3.6.3 (LWE problem). Given a modulus $q \in \mathbb{Z}_{\geq 1}$ and an error distribution χ over $\mathbb{Z}/q\mathbb{Z}$, the search version of the LWE problem (or, simply the LWE problem) asks to find, with high probability, the secret vector $s \in (\mathbb{Z}/q\mathbb{Z})^n$, given an arbitrary number of samples from the LWE distribution $A_{\chi,s}$.

There is also a decision version of the LWE problem.

Definition 3.6.4 (dLWE problem). Given a modulus $q \in \mathbb{Z}_{\geq 1}$ and an error distribution χ on $\mathbb{Z}/q\mathbb{Z}$, the decision version of the LWE problem (dLWE), asks to distinguish between samples (a, b) generated from the distribution $A_{\chi,s}$ and samples generated from the uniform distribution over $(\mathbb{Z}/q\mathbb{Z})^n \times \mathbb{Z}/q\mathbb{Z}$. Equivalently, the dLWE problem asks to decide whether b is generated as $\langle a, s \rangle + e$ or as a random value.

Both the search version and the decision version of the LWE problem are hard to solve, and they are equivalent.

The LWE problem is broadly used in post-quantum cryptography because the best algorithms that allow to attack LWE are exponential-time and the quantum algorithms in use today do not add significant speed-ups. Moreover there are reductions to other known difficult lattice problems: for example, for a modulus q that is polynomial in the dimension of the lattice n , we can rely on the hardness of solving GapSVP (Definition 3.2.5) or SIVP (Definition 3.2.8) with an approximation factor that is polynomial in n , also with a quantum computer.

Before stating the best known reduction in this setting, which is precisely the one that links (via the use of quantum algorithms) the LWE problem with GapSVP and SIVP, we explain how to link (with classical algorithms) LWE with the SVP and CVP.

Remark 3.6.5. Let us consider a LWE instance of the form

$$A \cdot s + e = b,$$

where $A \in M_{m,n}(\mathbb{Z}/q\mathbb{Z})$ and $b \in (\mathbb{Z}/q\mathbb{Z})^m$ (i.e they are m samples from the LWE distribution), and $e \in (\mathbb{Z}/q\mathbb{Z})^m, s \in (\mathbb{Z}/q\mathbb{Z})^n$ are vectors whose coordinates have small values. In this hypothesis we can link LWE to CVP. In fact, if we consider the m -dimensional lattice

$$\mathcal{L} = \Lambda_q^\perp(A) = \{v \in \mathbb{Z}^n \mid A \cdot v = 0 \bmod q\},$$

which is an instance of a q -ary lattice, the vector in $\mathcal{L}$ which is the closest to b , is, with high probability, the vector $A \cdot s$, which is at distance $\|e\|_2$. To show the connection between the LWE problem and SVP, we consider the lattice

$$\mathcal{L}' = \{v \in \mathbb{Z}^{n+m+1} \mid (A|I_m| - b) \cdot v = 0 \bmod q\}.$$

Then, the vector $(s^T, e^T, 1)^T$ is in $\mathcal{L}'$, since

$$(A|I_m| - b) \cdot \begin{pmatrix} s \\ e \\ 1 \end{pmatrix} = A \cdot s + e - b = b - b = 0 \bmod q.$$

Since both s and e have as coordinates small values, $(s^T, e^T, 1)^T$ is a shortest non-zero vector in $\mathcal{L}'$; so, if we find $(s^T, e^T, 1)^T$ we have both the secret vector s and the error vector e , so we have a solution to the LWE problem.

Now we can state the main result for LWE reductions: Regev showed in [53, Theorem 1.1] that GapSVP and SIVP are reducible (using quantum algorithms) to the LWE problem, if the LWE distribution χ is the one of Remark 3.6.2, i.e. $\bar{\Psi}_\alpha$. This is of strongly motivates the use of the LWE distribution $A_{s, \bar{\Psi}_\alpha}$ for lattice-based cryptosystems that are post-quantum.

Theorem 3.6.6. *Let $n, q \in \mathbb{Z}_{\geq 1}$ and $\alpha \in \mathbb{R}_{>0}$ such that $\alpha q > 2\sqrt{n}$. If there exists an efficient (classical) algorithm that solves the LWE problem with LWE distribution $A_{s, \bar{\Psi}_\alpha}$ (see Remark 3.6.2), then there exists an efficient quantum algorithm that approximates GapSVP and SIVP to within a factor in $\tilde{O}\left(\frac{n}{\alpha}\right)$ in the worst-case.*

This means that, if an adversary can use efficient classical algorithms to solve the LWE problem with LWE distribution $A_{s, \bar{\Psi}_\alpha}$, then he can use them and other efficient quantum and classical algorithms to solve with good approximation GapSVP and the SIVP. This means that if we assume that GapSVP or SIVP are hard to solve even in a quantum computer, then also the LWE problem with LWE distribution $A_{s, \bar{\Psi}_\alpha}$ is hard to solve.

The LWE problem we presented here is also called plain-LWE. There are many variants of the problem: the most famous are ring-LWE and module-LWE.

The main reason why these LWE variants have been introduced is to reduce the dimension of the keys in the cryptosystems that use the problem.

One can achieve that, for example, if we add structure to the LWE samples. Suppose the n is a power of 2 and that we take n samples $a_1, \dots, a_n \in \mathbb{Z}/q\mathbb{Z}$ in the following way: $a_1 = (a_1^{(1)}, \dots, a_n^{(1)}) \leftarrow \mathbb{Z}/q\mathbb{Z}$ is chosen uniformly at random, while $a_2, \dots, a_n$ have the following shape:

$$\begin{aligned} a_2 &= (a_2^{(1)}, \dots, a_n^{(1)}, -a_1^{(1)}) \\ &\dots \\ a_i &= (a_i^{(1)}, \dots, a_n^{(1)}, -a_1^{(1)}, \dots, -a_{i-1}^{(1)}) \\ &\dots \\ a_n &= (a_n^{(1)}, -a_1^{(1)}, \dots, -a_{n-1}^{(1)}). \end{aligned}$$

In order to represent these vectors we only need $\mathcal{O}(n)$ elements of $\mathbb{Z}/q\mathbb{Z}$, and this allows to obtain keys whose dimension is smaller to their general counterparts. This new construction replaces de facto the group $(\mathbb{Z}/q\mathbb{Z})^n$ with the quotient ring $(\mathbb{Z}/q\mathbb{Z})[x]/(x^n + 1)$, and for this reason it is called ring-LWE. The additional request that n should be a power of 2 guarantees the irreducibility of the polynomial $x^n + 1$ over the rational field.

We can now formally define this variant of the LWE problem, which was first introduced by Lyubashevsky, Peikert and Regev in [42].

Definition 3.6.7 (Ring-LWE). Let us fix a parameter $n \geq 1$ (usually a power of 2), a prime module $q \geq 2$ and we define the ring $R_q = (\mathbb{Z}/q\mathbb{Z})/(x^n + 1)$. We fix also a error probability distribution χ over R_q and a (class of a) polynomial $s \in R_q$, which we keep secret. The ring-LWE distribution $A_{s,\chi}^R$ over $R_q \times R_q$ is obtained in the following way:

- (1) We choose $a \leftarrow R_q$ uniformly at random.
- (2) Choose an error term $e \xleftarrow{\chi} R_q$ in R_q according to the distribution χ .
- (3) Compute in R_q the (class of the) polynomial $b = a \cdot s + e$.
- (4) Output the pair $(a, b) \in R_q \times R_q$

Definition 3.6.8 (Ring-LWE and d-ring-LWE problems). The search version of the ring-LWE problem (or, simply, ring-LWE) asks an adversary to get, with high probability, the secret polynomial $s \in R_q$, having the possibility to have an arbitrary number of samples from $A_{s,\chi}^R$. The decisional version of the ring-LWE problem (or, simply, d-ring-LWE) consists in distinguishing ring-LWE samples from samples coming from an uniform distribution over $R_q \times R_q$.

The main difference between LWE and ring-LWE is the fact that in the latter the operations are executed in a quotient of a polynomial ring, so the matrix product is substituted with the polynomial multiplication in which we identify x^n with -1 . For the parameters, n is chosen as a power of 2, while q is usually a prime such that $q \equiv 1 \pmod{2n}$. The error distribution χ generalizes the one for LWE: the coefficients of the polynomial e are chosen to be from identically-distributed and independent Gaussians.

In [42], they prove that ring-LWE is hard to solve, and they also prove a reduction from the search version to the decisional one.

The other significant variant, module-LWE, refines LWE, but it also generalizes ring-LWE. It was introduced by Langlois and Stehlé in [39]. The structure is the same as LWE, but we replace the ring $\mathbb{Z}/q\mathbb{Z}$ with the ring $R_q = (\mathbb{Z}/q\mathbb{Z})[x]/(x^n + 1)$.

Definition 3.6.9 (Module-LWE). Let $n, d \geq 1$ two parameters, $q \geq 2$ a prime module and let $R_q = (\mathbb{Z}/q\mathbb{Z})[x]/(x^n + 1)$. Let χ be an error probability distribution over R_q . We choose a (column) vector $s \in R_q^d$ of (classes of) polynomials and we keep it secret. The module-LWE distribution $A_{s,\chi}^M$ over $R_q^d \times R_q$ is obtained in the following way:

- (1) We choose $a \leftarrow R_q^d$ uniformly at random.
- (2) Choose an error term $e \leftarrow^{\chi} R_q$ in R_q according to the distribution χ .
- (3) Compute in R_q the polynomial $b = a^T \cdot s + e$, where $\cdot$ is the matrix product.
- (4) Output the pair $(a, b) \in R_q^d \times R_q$.

Definition 3.6.10 (Module-LWE and d-module-LWE problems). The search version of the module-LWE problem (or, simply module-LWE) asks an adversary to get, with high probability, the secret vector $s \in R_q^d$, having the possibility to have an arbitrary number of samples from $A_{s,\chi}^M$. The decision version of the module-LWE problem (or, simply, d-module-LWE) consists in distinguishing module-LWE samples from samples coming from a uniform distribution over $R_q^d \times R_q$.

Module-LWE is a generalization of ring-LWE, since ring-LWE is just module-LWE with $d = 1$.

The hardness of the most famous post-quantum digital signatures, CRYSTALS - Dilithium, is based on the hardness of module-LWE. In the next section we will see how a simplified version of CRYSTALS - Dilithium works.

3.7 Two Lattice-based Cryptographic Constructions: GGH and CRYSTALS - Dilithium

In the last section of this chapter we want to describe one of the state-of-the-art systems in lattice-based post-quantum cryptography: the digital signature CRYSTALS - Dilithium. Before that, we want to present a system in which the underlying lattice construction is immediate.

The cryptosystem we are talking about is called GGH, in honor of Goldreich, Goldwasser and Halevi, who first proposed it in 1997 in [33].

Suppose that Bob wants to send Alice a message.

First of all, Alice begins by choosing a set of linearly independent (column) vectors

$$v_1, v_2, \dots, v_n \in \mathbb{Z}^n$$

that are reasonably orthogonal to one another. One way to do this is to fix a parameter d and choose the (integer) coordinates of $v_1, v_2, \dots, v_n$ randomly between $-d$ and d . Alice can check that her choice of vectors is good by computing the Hadamard ratio of her basis and verifying that it is not too small. The vectors $v_1, \dots, v_n$ are Alice's private key. For convenience, as usual we let B be the $n \times n$ matrix whose columns are the vectors $v_1, \dots, v_n$ and we let $\mathcal{L}$ be the lattice generated by these vectors.

Alice next chooses an $n \times n$ matrix U with integer coefficients and such that $\det(U) = \pm 1$. One way to create U is as a product of a large number of randomly chosen elementary matrices. She then computes

$$W = UV.$$

The column vectors $w_1, \dots, w_n$ of W are a new basis for the lattice $\mathcal{L}$. They are Alice's public key.

When Bob wants to send a message to Alice, he selects a small vector $m = (m_1, \dots, m_n)^T$ as his plaintext, i.e. m might be a binary vector. Bob also chooses a small random perturbation vector r that acts as an ephemeral key. For example he might choose the coordinates of r randomly between $-\delta$ and δ , where δ is a fixed public parameter. He then computes the vector

$$e = Wm + r = \sum_{i=1}^n w_i m_i + r,$$

which is his ciphertext. Notice that e is not a lattice point, but it is close to the lattice point Wm , since r is small.

Decryption is straightforward: Alice uses Babai's Algorithm (Theorem 3.2.4) with the good basis $v_1, \dots, v_n$ to find a vector v in $\mathcal{L}$ that is close to e . Since she is using a good basis and r is small, the lattice vector she finds is Wm . She then solves the system $v = Wm$ to recover m .

Example 3.7.1. Let us look at the GGH cryptosystem with a 3-dimensional example. For Alice's private good basis we take

$$v_1 = (-97, 19, 19)^T, \quad v_2 = (-36, 80, 86)^T, \quad v_3 = (-184, -64, 78)^T.$$

The lattice $\mathcal{L}$ is spanned by v_1, v_2, v_3 and these vectors (according to the Hadamard ratio, which is circa 0,7462) are sufficiently orthogonal.

Alice multiplies her private basis (contained in the rows of the matrix V) by the matrix

$$\begin{pmatrix} 4327 & 3297 & 5464 \\ -155447 & -11770 & -19506 \\ 23454 & 17871 & 29617 \end{pmatrix},$$

which has determinant $\det(U) = -1$, to create her public basis

$$\begin{aligned} w_1 &= (-4179163, -1882253, 583183)^T, \\ w_2 &= (-3184353, -1434201, 444361)^T, \\ w_3 &= (-5277320, -2376852, 736426)^T. \end{aligned}$$

The vectors in the new basis are highly non-orthogonal (as highlighted in the Hadamard ratio, which is very close to zero). So this public basis is a very bad basis for the scope of Babai's Algorithm.

Bob decides to send Alice the plaintext $m = (86, -35, -32)^T$ using the random perturbation $r = (-4, -3, 2)^T$. The corresponding ciphertext is

$$e = \begin{pmatrix} -4179163 & -3184353 & -5277320 \\ -1882253 & -1434201 & -2376852 \\ 583183 & 444361 & 736426 \end{pmatrix} \begin{pmatrix} 86 \\ -35 \\ -32 \end{pmatrix} + \begin{pmatrix} -4 \\ -3 \\ 2 \end{pmatrix} = \begin{pmatrix} -79081427 \\ -35617462 \\ 11035473 \end{pmatrix}.$$

Alice uses Babai's Algorithm to decrypt. She first writes e as a linear combination of her private basis with real coefficients:

$$e \approx 81878.97v_1 - 292300.00v_2 + 443815.04v_3.$$

She rounds the coefficients to the nearest integer and computes a lattice vector

$$v = 81879v_1 - 292300v_2 + 443815v_3 = (-79081423, -35617459, 11035471)^T$$

that is close to e . Then, she recovers m by expressing v as a linear combination of the public basis and reading off the coefficients:

$$v = 86w_1 - 35w_2 - 32w_3.$$

Now, suppose that an adversary Eve tries to decrypt Bob's message, but she knows only the public basis w_1, w_2, w_3 . If she applies Babai's Algorithm using the public basis, she finds that

$$e \approx 75.76w_1 - 34.52w_2 - 24.18w_3.$$

Rounding, she obtains a lattice vector

$$v' = 75w_1 - 35w_2 - 24w_3 = (-79508353, -35809745, 11095049)^T$$

that is somewhat close to e . However, this lattice vector gives the incorrect plaintext $(76, -35, -24)^T$, not the correct one $m = (86, -35, -32)^T$. It is worth to describe and compare how well Babai's Algorithm did for the different basis. We find that

$$\|e - v\| \approx 5.3852 \quad \text{and} \quad \|e - v'\| \approx 472000.$$

We should note that the GGH cryptosystem is not secure in dimension 3, since even if we use numbers that are large enough to make an exhaustive search impractical, there are efficient algorithms to find good bases in low dimension (for example, the already described LLL Reduction Algorithm).

Remark 3.7.2. GGH is an example of a probabilistic cryptosystem, since a single plaintext leads to many different ciphertexts due to the choice of the random perturbation r . This leads to a potential danger if Bob sends the same message twice using different random perturbations, or sends different messages using the same random perturbation. Thus, in practice, the random perturbation r is determined by applying a hash function to the plaintext m .

We should advise the reader that the GGH cryptosystem has not been broken yet, but it has been rendered impractical by an attack done by Phong Nguyen in 1999 [50].

Now we are ready to give an overview of the basic concepts behind a simplified version of recently standardized lattice-based signature scheme CRYSTALS - Dilithium [24], [7]. CRYSTALS - Dilithium was standardized in 2024 in the NIST publication FIPS 204 as the Module-Lattice-Based Digital Signature Standard (ML-DSA) [49].

To give an insight on the functioning of the CRYSTALS - Dilithium signature scheme, we have to introduce some notation.

In Dilithium we will work with polynomials in the ring $R_q = (\mathbb{Z}/q\mathbb{Z})/(x^n+1)$, with $q = 2^{23} - 2^{13} + 1 = 8380417$ a prime integer, and $n = 256 = 2^8$.

Whenever $a \in \mathbb{Z}/q\mathbb{Z}$, we can choose a representative $\tilde{a}$ of it in the interval of integers $\left\{-\frac{q-1}{2}, \dots, \frac{q-1}{2}\right\}$. We define the absolute value of $a \in \mathbb{Z}_q$ as the absolute value of its representative $\tilde{a} \in \left\{-\frac{q-1}{2}, \dots, \frac{q-1}{2}\right\} \subseteq \mathbb{Z}$, i.e.

$$|a| = |\tilde{a}|.$$

Now, let $\alpha \in \mathbb{N}$ be a natural number such that α is a divisor of $q - 1$ and α is even. If $a \in \mathbb{Z}/q\mathbb{Z}$ we can take its representative in $\mathbb{Z}$ in the interval $\{0, \dots, q - 1\}$ and denote it by $\hat{a}$. Then, there exist two unique integers $\text{HighBits}(\hat{a}, \alpha), \text{LowBits}(\hat{a}, \alpha)$ (which are, respectively, the high order bits and the low order bits of $\hat{a}$ with respect to α) such that

$$\hat{a} = \alpha \cdot \text{HighBits}(\hat{a}, \alpha) + \text{LowBits}(\hat{a}, \alpha),$$

with $\text{LowBits}(\hat{a}, \alpha) \in \{-\frac{\alpha}{2} + 1, \dots, \frac{\alpha}{2}\}$.

For example, with $q = 8380417$ as before, $a = 7122341 \in \mathbb{Z}_q$ e $\alpha = \frac{q-1}{16} = 523776$, we have that

$$\widehat{7122341} = \alpha \cdot 14 - 210523,$$

where $\text{HighBits}(\hat{a}, \alpha) = 14$ and $\text{LowBits}(\hat{a}, \alpha) = -210523 \in \{-\frac{\alpha}{2} + 1, \dots, \frac{\alpha}{2}\}$.

Wherever there is no ambiguity, we will drop the hat from the representative of $a \in \mathbb{Z}/q\mathbb{Z}$ and simply write

$$a = \alpha \cdot \text{HighBits}(a, \alpha) + \text{LowBits}(a, \alpha).$$

The polynomial ring R_q is comprised of all the polynomials of degree $\leq n-1$ with coefficients in $\mathbb{Z}/q\mathbb{Z}$:

$$R_q = \left\{ f = \sum_{i=0}^{n-1} a_i x^i \mid a_i \in \mathbb{Z}/q\mathbb{Z} \ \forall i = 0, \dots, n-1 \right\}.$$

If we have $g = \sum_{j=0}^m b_j x^j \in (\mathbb{Z}/q\mathbb{Z})[x]$, we can map it in R_q and view it as the polynomial

$$\sum_{i=0}^{n-1} \left(\sum_{\substack{j=0 \\ j \equiv i \pmod n}}^m (-1)^{\frac{j-i}{n}} \cdot b_j \right) x^i.$$

In practice, if we want to see g in R_q , the previous reduction is equivalent to substituting x^n with -1 every time we encounter it in the writing of g .

An interesting and useful way to represent polynomials in R_q is via the vectors of their coefficients:

$$f = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} \in R_q \longleftrightarrow (a_0, \dots, a_{n-1}) \in (\mathbb{Z}/q\mathbb{Z})^n.$$

This also allows us to define the infinite norm of $f \in R_q$. If $f = \sum_{i=0}^{n-1} a_i x^i$ then

$$\|f\|_\infty = \|(a_0, \dots, a_{n-1})\|_\infty = \max_{i=0, \dots, n-1} |a_i|,$$

where the absolute values of the a_i s are the ones over $\mathbb{Z}/q\mathbb{Z}$ we saw before.

Finally, whenever we write $R_{q(\gamma)}$ for a $\gamma \in \mathbb{N}$, we intend the set of all the polynomials in R_q having infinite norm less or equal than γ :

$$R_{q(\gamma)} = \{f \in R_q \mid \|f\|_\infty \leq \gamma\}.$$

We are now ready to present a simplified version of CRYSTALS - Dilithium (as in [7]), which is the a basic approach to the complete scheme. By "simplified" we mean without the tools that allow the compression of the public key.

We recall that a digital signature scheme is a protocol between a signer S and a verifier V . The signer S wants to sign a message $m \in M$ (with M space of all the possible messages) with his secret key sk and send it to the verifier V , who has to be able to verify the signature of S with a public key in a way that reveals nothing about the secret key of S .

In this version of Dilithium, the creation of the couple (public key, secret key) is done by the signer S in the following way. S picks a random matrix A in the space of all the $k \times \ell$ matrices with entries in R_q ($M_{k,\ell}(R_q)$). Then S picks two random vectors s_1 and s_2 , respectively in $R_{q(\eta)}^\ell$ and in $R_{q(\eta)}^k$, with η a public security parameter. Now, he computes $t = As_1 + s_2$. Then the couple (public key, secret key) is

$$\text{pk} = (A, t), \text{ sk} = (s_1, s_2).$$

Key generation, S

- 1 : $A \leftarrow M_{k,\ell}(R_q)$
- 2 : $(s_1, s_2) \leftarrow R_{q(\eta)}^\ell$
- 3 : $t = As_1 + s_2$
- 4 : $\text{pk} = (A, t), \text{sk} = (s_1, s_2).$

After generating the public key and the secret key, the signer S has to sign the message $m \in M$ using his secret key sk . In order to do that, he needs to select randomly a vector of polynomials y in $R_{q(\gamma_1-1)}^\ell$, where $\gamma_1 = \frac{q-1}{16}$ is a security parameter. Then S computes the high order bits w_1 of Ay with respect to $2\gamma_2$, where $\gamma_2 = \frac{q-1}{32} = \frac{\gamma_1}{2}$ is another security parameter. At this point S needs a random function (which can be modeled on a hash function)

$$\mathcal{H} : \{0, 1\}^* \longrightarrow B_\tau$$

that can map arbitrary binary strings in

$$B_\tau = \{f \in R_q \mid f \text{ has exactly } \tau \text{ coefficients } \pm 1 \text{ and the others are } 0\} \subseteq R_q.$$

This function works in the following way. First of all a random binary string $b \in \{0, 1\}^*$ is mapped in $\tilde{b} \in \{0, 1\}^{256}$ via the hash function SHAKE-256 ($\tilde{b} = \text{SHAKE-256}(b)$). Then the output $\tilde{b}$ is again given, as a new input, to SHAKE-256, to produce $\tilde{\tilde{b}} = \text{SHAKE-256}(\tilde{b}) \in \{0, 1\}^{256}$. This $\tilde{\tilde{b}}$ contains the randomness needed in steps 3 and 4 by the following algorithm, which returns an element in B_τ (written, according to the usual notation, as a vector).

SampleInBall

- 1 : Initialize $c = (c_0, c_1, \dots, c_{255}) = (0, 0, \dots, 0)$
- 2 : **for** $i = 256 - \tau$ to 255 **do**
- 3 : $j \leftarrow \{0, 1, \dots, i\}$
- 4 : $s \leftarrow \{0, 1\}$
- 5 : $c_i = c_j$
- 6 : $c_j = (-1)^s$
- 7 : **endfor**
- 8 : **return** c .

Let us see how the randomness for j and s in SampleInBall works.

From $\tilde{\tilde{b}}$, SampleInBall initially extracts the first 64 bits. Then, the first τ bits (usually $\tau = 60$, for sure $\tau \leq 64$) of these 64 bits are interpreted as τ random sign bits

$$s_k \in \{0, 1\}, k = 0, \dots, \tau - 1.$$

The other $64 - \tau$ bits are discarded. The remaining $256 - 64 = 192$ bits of $\tilde{\tilde{b}}$ are divided into groups of 8 bits parameterized by the i in the for iteration of the algorithm. Every group of 8 bits then represents a number from 0 to 255. For every i in the for cycle, if the corresponding group of 8 bits represents a number in $\{0, \dots, i\}$ we select it as our j , otherwise we do some rejection sampling until we obtain a number in $\{0, \dots, i\}$ which we could select as our j . On the other hand, for every i in the for cycle, we select s as the s_{i-196} we calculated before.

Now S can create a challenge c for himself by computing $c = \mathcal{H}(m, w_1) \in B_\tau$ and generate the response $z = y + c \cdot s_1$. Note that c is a polynomial in $B_\tau \subseteq R_q$ and $s_1 \in R_{q(\eta)}^\ell \subseteq R_q^\ell$ is a vector of

polynomials ${}^t(s_{1,1}, \dots, s_{1,\ell})$, so

$$c \cdot s_1 = {}^t(c \cdot s_{1,1}, \dots, c \cdot s_{1,\ell}) \in R_q^\ell.$$

Before outputting c and z as his signature, the signer S has to perform two checks to guarantee the security and the correctness of it.

Let $\beta = \max\{\|c \cdot s_1\|_\infty, \|c \cdot s_2\|_\infty\}$.

If $\|z\|_\infty \geq \gamma_1 - \beta$ or $\|\text{LowBits}(Ay - c \cdot s_2, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$, then the signer S has to restart his signing procedure from the beginning (the first check is done for security, while the second for correctness of the scheme). If the checks are successful then S can output his signature $\sigma = (z, c)$, and send it to the verifier V .

Signing, S , sk , m

```

1 :  $y \leftarrow R_{q(\gamma_1-1)}^\ell$ 
2 :  $w_1 = \text{HighBits}(Ay, 2\gamma_2)$ ;
3 :  $c = \mathcal{H}(m, w_1)$ ;
4 :  $z = y + c \cdot s_1$ ;
5 : if  $\|z\|_\infty \geq \gamma_1 - \beta$  or  $\|\text{LowBits}(Ay - c \cdot s_2, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$  then
6 :   goto line 1
7 : endif
8 : return  $\sigma = (z, c)$ .
```

In order to verify the signature on the message m , the verifier V has to do the following steps. First of all, using the public key (A, t) and the signature of S $\sigma = (z, c)$, he can compute

$$w'_1 = \text{HighBits}(Az - c \cdot t, 2\gamma_2).$$

Then, if $\|z\|_\infty \geq \gamma_1 - \beta$ and c coincides with $\mathcal{H}(m, w'_1)$, the verifier V accepts the signature of S . Otherwise, he rejects it.

Verifying, V , pk , m , σ

```

1 :  $w'_1 = \text{HighBits}(Az - c \cdot t, 2\gamma_2)$ 
2 : if  $\|z\|_\infty \geq \gamma_1 - \beta$  and  $c == \mathcal{H}(m, w'_1)$  then
3 :   accept
4 : else
5 :   reject.
```

Let us prove that an honest signer is always accepted.

First of all, it is clear that an honest signer always passes the first verifier's check $\|z\|_\infty \geq \gamma_1 - \beta$. Then, we focus on the verifier's second check. In order to justify that $c == \mathcal{H}(m, w'_1)$, we remind that $c = \mathcal{H}(m, w_1)$. So, it will be enough to prove that $w_1 = w'_1$, i.e.

$$\text{HighBits}(Ay, 2\gamma_2) = \text{HighBits}(Az - c \cdot t, 2\gamma_2).$$

By our assumptions, we have that

$$\begin{aligned} Az - c \cdot t &= A(y + c \cdot s_1) - c \cdot (As_1 + s_2) = \\ &= Ay + c \cdot As_1 - c \cdot As_1 - c \cdot s_2 = \end{aligned}$$

$$= Ay - c \cdot s_2.$$

So, we are left to demonstrate that

$$\text{HighBits}(Ay, 2\gamma_2) = \text{HighBits}(Ay - c \cdot s_2, 2\gamma_2).$$

Since the signer S controls that

$$\|\text{LowBits}(Ay - c \cdot s_2, 2\gamma_2)\|_\infty < \gamma_2 - \beta$$

before returning the signature and since β is defined as the maximum between $\|c \cdot s_1\|_\infty$ and $\|c \cdot s_2\|_\infty$, we have that

$$\|c \cdot s_2\|_\infty \leq \beta < \gamma_2 - \|\text{LowBits}(Ay - c \cdot s_2, 2\gamma_2)\|_\infty \leq \gamma_2,$$

so $\text{HighBits}(c \cdot s_2, 2\gamma_2) = 0$. From the latter we deduce our claim, i.e.

$$\text{HighBits}(Ay, 2\gamma_2) = \text{HighBits}(Ay - c \cdot s_2, 2\gamma_2).$$

We now see how CRYSTALS - Dilithium is related to lattice problems. The matrix chosen by the signer S is made of $k\ell$ polynomials:

$$A = \begin{pmatrix} f_{1,1}(x) & f_{1,2}(x) & \cdots & f_{1,\ell}(x) \\ f_{2,1}(x) & f_{2,2}(x) & \cdots & f_{2,\ell}(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_{k,1}(x) & f_{k,2}(x) & \cdots & f_{k,\ell}(x) \end{pmatrix} \in M_{k,\ell}(R_q).$$

A polynomial $f_{i,j}(x) \in R_q$ can be visualized as a vector whose entries correspond to its coefficients:

$$f_{i,j} = (f_0^{(i,j)}, f_1^{(i,j)}, \dots, f_{n-1}^{(i,j)}) \in (\mathbb{Z}/q\mathbb{Z})^n.$$

We can then expand A to a block matrix $A^{\text{exp}} \in M_{kn,\ell n}(\mathbb{Z}/q\mathbb{Z})$: every block $F_{i,j}$ is square with order n and has as columns $f_{i,j}$ and its cyclic rotations, also applying the identification $x^n = -1$. Then, every block $F_{i,j} \in M_n(\mathbb{Z}/q\mathbb{Z})$ and looks like this:

$$F_{i,j} = \begin{pmatrix} f_0^{(i,j)} & -f_{n-1}^{(i,j)} & \cdots & -f_1^{(i,j)} \\ f_1^{(i,j)} & f_0^{(i,j)} & \cdots & -f_2^{(i,j)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n-1}^{(i,j)} & f_{n-2}^{(i,j)} & \cdots & f_0^{(i,j)} \end{pmatrix} \in M_n(\mathbb{Z}/q\mathbb{Z}).$$

So

$$A^{\text{exp}} = \begin{pmatrix} F_{1,1} & F_{1,2} & \cdots & F_{1,\ell} \\ F_{2,1} & F_{2,2} & \cdots & F_{2,\ell} \\ \vdots & \vdots & \ddots & \vdots \\ F_{k,1} & F_{k,2} & \cdots & F_{k,\ell} \end{pmatrix} \in M_{kn,\ell n}(\mathbb{Z}/q\mathbb{Z}).$$

If we repeat the same process with s_1, s_2 and t , we get $s_1^{\text{exp}} \in R_q^{\ell n}, s_2^{\text{exp}} \in R_q^{\ell n}$ and $t^{\text{exp}} \in R_q^{\ell n}$. Then the key generation process of Dilithium $t = As_1 + s_2$ can be interpreted as the matrix product and sum of matrices with entries in $\mathbb{Z}/q\mathbb{Z}$:

$$t^{\text{exp}} = A^{\text{exp}} \cdot s_1^{\text{exp}} + s_2^{\text{exp}}.$$

Considering the q -ary lattice

$$\mathcal{L} = \{A^{\text{exp}} \cdot z \mid z \in \mathbb{Z}^{\ell n}\} \subseteq \mathbb{Z}^{kn},$$

and reasoning on the fact that the entries of the vector s_2^{exp} are small, the LWE problem $t^{\text{exp}} = A^{\text{exp}} \cdot s_1^{\text{exp}} + s_2^{\text{exp}}$ on which the hardness of CRYSTALS - Dilithium is based can be linked to an approximate version of the CVP: t^{exp} is close to the lattice point $A^{\text{exp}} \cdot s_1^{\text{exp}}$.

Chapter 4

Elliptic Curves and Isogenies

In this chapter we will talk about another very interesting and profound mathematical area which has great importance for post-quantum cryptography, with one cryptosystem, SQIsign, whose variant has reached round three of the 2022 additional NIST call for digital signatures. In this chapter we will follow the expositions of Husemöller [35], Sconza [57] and Silverman [61, 62].

4.1 Definition and Properties

In the following, K will be a field (non necessarily algebraically closed) and $\overline{K}$ will be its algebraic closure.

Definition 4.1.1 (Elliptic curve). An elliptic curve is a pair (E, O_E) where E is a non-singular projective curve of genus 1 and $O_E \in E$ is a fixed point, called base point. If K is a field, we indicate with $E(K)$ the set of K -rational points of E , which is the set of points in E which have coordinates in K . We say that the elliptic curve E is defined over K if E is defined by a polynomial with coefficients in K : in this case we write E/K .

Definition 4.1.2 (Weierstrass equation). A Weierstrass equation for a curve in $\mathbb{P}^2$ is an equation of the form

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3,$$

with $a_1, \dots, a_6 \in \overline{K}$. We indicate with $K(E)$ the function field of E over K .

The next theorem, whose proof can be found in [61, III.3.1], shows that every elliptic curve can be written as a planar cubic and, conversely, every non-singular Weierstrass curve is an elliptic curve with base point $[0 : 1 : 0]$, if we consider it in the projective plane $\mathbb{P}^2$.

Theorem 4.1.3. *Let E be an elliptic curve defined over a field K . There exist functions $x, y \in K(E)$ such that the map $\phi : E \rightarrow \mathbb{P}^2$, $\phi = [x : y : 1]$ gives an isomorphism of E/K into a curve given by a Weierstrass equation*

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6 \tag{4.1}$$

with $a_1, \dots, a_6 \in K$ and satisfying $\phi(O_E) = [0 : 1 : 0]$. Conversely, every non-singular cubic curve given by a Weierstrass equation as in (4.1) is an elliptic curve defined over K with base point $[0 : 1 : 0]$. Any two Weierstrass equations for E as in (4.1) are related by an affine change of variables.

Thanks to the previous theorem we generally write the Weierstrass equation for an elliptic curve using the non-homogeneous coordinates $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$, i.e.

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

always remembering that there is the extra point $[0 : 1 : 0]$, which is also called point at infinity.

Then, if we use a Weierstrass equation for an elliptic curve E , we have that E is defined over K if $a_1, \dots, a_6 \in K$.

Definition 4.1.4 (Discriminant). Let E be an elliptic curve given by a Weierstrass equation. We define the discriminant of the Weierstrass equation as

$$\Delta(E) = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6,$$

with

$$b_2 = a_1^2 + 4a_2, \quad b_4 = 2a_4 + a_1a_3, \quad \text{and} \quad b_6 = a_3^2 + 4a_6.$$

Remark 4.1.5. Notice that, if we work over a field $\overline{K}$ of characteristic different from 2, 3 (and it is generally the case of cryptography), we can write the Weierstrass equation in the simpler way

$$E : y^2 = x^3 + Ax + B, \tag{4.2}$$

for some $A, B \in \overline{K}$. In this case the discriminant is

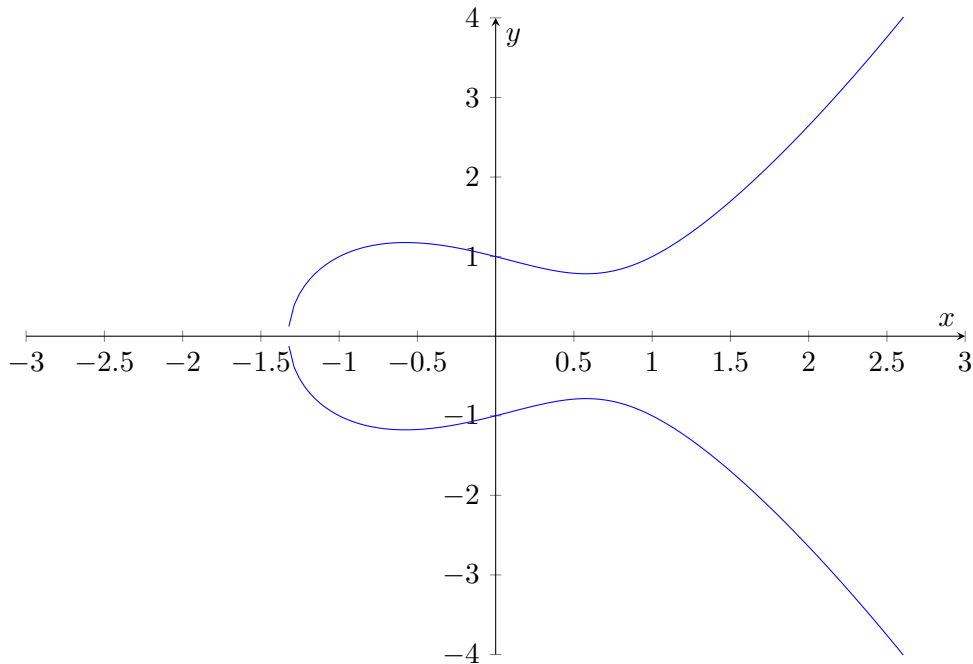
$$\Delta(E) = -(4A^3 + 27B^2),$$

and it is exactly the discriminant of the cubic polynomial $x^3 + Ax + B$.

The discriminant is the invariant of the Weierstrass equation that tells us if it is non-singular. Therefore, thanks to Theorem 4.1.3, it is the invariant that tells us if a curve in Weierstrass equation is an elliptic curve or not.

Theorem 4.1.6. [61, III.1.4a] *Let E be a curve given by a Weierstrass equation. It is non-singular (and therefore an elliptic curve) if and only if $\Delta(E) \neq 0$.*

Example 4.1.7. In this example we can see the elliptic curve over $\mathbb{R}$ with Weierstrass equation $y^2 = x^3 - x + 1$.



The most important invariant of an elliptic curve is the j -invariant.

Definition 4.1.8 (j -invariant). Let E be an elliptic curve given by a Weierstrass equation. The j -invariant of E is defined as

$$j(E) = \frac{(b_2^2 - 24b_4)^3}{\Delta(E)},$$

with $\Delta(E)$ the discriminant of E and b_2, b_4 as in Remark 4.1.5.

As we can see in the previous definition, the j -invariant is well-defined only for elliptic curves and not for all Weierstrass curves: we need the discriminant $\Delta(E)$ to be non-zero in order to invert it in the definition of $j(E)$.

The name j -invariant comes from the fact that if we apply an affine change of variables, the value of $j(E)$ does not change; it is thus independent of the Weierstrass equation chosen for E .

If we work with characteristic different from 2, 3 we can write $E : y^2 = x^3 + Ax + B$ and in this case the j -invariant can be computed as

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

The j -invariant $j(E)$ of an elliptic curve E also determines the class of elliptic curves which are isomorphic to E over $\overline{K}$.

Theorem 4.1.9. [61, III.1.4b,c] *Two elliptic curves E_1, E_2 are isomorphic over $\overline{K}$ if and only if $j(E_1) = j(E_2)$. Moreover, let $j_0 \in \overline{K}$. Then there exists an elliptic curve E defined over $K(j_0)$ (the field K algebraically extended adding j_0) such that $j(E) = j_0$.*

A consequence of the theorem above is that, if $j_0 \in K$ (with K a field), then there exists an elliptic curve E/K defined over K such that the j -invariant $j(E)$ is precisely j_0 .

The main difference between elliptic curves and other curves, that allow them to be used in many applications, is the fact that the set of points of an elliptic curve can be provided with a commutative group structure.

The most effective way in which we can describe the group operation is geometrically. Let E be an elliptic curve given by a Weierstrass equation and let P, Q be two points on E . We define the point $P + Q$ as follows. Draw the line r through P and Q (if $P = Q$ we consider as r the line tangent to E in P). Thanks to Bezout's Theorem, since the equation has degree 3, the line r intersects E in three points, say P, Q and R (not necessarily distinct). Let r' be the line through R and the point at infinity O_E . Then r' intersects E at R, O_E and a third point that we denote by $P + Q$.

Theorem 4.1.10. [61, III.2.2] *Let E be an elliptic curve given by a Weierstrass equation equipped with the operation $+$ described before. Then the following properties hold:*

- (1) *If a line intersects E at the (non necessarily distinct) points P, Q, R , then*

$$(P + Q) + R = O_E.$$

- (2) *$P + O_E = P$ for all $P \in E$.*

- (3) *$P + Q = Q + P$ for all $P, Q \in E$.*

- (4) *Let $P \in E$. There exists a point, denoted by $-P$, such that*

$$P + (-P) = O_E.$$

(5) Let $P, Q, R \in E$. Then

$$(P + Q) + R = P + (Q + R).$$

This theorem shows in particular that $(E, +)$ is an abelian group with identity element O_E . Another result in [61, III.2.2] shows that, if E is defined over K , then the set of K -rational points of E , denoted by $E(K)$ is a subgroup of $(E, +)$.

Since elliptic curves have an abelian group structure, we are now ready to talk about group homomorphisms between them, i.e isogenies.

Definition 4.1.11 (Isogeny). Let E_1, E_2 be two elliptic curves. An isogeny from E_1 to E_2 is a morphism of curves $\varphi : E_1 \rightarrow E_2$ such that $\varphi(O_{E_1}) = O_{E_2}$. Two elliptic curves E_1, E_2 are called isogenous if there exists an isogeny φ from E_1 to E_2 such that $\varphi(E_1) \neq \{O_{E_2}\}$, i.e φ is not the zero map.

If E_1 and E_2 are defined over K , then we can consider an isogeny between them defined over K : this means that we can consider isogenies of the form $\varphi : E_1 \rightarrow E_2$ written as $\varphi(x, y) = (R_1(x, y), R_2(x, y))$ with R_1, R_2 rational functions with coefficients in K .

We denote the set of isogenies from E_1 to E_2 by $\text{Hom}(E_1, E_2)$ and, since elliptic curves are abelian groups, this set is an abelian group with the addition defined pointwise. If $E_1 = E_2 = E$, we denote with $\text{End}(E)$ the group $\text{Hom}(E, E)$. When we consider isogenies defined over K we put the subscript $\text{Hom}_K(E_1, E_2)$ and $\text{End}_K(E)$.

Example 4.1.12. An important family of isogenies is given by the multiplication-by- m isogenies for $m \in \mathbb{Z}_{\geq 1}$:

$$\begin{array}{ll} [m] : E \rightarrow E & [-m] : E \rightarrow E \\ P \mapsto \underbrace{P + \dots + P}_{m \text{ times}} & P \mapsto \underbrace{(-P) + \dots + (-P)}_{m \text{ times}} \end{array}$$

We indicate with $[0]$ the zero isogeny.

Since a morphism of curves is either constant or surjective, we have from the definition that an isogeny $\varphi : E_1 \rightarrow E_2$ satisfies either $\varphi(E_1) = O_{E_2}$ or $\varphi(E_1) = E_2$.

The following result assures us that isogenies respect the abelian group structure of elliptic curves, i.e. isogenies are also group homomorphisms.

Theorem 4.1.13. [61, III.4.8] Let $\varphi : E_1 \rightarrow E_2$ be an isogeny. In particular, φ is a group homomorphism.

We can now define what is the degree of an isogeny.

Definition 4.1.14 (Degree of an isogeny). Let $\varphi : E_1 \rightarrow E_2$ an isogeny defined over a field K and let $K(E_1), K(E_2)$ be the function fields of E_1 and E_2 . We obtain the injection of function fields

$$\begin{aligned} \varphi^* : K(E_2) &\hookrightarrow K(E_1) \\ f &\mapsto f \circ \varphi. \end{aligned}$$

We define the degree of the isogeny φ , denoted as $\deg(\varphi)$ as the degree of the field extension $K(E_1)/\varphi^*(K(E_2))$, i.e

$$\deg(\varphi) = [K(E_1) : \varphi^*(K(E_2))].$$

Moreover, we use the convention that $\deg([0]) = 0$.

We will see that if we work with fields K such that $\text{char}(K) \neq 2$ then there are equivalent and more computable definitions of degree of isogenies.

The following proposition specifies that isogenies over fields K with $\text{char}(K) \neq 2$ can be written in a nice way.

Proposition 4.1.15. *[57, Proposition 2.6] Let E_1, E_2 be two elliptic curves over K with $\text{char}(K) \neq 2$, and let $\varphi : E_1 \rightarrow E_2$ be an isogeny defined over K . Then, φ can be represented as an affine rational map of the form*

$$\varphi(x, y) = \left(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)}y \right),$$

where $u, v, s, t \in K[x]$ are univariate polynomials in x such that $\gcd(u, v) = \gcd(s, t) = 1$. Moreover, if there exists $\bar{x} \in \bar{K}$ such that $t(\bar{x}) = 0$, then $v(\bar{x}) = 0$.

From this proposition we get the operative version of the degree definition.

Theorem 4.1.16. *[57, Theorem 2.12] Let K be a field such that $\text{char}(K) \neq 2$. Let $\varphi : E_1 \rightarrow E_2$ an isogeny defined over K written in the form*

$$\varphi(x, y) = \left(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)}y \right),$$

with $u, v, s, t \in K[x]$ and $\gcd(u, v) = \gcd(s, t) = 1$. Then

$$\deg(\varphi) = \max\{\deg(u), \deg(v)\},$$

where $\deg(u)$ and $\deg(v)$ are the degrees as polynomials.

Definition 4.1.17 (Separable, inseparable, purely inseparable (isogeny)). Let $\varphi : E_1 \rightarrow E_2$ be an isogeny defined over a field K . Recall the injection of function fields of Definition 4.1.14. We say that φ is separable, inseparable or purely inseparable if the extension $K(E_1)/\varphi^*(K(E_2))$ has the corresponding property. We denote with $\deg_s(\varphi)$ the separable degree of the extension, while with $\deg_i(\varphi)$ we denote the inseparable degree.

Another important object for isogenies is their kernel.

Theorem 4.1.18. *[61, III.4.10] Let $\varphi : E_1 \rightarrow E_2$ be a non-zero isogeny. Then*

$$\deg_s(\varphi) = |\text{Ker}(\varphi)|.$$

In particular, if φ is separable, we have that $\deg(\varphi) = |\text{Ker}(\varphi)|$; instead, if φ is inseparable, we have that $\deg(\varphi) > |\text{Ker}(\varphi)|$.

A very important consequence of the previous theorem is that the kernel of a non-zero isogeny is always finite. The following result assures us that for every finite subgroup H of an elliptic curve E , there exists an isogeny starting from E with kernel H .

Theorem 4.1.19. *[61, III.4.12] Let E be an elliptic curve and let $H \subseteq E$ be a finite subgroup of E . Then, there exists a unique elliptic curve E' and a separable isogeny $\varphi : E \rightarrow E'$ such that $\text{Ker}(\varphi) = H$.*

Definition 4.1.20 (m -torsion subgroup, complex multiplication). Let E be an elliptic curve defined over K and let $m \in \mathbb{Z}_{\geq 1}$. The m -torsion subgroup of E is the set of points of order m , i.e. the kernel of the multiplication-by- m isogeny $[m]$:

$$E[m] = \{P \in E(\bar{K}) \mid [m]P = O\}.$$

This definition highlights that there is an injection

$$[\cdot] : \mathbb{Z} \hookrightarrow \text{End}(E).$$

In the case this map is not surjective, we say that E has complex multiplication.

Definition 4.1.21 (Cyclic (isogeny)). Let $\varphi : E_1 \rightarrow E_2$ be an isogeny. We say that φ is cyclic if $\text{Ker}(\varphi)$ is a cyclic group.

Remark 4.1.22. Thanks to Theorem 4.1.18, the kernel of a separable isogeny $\varphi : E_1 \rightarrow E_2$ of prime degree p has cardinality p , so is automatically cyclic. Moreover, a non-separable isogeny ψ of prime degree p has trivial kernel, since

$$p = \deg(\psi) > |\text{Ker}(\psi)| = \deg_s(\psi) \text{ and } \deg_s(\psi) | p,$$

which forces $|\text{Ker}(\psi)| = 1$.

The previous remark, together with [57, Proposition 2.21], give us the following theorem.

Theorem 4.1.23. *An isogeny of composite degree can always be decomposed into a sequence of cyclic isogenies of prime degree.*

If there is an isogeny $\varphi : E_1 \rightarrow E_2$, the next theorem shows that there always exists a dual isogeny $\hat{\varphi} : E_2 \rightarrow E_1$ with some nice properties.

Theorem 4.1.24. [61, III.6.1, III.6.2] *Let $\varphi : E_1 \rightarrow E_2$ be a non-zero isogeny of degree m . Then, there exists a unique isogeny $\hat{\varphi} : E_2 \rightarrow E_1$ with the following properties:*

- (1) *We have that $\hat{\varphi} \circ \varphi = [m]_{E_1}$ and $\varphi \circ \hat{\varphi} = [m]_{E_2}$.*
- (2) *Let $\psi : E_2 \rightarrow E_3$ be another non-zero isogeny. Then $\widehat{\psi \circ \varphi} = \hat{\varphi} \circ \hat{\psi}$.*
- (3) *Let $\gamma : E_1 \rightarrow E_2$ be another non-zero isogeny. Then $\widehat{\varphi + \gamma} = \hat{\varphi} + \hat{\gamma}$.*
- (4) *For all $m \in \mathbb{Z}$ we have that $\widehat{[m]} = [m]$ and $\deg([m]) = m^2$.*
- (5) $\deg(\hat{\varphi}) = \deg(\varphi)$.
- (6) $\hat{\hat{\varphi}} = \varphi$.

Definition 4.1.25 (Dual isogeny). Let $\varphi : E_1 \rightarrow E_2$ a non-zero isogeny. The dual isogeny to φ is the unique isogeny $\hat{\varphi} : E_2 \rightarrow E_1$ described in Theorem 4.1.24. We also set $[\hat{0}] = [0]$.

The next proposition tells us the structure of the m -torsion subgroups for various values of m .

Proposition 4.1.26. [57, Proposition 2.26] *Let E be an elliptic curve and let $m \in \mathbb{Z}$ with $m \neq 0$.*

- (1) *If either $\text{char}(K) = 0$ or $\text{char}(K) = p > 0$ with $p \nmid m$, then*

$$E[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}.$$

- (2) *If $\text{char}(K) = p > 0$, then one of the following is true:*

$$(2.1) \ E[p^e] = \{O_E\} \text{ for all } e \in \mathbb{Z}_{\geq 1}.$$

$$(2.2) \ E[p^e] \cong \mathbb{Z}/p^e\mathbb{Z} \text{ for all } e \in \mathbb{Z}_{\geq 1}.$$

- (3) *If $\text{char}(K) = p > 0$ and $p \mid m$, write $m = p^r n$ with $p \nmid n$. Then*

$$E[m] \cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \text{ or } E[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}.$$

4.2 The Endomorphism Ring

Now we characterize which rings can appear as the endomorphism ring $\text{End}(E)$ of an elliptic curve E .

Definition 4.2.1 (Order). Let $\mathcal{K}$ be a finitely generated $\mathbb{Q}$ -algebra (non necessarily commutative). An order $\mathcal{O}$ of $\mathcal{K}$ is a subring of $\mathcal{K}$ that is finitely generated as a $\mathbb{Z}$ -module and such that it satisfies $\mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q} = \mathcal{K}$.

An equivalent definition is obtained replacing the last condition with the request that $\mathcal{O}$ contains a $\mathbb{Q}$ -basis of $\mathcal{K}$. In fact, if $\mathcal{O}$ contains a $\mathbb{Q}$ -basis of $\mathcal{K}$ then $\mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q} = \mathcal{K}$. Vice versa, if $\mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q} = \mathcal{K}$, then

$$\text{rank}_{\mathbb{Z}}(\mathcal{O}) = \dim_{\mathbb{Q}}(\mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q}) = \dim_{\mathbb{Q}}(\mathcal{K}).$$

Suppose $\dim_{\mathbb{Q}}(\mathcal{K}) = n$. Then, there exist $\omega_1, \dots, \omega_n \in \mathcal{O}$ forming a $\mathbb{Z}$ -basis of $\mathcal{O}$. By contradiction, $\omega_1, \dots, \omega_n$ are linearly dependent over $\mathbb{Q}$. Then,

$$n = \text{rank}_{\mathbb{Z}}(\mathcal{O}) = \dim_{\mathbb{Q}}(\langle \omega_1, \dots, \omega_n \rangle_{\mathbb{Q}}) < n,$$

and this is absurd. So $\omega_1, \dots, \omega_n \in \mathcal{O}$ is a $\mathbb{Q}$ -basis of $\mathcal{K}$.

Definition 4.2.2 (Quaternion algebra). A quaternion algebra is an algebra of the form

$$\mathcal{K} = \mathbb{Q} + \alpha\mathbb{Q} + \beta\mathbb{Q} + \alpha\beta\mathbb{Q},$$

with $\alpha^2, \beta^2 \in \mathbb{Q}$, $\alpha^2 < 0, \beta^2 < 0$ and $\beta\alpha = -\alpha\beta$.

Now we can state a classification theorem, whose proof can be found in [61, Theorem 2.33].

Theorem 4.2.3. *Let R be a ring with $\text{char}(R) = 0$ with no zero-divisors. Suppose that R has rank at most 4 as a $\mathbb{Z}$ -module and has an anti-involution $\alpha \mapsto \hat{\alpha}$ such that*

$$\widehat{\alpha + \beta} = \hat{\alpha} + \hat{\beta}, \widehat{\alpha\beta} = \hat{\beta}\hat{\alpha}, \hat{\hat{\alpha}} = \alpha, \hat{a} = a \text{ for all } a \in \mathbb{Z}.$$

Moreover, suppose that for $\alpha \in Rm$ the product $\alpha\hat{\alpha}$ is a non-negative integer and it is 0 if and only if $\alpha = 0$. Then, R is one of the following types of rings:

- (1) $R \cong \mathbb{Z}$.
- (2) R is an order in an imaginary quadratic field.
- (3) R is an order in a quaternion algebra over $\mathbb{Q}$.

From [61], we get that, for an elliptic curve E , its endomorphism ring $\text{End}(E)$ satisfies all the hypothesis of the theorem above, with the anti-involution mapping an isogeny to the dual isogeny. So, we have the following immediate consequence.

Theorem 4.2.4. *Let E be an elliptic curve. Then its endomorphism ring is either $\mathbb{Z}$, an order in an imaginary quadratic field, or an order in a quaternion algebra.*

When one works in positive characteristic, there is an important map that shall be considered, the Frobenius map. So, assume now that we consider a field K with $\text{char}(K) = p > 0$ and let $q = p^r$. Let E/K be an elliptic curve defined over K written in Weierstrass form as

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

We indicate with $E^{(q)}$ the curve obtained from E from raising each coefficient of E to the q -th power, i.e.

$$E^{(q)} : y^2 + a_1^q xy + a_3^q y = x^3 + a_2^q x^2 + a_4^q x + a_6^q.$$

Thanks to Fermat's Little Theorem, one can prove (with simple computations) that $E^{(q)}$ is an elliptic curve too. The natural map between E and $E^{(q)}$ is called the Frobenius map.

Definition 4.2.5 (Frobenius map). Let E/K be an elliptic curve defined over the field K such that $\text{char}(K) = p > 0$. Let $q = p^r$ for some $r \in \mathbb{Z}_{\geq 1}$. The q -th power Frobenius morphism is a morphism of curves defined as follows

$$\begin{aligned} \phi_q^E : E &\rightarrow E^{(q)} \\ (x, y) &\mapsto (x^q, y^q). \end{aligned}$$

Using Fermat's Little Theorem one can see that ϕ_q^E is well-defined.

In the next result, if we work with a field K of positive characteristic $p > 0$ and an elliptic curve E/K , we see that the possible forms of $E(p)$ and $\text{End}(E)$ are closely related.

Theorem 4.2.6. [61, Chapter V, Theorem 3.1] *Let K be a field of positive characteristic $p > 0$ and let E/K be an elliptic curve defined over K . For each integer $r \in \mathbb{Z}_{\geq 1}$ we indicate with*

$$\phi_r : E \rightarrow E^{(p^r)} \text{ and } \hat{\phi}_r : E^{(p^r)} \rightarrow E$$

the r -th power Frobenius morphism and its dual (respectively).

(1) *The following are equivalent:*

- (1.1) $E[p^r] = 0$ for a certain $r \geq 1$.
- (1.2) $E[p^r] = 0$ for all $r \geq 1$.
- (1.3) $\hat{\phi}_r$ is inseparable for a certain $r \geq 1$.
- (1.4) $\hat{\phi}_r$ is inseparable for all $r \geq 1$.
- (1.5) The map $[p] : E \rightarrow E$ is inseparable and $j(E) \in \mathbb{F}_{p^2}$.
- (1.6) $\text{End}(E)$ is an order in a quaternion algebra over $\mathbb{Q}$.

(2) *If the equivalent conditions in (1) do not hold, then*

$$E[p^r] \cong \mathbb{Z}/p^r\mathbb{Z} \text{ for all } r \geq 1.$$

Moreover, if $j(E) \in \overline{\mathbb{F}_p}$, then $\text{End}(E)$ is an order in an imaginary quadratic field.

Now, we are ready to define the subsets of ordinary and supersingular elliptic curves.

Definition 4.2.7 (Supersingular elliptic curves). Let E be an elliptic curve over a field K of characteristic $p > 0$. We say that E is supersingular if it satisfies the equivalent properties given in (1) of Theorem 4.2.6. Otherwise, we say that E is ordinary.

Even though in cryptography one is usually interested in elliptic curves and isogenies over finite fields, we can generalize the previous definition also for elliptic curves E defined over a field K non necessarily of positive characteristic. We say that E is supersingular if $\text{End}(E)$ is an order in a quaternion algebra. Otherwise, if $\text{End}(E)$ is either isomorphic to $\mathbb{Z}$ or is an order in an imaginary quadratic field, then we call E ordinary.

If we consider E an elliptic curve defined over a field $\mathbb{F}_q$, $q = p^r$ (p prime), its j -invariant $j(E)$ is an element in $\mathbb{F}_q$. Then, in particular $j(E) \in \overline{\mathbb{F}_q}$. So, in this case, the endomorphism ring is always strictly bigger than $\mathbb{Z}$.

In the last part of this section we state a result that affirms that the property of being ordinary or supersingular is an isogeny invariant.

Theorem 4.2.8. [57, Theorem 2.45] *Let K be a field of positive characteristic $p > 0$. Let $\varphi : E_1 \rightarrow E_2$ be a non-zero isogeny between two curves defined over K . Then E_1 is supersingular (resp. ordinary) if and only if E_2 is supersingular (resp. ordinary).*

Now we see that the Frobenius endomorphism plays an important role in determining what is the imaginary quadratic field associated with the endomorphism ring of an ordinary elliptic curve.

In the following, given an elliptic curve E , we use the notation

$$\text{End}^0(E) = \text{End}(E) \otimes \mathbb{Q}.$$

Theorem 4.2.9. [57, Theorem 3.37] *Let $E/\mathbb{F}_q$ be an ordinary elliptic curve with $\text{char}(\mathbb{F}_q) > 2$. Let $\phi_q^E : E \rightarrow E$ be the q -th Frobenius endomorphism. Then*

$$\text{End}^0(E) = \mathbb{Q}(\phi_q^E)$$

is an imaginary quadratic field.

We note that from this theorem we get that $\text{End}^0(E) \cong \mathbb{Q}(\sqrt{D})$, where $D = a^2 - 4q < 0$ and $a = q + 1 - |E(\mathbb{F}_q)|$.

4.3 Elliptic Curves over Finite Fields

Let $q = p^r$ with p prime. Then $\mathbb{F}_q$ is a finite field of characteristic $p > 0$ and let E be an elliptic curve defined over $\mathbb{F}_q$. We can consider the Frobenius morphism $\phi_q^E : E \rightarrow E^{(q)}$ that maps $(x, y) \mapsto (x^q, y^q)$ and $O_E \mapsto O_{E^{(q)}}$.

The next result shows that, in our hypotheses, the q -th Frobenius morphism is in fact an endomorphism.

Proposition 4.3.1. [57, Proposition 3.1] *Let E be an elliptic curve defined over $\mathbb{F}_q$ and let $(x, y) \in E(\overline{\mathbb{F}_q})$. Then:*

- (1) $\phi_q^E(x, y) \in E(\overline{\mathbb{F}_q})$.
- (2) $(x, y) \in E(\mathbb{F}_q)$ if and only if $\phi_q^E(x, y) = (x, y)$, i.e. (x, y) is a fixed point for ϕ_q^E .

The previous result allows us to call ϕ_q^E the Frobenius endomorphism. In the following, if the elliptic curve E on which we are considering the Frobenius endomorphism is obvious from the context, we omit it from ϕ_q^E and use the notation ϕ_q .

The number of $\mathbb{F}_q$ -rational points of an elliptic curve defined over $\mathbb{F}_q$ is very important in applications. A bound on this number of points is due to Hasse, and its proof (which we omit here but can be found for example in [57, Theorem 3.11]) involves manipulations with the Frobenius endomorphism.

Theorem 4.3.2 (Hasse). *Let E be an elliptic curve defined over $\mathbb{F}_q$, with $q = p^r$ power of a prime p . Then*

$$||E(\mathbb{F}_q)| - q - 1| \leq 2\sqrt{q}.$$

We know, thanks to Theorem 4.2.6 that an endomorphism ring of an elliptic curve defined over a finite field $\mathbb{F}_q$ can be an order in an imaginary quadratic field (for E ordinary elliptic curve) or an order in a quaternion algebra (for E supersingular elliptic curve). We now study the second case.

We have a result that characterizes supersingular curves defined over $\mathbb{F}_q$ thanks to the number of $\mathbb{F}_q$ -rational points they have.

Proposition 4.3.3. [57, Proposition 3.16] *Let E be an elliptic curve defined over $\mathbb{F}_q$, with $q = p^r$ power of a prime p (so $\text{char}(\mathbb{F}_q) = p$). Let $a = q + 1 - |E(\mathbb{F}_q)|$. Then E is supersingular if and only if $a = 0 \pmod p$, which is true if and only if $|E(\mathbb{F}_q)| = 1 \pmod p$.*

An immediate corollary of this proposition and Hasse's Theorem is the following.

Corollary 4.3.4. *Let $p \geq 5$ a prime number and E an elliptic curve defined over $\mathbb{F}_p$. then E is supersingular if and only if $|E(\mathbb{F}_p)| = p + 1$.*

The last result of this section is a powerful result that allows us to decide whenever two elliptic curves E, E' defined over $\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$. This theorem is due to Sato and Tate (a proof can be found in [57, Theorem 3.44]).

Theorem 4.3.5 (Sato-Tate). *Let $\mathbb{F}_q$ be a finite field with $q = p^r$ power of a prime p . Then two elliptic curves E, E' defined over $\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if they have the same number of $\mathbb{F}_q$ -rational points, i.e. $|E(\mathbb{F}_q)| = |E'(\mathbb{F}_q)|$.*

4.4 Elliptic Curves over $\mathbb{C}$

In order to talk about elliptic curves over the complex field $\mathbb{C}$ we have to introduce the concept of complex lattice.

Definition 4.4.1 (Complex lattice). A complex lattice Λ is a discrete subgroup of $\mathbb{C}$ that contains a $\mathbb{R}$ -basis of $\mathbb{C}$. Explicitly,

$$\Lambda = \text{span}_{\mathbb{Z}}(\omega_1, \omega_2),$$

with $\omega_1, \omega_2 \in \mathbb{C}$ such that $\omega_1 \neq \lambda \omega_2$ for all $\lambda \in \mathbb{R}$. We call $[\omega_1, \omega_2]$ a basis for Λ

The notations and nomenclature of complex lattices clearly extends from the one of n -dimensional lattices $\mathcal{L} \subseteq \mathbb{R}^n$.

Definition 4.4.2 (Complex torus). Let $\Lambda \subseteq \mathbb{C}$ be a complex lattice. The quotient $\mathbb{C}/\Lambda$ is called a complex torus.

The additive structure of $\mathbb{C}$ naturally carries over to $\mathbb{C}/\Lambda$.

Definition 4.4.3 (Homothetic (complex lattices)). Two complex lattices $\Lambda, \Lambda' \subseteq \mathbb{C}$ are said to be homothetic if there exists a complex number $\alpha \in \mathbb{C} \setminus \{0\}$ such that $\Lambda = \alpha \Lambda'$.

We are only interested in complex tori up to homothety. To classify them, we introduce Einstein series and the Weierstrass $\wp$ -function.

Definition 4.4.4 (Einstein series). Let $\Lambda \subseteq \mathbb{C}$ be a complex lattice. The Einstein series of weight $2k$ for Λ is the series

$$G_{2k}(\Lambda) = \sum_{0 \neq \omega \in \Lambda} \frac{1}{\omega^{2k}}.$$

Usually one sets

$$g_2(\Lambda) = 60G_4(\Lambda) \text{ and } g_3(\Lambda) = 140G_6(\Lambda).$$

Definition 4.4.5 (Weierstrass $\wp$ -function). Let $\Lambda \subseteq \mathbb{C}$ be a complex lattice. The Weierstrass $\wp$ -function associated to Λ is the series

$$\wp(z, \Lambda) = \frac{1}{z} + \sum_{0 \neq \omega \in \Lambda} \left(\frac{1}{(z - \omega)^2} - \frac{1}{\omega^2} \right).$$

The main theorem that connects complex tori and elliptic curves over $\mathbb{C}$ is the following. A proof can be found in [57, Theorem 4.21].

Theorem 4.4.6. *Let $\Lambda = \text{span}_{\mathbb{Z}}(\omega_1, \omega_2) \subseteq \mathbb{C}$ be a complex lattice. The Weierstrass $\wp$ -function has the following properties:*

- (1) *The Laurent series of $\wp(z) = \wp(z, \Lambda)$ around $z = 0$ is*

$$\wp(z) = \frac{1}{z^2} + \sum_{k=1}^{+\infty} (2k+1)G_{2k+1}z^{2k}.$$

- (2) *$\wp(z)$ satisfies the differential equation*

$$\wp'(z)^2 = 4\wp(z)^3 - g_2\wp(z) - g_3.$$

- (3) *The curve $E : y^2 = 4x^3 - g_2x - g_3$ is an elliptic curve over $\mathbb{C}$. Moreover, the map*

$$\begin{aligned} \Phi : \mathbb{C}/\Lambda &\rightarrow E \\ 0 &\mapsto [0 : 1 : 0] \\ z &\mapsto [\wp(z) : \wp'(z) : 1] \end{aligned}$$

is an isomorphism of groups.

We indicate the elliptic curve E associated to a complex lattice $\Lambda \subseteq \mathbb{C}$ with E_Λ , i.e., with the notations of the previous theorem, $E_\Lambda = \Phi(\mathbb{C}/\Lambda)$.

Now we want to associate a unique complex lattice to each elliptic curve over $\mathbb{C}$.

Definition 4.4.7 (Modular j -invariant). The modular j -invariant is the function on complex lattices defined by

$$j(\Lambda) = 1728 \frac{g_2(\Lambda)^3}{g_2(\Lambda)^3 - 27g_3(\Lambda)^2}.$$

Notice that if we have an elliptic curve E_Λ over $\mathbb{C}$ (which has characteristic 0) in short Weierstrass form, then we have the correspondence between j -invariants and modular j -invariants:

$$j(E_\Lambda) = j(\Lambda).$$

The theorem that allows us to associate a unique complex lattice to each elliptic curve over $\mathbb{C}$ is called Uniformization Theorem (a proof can be found in [57, Theorem 4.27]).

Theorem 4.4.8 (Uniformization Theorem). *Let $A, B \in \mathbb{C}$ two complex numbers satisfying $4A^3 + 27B^2 \neq 0$. Then there exists a unique complex lattice Λ such that $g_2(\Lambda) = -4A$ and $g_3(\Lambda) = -4B$.*

Using the correspondence between elliptic curves and complex tori we now have a new way of looking at their group structure: the natural group law on a torus $\mathbb{C}/\Lambda$ matches the group law on the elliptic curve E_Λ .

Now, we can also link isogenies and holomorphic maps between complex tori.

Theorem 4.4.9. *[57, Corollary 4.29] Let $E_1/\mathbb{C}$ and $E_2/\mathbb{C}$ two elliptic curves defined over $\mathbb{C}$ with corresponding lattices Λ_1 and Λ_2 . Then E_1 is isomorphic over $\mathbb{C}$ to E_2 if and only if Λ_1 and Λ_2 are homothetic.*

Working on elliptic curves over $\mathbb{C}$, thus over complex lattices, is often easier. Moreover some results proven over $\mathbb{C}$ can be transferred to finite fields, which are the most relevant for cryptographic applications.

4.5 Orders in Imaginary Quadratic Fields and the Ideal Class Group

Recall that an order $\mathcal{O}$ of a finitely generated $\mathbb{Q}$ -algebra $\mathcal{K}$ is a subring of $\mathcal{K}$ that is a finitely generated $\mathbb{Z}$ -module such that $\mathcal{O} \otimes \mathbb{Q} = \mathcal{K}$. We said that we can replace the last property with the requirement that $\mathcal{O}$ contains a $\mathbb{Q}$ -basis of $\mathcal{K}$. We now focus on studying orders only in quadratic fields.

Lemma 4.5.1. *[57, Lemma 5.1] Let $\mathcal{K}$ be a finite extension of $\mathbb{Q}$ of degree n and let $M \subseteq \mathcal{K}$ be a finitely generated $\mathbb{Z}$ -module. Then M is a free $\mathbb{Z}$ -module and M has rank n if and only if M contains a $\mathbb{Q}$ -basis of $\mathcal{K}$.*

So an order in a quadratic field $\mathcal{K}$ is a free $\mathbb{Z}$ -module of rank 2. This allows us to interpret orders as complex lattices. Moreover, since the ring of integers $\mathcal{O}_\mathcal{K}$ is a free $\mathbb{Z}$ -module of rank 2, it is an order of $\mathcal{K}$.

Definition 4.5.2 (Discriminant). Let $\mathcal{K} = \mathbb{Q}(\sqrt{D})$ be a quadratic field with D a squarefree integer. The discriminant $d_\mathcal{K}$ of $\mathcal{K}$ is

$$d_\mathcal{K} = \begin{cases} D & \text{if } D \equiv 1 \pmod{4} \\ 4D & \text{if } D \equiv 2, 3 \pmod{4}. \end{cases}$$

Proposition 4.5.3. *[57, Proposition 5.4] Let $\mathcal{K} = \mathbb{Q}(\sqrt{D})$ be a quadratic field with D a squarefree integer and let $d_\mathcal{K}$ be its discriminant. Then the ring of integers $\mathcal{O}_\mathcal{K}$ can be written as*

$$\mathcal{O}_\mathcal{K} = \text{span}_\mathbb{Z}(1, \omega_\mathcal{K}) \text{ with } \omega_\mathcal{K} = \frac{d_\mathcal{K} + \sqrt{d_\mathcal{K}}}{2}.$$

There is a similar description for all orders in quadratic fields ([57, Proposition 5.5]).

Proposition 4.5.4. *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ of discriminant $d_\mathcal{K}$. Then $\mathcal{O} \subseteq \mathcal{O}_\mathcal{K}$ and $\mathcal{O}$ has finite index in $\mathcal{O}_\mathcal{K}$. Moreover, if we set $f = [\mathcal{O}_\mathcal{K} : \mathcal{O}]$, then*

$$\mathcal{O} = \mathbb{Z} + f\mathcal{O}_\mathcal{K} = \text{span}_\mathbb{Z}(1, f\omega_\mathcal{K}),$$

where $\omega_\mathcal{K}$ is the same as in Proposition 4.5.3.

Definition 4.5.5 (Conductor). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. The index $f = [\mathcal{O}_\mathcal{K} : \mathcal{O}]$ is called the conductor of $\mathcal{O}$.

Thanks to the proposition above, we have that $\mathcal{O}_\mathcal{K}$ is the maximal order with respect to inclusion.

Definition 4.5.6 (Discriminant). Let $\mathcal{O} = \text{span}_{\mathbb{Z}}(\alpha, \beta)$ be an order in a quadratic field $\mathcal{K}$ and let $m \mapsto m'$ be the non-trivial automorphism of $\mathcal{K}$. The discriminant of $\mathcal{O}$ is the number

$$\text{disc}(\mathcal{O}) = D = \left(\det \begin{pmatrix} \alpha & \beta \\ \alpha' & \beta' \end{pmatrix} \right)^2.$$

The discriminant of $\mathcal{O}$ is independent of the basis used and if we compute it using $\mathcal{O} = \text{span}_{\mathbb{Z}}(1, f\omega_{\mathcal{K}})$ we obtain $D = f^2 d_{\mathcal{K}}$. It follows that $\mathcal{K} = \mathbb{Q}(\sqrt{D})$. Moreover, the discriminant satisfies $D = 0, 1 \pmod{4}$. Notice also that the discriminant of $\mathcal{O}_{\mathcal{K}}$ is $d_{\mathcal{K}}$.

Proposition 4.5.7. [57, Proposition 5.8] *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. Then, it is uniquely determined by its discriminant. Conversely, let D be a non-square integer such that $D = 0, 1 \pmod{4}$. Then there is an order in a quadratic field whose discriminant is D .*

Now we investigate the ideals of an order $\mathcal{O}$, to define the ideal class group $\text{cl}(\mathcal{O})$ associated to $\mathcal{O}$.

Proposition 4.5.8. *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ and let $\mathfrak{a}$ be a non-zero ideal of $\mathcal{O}$. Then:*

- (1) $\mathfrak{a}$ is a free $\mathbb{Z}$ -module of rank 2.
- (2) The quotient $\mathcal{O}/\mathfrak{a}$ is finite.
- (3) Every non-zero prime ideal of $\mathcal{O}$ is maximal.
- (4) $\mathcal{O}$ is Noetherian.

Proof. See [57, Lemma 5.9, Proposition 5.10]. □

Definition 4.5.9 (Norm (of an ideal)). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ and let $\mathfrak{a}$ be a non-zero ideal of $\mathcal{O}$. We define the norm of $\mathfrak{a}$ as

$$N(\mathfrak{a}) = |\mathcal{O}/\mathfrak{a}| \in \mathbb{Z}_{\geq 0}.$$

We can also interpret $N(\mathfrak{a})$ as the ratio of the areas of fundamental parallelograms for $\mathfrak{a}$ and $\mathcal{O}$, viewed as lattices in the $\mathbb{Q}$ -vector space $\mathcal{K}$ [64, Definition 18.3].

If $\alpha \in \mathcal{O}$, then the norm of α can be two a priori different things: as a complex number $N(\alpha) = \alpha\bar{\alpha}$, but we can also consider the principal $\mathcal{O}$ -ideal generated by α and $N(\alpha\mathcal{O}) = |\mathcal{O}/\alpha\mathcal{O}|$. One can prove that these two definitions of norm agree [64, Lemma 18.4].

From a simple computation we get the next lemma.

Lemma 4.5.10. *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$, let $\alpha \in \mathcal{O}$ be non-zero and let $\mathfrak{a}$ be a non-zero $\mathcal{O}$ -ideal. Then*

$$N(\alpha\mathfrak{a}) = N(\alpha)N(\mathfrak{a}).$$

In general, it is not true that the ideal norm is multiplicative, but it is true if we consider two proper $\mathcal{O}$ -ideals.

Lemma 4.5.11. [57, Lemma 5.16] *Let $\mathcal{K}$ be a quadratic field and let $\mathcal{O}_1, \dots, \mathcal{O}_n$ be orders in $\mathcal{K}$. Then there is a prime integer p such that $p\mathcal{O}_i$ is a prime $\mathcal{O}_i$ -ideal for all $i = 1, \dots, n$.*

Let $\mathfrak{a}$ be a non-zero $\mathcal{O}$ -ideal, where $\mathcal{O}$ is an order in a quadratic field $\mathcal{K}$. We denote

$$\mathcal{O}(\mathfrak{a}) = \{\beta \in \mathcal{K} \mid \beta\mathfrak{a} \subseteq \mathfrak{a}\}.$$

If $\mathfrak{a}$ and $\mathfrak{b}$ are two $\mathcal{O}$ -ideals and they are homothetic as lattices, then $\mathcal{O}(\mathfrak{a}) = \mathcal{O}(\mathfrak{b})$.

Definition 4.5.12 (Proper ($\mathcal{O}$ -ideal)). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ and let $\mathfrak{a}$ be an $\mathcal{O}$ -ideal. We say that $\mathfrak{a}$ is proper if $\mathcal{O}(\mathfrak{a}) = \mathcal{O}$.

Notice that $\mathcal{O} \subseteq \mathcal{O}(\mathfrak{a})$, since $\mathfrak{a}$ is an ideal of $\mathcal{O}$, but equality may not occur. Notice also that $\mathcal{O}(\mathfrak{a})$ is an order in $\mathcal{K}$.

Principal ideals are always proper and all ideals of $\mathcal{O}_{\mathcal{K}}$ are proper. We now extend this terminology to fractional ideals.

Definition 4.5.13 (Fractional $\mathcal{O}$ -ideal, Proper fractional $\mathcal{O}$ -ideal). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. For any $\lambda \in \mathcal{K} \setminus \{0\}$ and for every $\mathcal{O}$ -ideal $\mathfrak{a}$, the $\mathcal{O}$ -module

$$\lambda\mathfrak{a} = \{\lambda\alpha \mid \alpha \in \mathfrak{a}\}$$

is called a fractional $\mathcal{O}$ -ideal. A fractional $\mathcal{O}$ -ideal $\mathfrak{b}$ is proper if

$$\mathcal{O} = \{\beta \in \mathcal{K} \mid \beta\mathfrak{b} \subseteq \mathfrak{b}\}.$$

We also use the notation $\mathcal{O}(\mathfrak{b})$ to denote $\{\beta \in \mathcal{K} \mid \beta\mathfrak{b} \subseteq \mathfrak{b}\}$.

Multiplication of fractional ideals $\mathfrak{b} = \lambda\mathfrak{a}$ and $\mathfrak{b}' = \lambda'\mathfrak{a}'$ is defined as

$$\mathfrak{b}\mathfrak{b}' = (\lambda\lambda')\mathfrak{a}\mathfrak{a}'$$

where $\mathfrak{a}\mathfrak{a}'$ is the product of the ideals $\mathfrak{a}$ and $\mathfrak{a}'$.

Without loss of generality, we can assume $\lambda = 1/\beta$ with $\beta \in \mathcal{O}$ (if $\lambda = \alpha/\beta$ we replace $\mathfrak{a}$ with $\alpha\mathfrak{a}$). Moreover, we can also assume $\lambda = 1/b$ with $b \in \mathbb{Z}_{\geq 1}$ (see [57, Remark 5.21 (ii)]).

Remark 4.5.14. Every $\mathcal{O}$ -ideal is a fractional $\mathcal{O}$ -ideal. Conversely, fractional ideals that lie in $\mathcal{O}$ are $\mathcal{O}$ -ideals. Notice that in particular also $\mathcal{O}$ is a fractional $\mathcal{O}$ -ideal, which acts as the multiplicative identity with respect to multiplication of fractional $\mathcal{O}$ -ideals.

We now extend the concept of norm to fractional ideals.

Definition 4.5.15. Let $\mathfrak{b} = \frac{1}{b}\mathfrak{a}$ be a non-zero fractional $\mathcal{O}$ -ideal, with $b \in \mathbb{Z}_{\geq 1}$ and $\mathfrak{a}$ a non-zero $\mathcal{O}$ -ideal. The norm of $\mathfrak{b}$ is

$$N(\mathfrak{b}) = \frac{N(\mathfrak{a})}{N(b)} = \frac{N(\mathfrak{a})}{b^2} \in \mathbb{Q}_{>0}.$$

Again, thanks to the previous results, $N(\mathfrak{b})$ does not depend on the choices of b and $\mathfrak{a}$.

Now, since we have fractional ideals, we can introduce invertible ideals.

Definition 4.5.16 (Invertible fractional ideal). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. A fractional $\mathcal{O}$ -ideal $\mathfrak{b}$ is invertible if there exists another fractional $\mathcal{O}$ -ideal $\mathfrak{c}$ such that $\mathfrak{b}\mathfrak{c} = \mathcal{O}$. We indicate such a $\mathfrak{c}$ with $\mathfrak{b}^{-1}$, since one can prove that the inverse is unique.

In general a fractional $\mathcal{O}$ -ideal is not invertible, but principal fractional ideals, i.e. ideals of the form $\alpha\mathcal{O}$ with $\alpha \in \mathcal{K} \setminus \{0\}$ are invertible.

Definition 4.5.17 (Ideal group). The set of invertible fractional $\mathcal{O}$ -ideals form a group under multiplication, which is called the ideal group of $\mathcal{O}$ and we denote it with $I(\mathcal{O})$.

Lemma 4.5.18. [57, Lemma 5.24] *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ and let $\mathfrak{b} = \lambda\mathfrak{a}$ be a fractional $\mathcal{O}$ -ideal. Then $\mathfrak{a}$ is proper if and only if $\mathfrak{b}$ is proper, and $\mathfrak{a}$ is invertible if and only if $\mathfrak{b}$ is invertible.*

The next proposition, whose proof can be found in [57, Theorem 5.25], says that the invertible $\mathcal{O}$ -ideals are exactly the proper $\mathcal{O}$ -ideals.

Proposition 4.5.19. *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ and let $\mathfrak{a} = \text{span}_{\mathbb{Z}}(\alpha, \beta)$ be an $\mathcal{O}$ -ideal. Then $\mathfrak{a}$ is proper if and only if it is invertible. Moreover, whenever $\mathfrak{a}$ is invertible, we have $\mathfrak{a}\bar{\mathfrak{a}} = (N(\mathfrak{a}))$, where $\bar{\mathfrak{a}} = \text{span}_{\mathbb{Z}}(\bar{\alpha}, \bar{\beta})$ and $(N(\mathfrak{a}))$ is the principal $\mathcal{O}$ -ideal generated by $N(\mathfrak{a})$. Then, the inverse of $\mathfrak{a}$ is the fractional $\mathcal{O}$ -ideal $\mathfrak{a}^{-1} = \frac{1}{N(\mathfrak{a})}\bar{\mathfrak{a}}$.*

This last proposition tells us that $I(\mathcal{O})$ is the group under multiplication of proper fractional $\mathcal{O}$ -ideals. If we denote with $P(\mathcal{O})$ the set of principal fractional $\mathcal{O}$ -ideals, we have that $P(\mathcal{O})$ is a subgroup of $I(\mathcal{O})$, and this means that we can take the quotient.

Definition 4.5.20 (Ideal class group). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. We define the ideal class group of $\mathcal{O}$ as the quotient

$$\text{cl}(\mathcal{O}) = I(\mathcal{O})/P(\mathcal{O}).$$

Two of the most important things about the ideal class group are its finiteness and commutativity.

Theorem 4.5.21. [18, Theorem 7.7 (ii)] *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. Then the ideal class group $\text{cl}(\mathcal{O})$ is finite and commutative.*

The cardinality of the ideal class group is very important.

Definition 4.5.22 (Class number). Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$. The class number of $\mathcal{O}$ is

$$h(\mathcal{O}) = |\text{cl}(\mathcal{O})|.$$

Theorem 4.5.23. [57, Theorem 5.28] *Let $\mathcal{O}$ be an order in an imaginary quadratic field $\mathcal{K}$. Then:*

- (1) *Every ideal class in $\text{cl}(\mathcal{O})$ contains infinitely many ideals of prime norm.*
- (2) *Given a non-zero integer n , every ideal class in $\text{cl}(\mathcal{O})$ contains a proper $\mathcal{O}$ -ideal whose norm is relatively prime to n .*

Now we see that the ideal norm is multiplicative if we consider proper $\mathcal{O}$ -ideals.

Proposition 4.5.24. [57, Proposition 5.31] *Let $\mathcal{O}$ be an order in a quadratic field $\mathcal{K}$ and let $\mathfrak{a}, \mathfrak{b}$ two invertible (equivalently, proper) fractional $\mathcal{O}$ -ideals. Then*

$$N(\mathfrak{a}\mathfrak{b}) = N(\mathfrak{a})N(\mathfrak{b}).$$

Definition 4.5.25 (Legendre symbol). Let $p > 2$ be a prime number and let $a \in \mathbb{Z}$. We define the Legendre symbol as follows:

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } t^2 = a \text{ has a solution } t \in \mathbb{F}_p \text{ and } a \not\equiv 0 \pmod{p} \\ -1 & \text{if } t^2 = a \text{ has no solution } t \in \mathbb{F}_p \\ 0 & \text{if } a \equiv 0 \pmod{p}. \end{cases}$$

For $p = 2$ we define

$$\left(\frac{a}{2}\right) = \begin{cases} 1 & \text{if } a \equiv \pm 1 \pmod{8} \\ -1 & \text{if } a \equiv \pm 3 \pmod{8} \\ 0 & \text{if } p \mid a. \end{cases}$$

Theorem 4.5.26. [57, Theorem 5.33] Let $\mathcal{K}$ be an imaginary quadratic field of discriminant $d_{\mathcal{K}}$ with ring of integers $\mathcal{O}_{\mathcal{K}} = \text{span}_{\mathbb{Z}}(1, \omega)$ and let p be a prime integer. Every $\mathcal{O}_{\mathcal{K}}$ -ideal of norm p is of the form

$$\mathfrak{p} = \text{span}_{\mathbb{Z}}(p, \omega - r),$$

where $r \in \mathbb{Z}$ is a root of the minimal polynomial of ω modulo p . Moreover, the number of such ideals $\mathfrak{p}$ is $1 + \left(\frac{d_{\mathcal{K}}}{p}\right) \in \{0, 1, 2\}$ and the factorization of the principal $\mathcal{O}_{\mathcal{K}}$ -ideal generated by p into prime ideals is

$$(p) = \begin{cases} \mathfrak{p}\bar{\mathfrak{p}} & \text{if } \left(\frac{d_{\mathcal{K}}}{p}\right) = 1 \\ \mathfrak{p}^2 & \text{if } \left(\frac{d_{\mathcal{K}}}{p}\right) = 0 \\ (p) & \text{if } \left(\frac{d_{\mathcal{K}}}{p}\right) = -1, \end{cases}$$

with $\mathfrak{p} \neq \bar{\mathfrak{p}}$ in the first case.

This theorem implies that there are only three possibilities for the factorization of the ideal $p\mathcal{O}_{\mathcal{K}}$ in $\mathcal{O}_{\mathcal{K}}$, and we define them as follows.

Definition 4.5.27. Let p be a prime integer and let $\mathcal{K}$ be an imaginary quadratic field. We say that in $\mathcal{O}_{\mathcal{K}}$ the principal ideal $(p) = p\mathcal{O}_{\mathcal{K}}$

- (1) splits completely if it factors into two distinct prime ideals with norm equal to p .
- (2) ramifies if it is the square of a prime ideal.
- (3) is inert if it is already prime.

There is also a generalization of Theorem 4.5.26 to $\mathcal{O}$ -ideals with $\mathcal{O}$ an order in an imaginary quadratic field.

Theorem 4.5.28. [57, Corollary 5.39] Let $\mathcal{O}$ be an order of discriminant D in an imaginary quadratic field $\mathcal{K}$ and let p be a prime integer. When $p \mid [\mathcal{O}_{\mathcal{K}} : \mathcal{O}]$ there are no proper $\mathcal{O}$ -ideals of norm p . Otherwise, there are $1 + \left(\frac{D}{p}\right) \in \{0, 1, 2\}$ proper $\mathcal{O}$ -ideals of norm p depending, respectively, whether p is inert, ramified or splits as a principal $\mathcal{O}_{\mathcal{K}}$ ideal.

From what we saw before in this chapter, we have that, for any complex lattice $\Lambda \subseteq \mathbb{C}$, we have ring isomorphisms

$$\text{End}(E_{\Lambda}) \cong \{\alpha \in \mathbb{C} \mid \alpha\Lambda \subseteq \Lambda\} \cong \text{End}(\mathbb{C}/\Lambda).$$

We focus on the complex multiplication case, where $\text{End}(E_{\Lambda})$ is an order $\mathcal{O}$ in an imaginary quadratic field $\mathcal{K}$. In this case we say that E_{Λ} has complex multiplication by $\mathcal{O}$.

The order $\mathcal{O}$ is a lattice in itself, and we want to understand how the lattices Λ and $\mathcal{O}$ are related.

Lemma 4.5.29. With the notation as above, if $\Lambda = \mathcal{O}$, we have that $\text{End}(E_{\mathcal{O}}) = \mathcal{O}$.

Proof. If $\alpha \in \text{End}(E_{\mathcal{O}})$, then $\alpha\mathcal{O} \subseteq \mathcal{O}$ and so $\alpha \in \mathcal{O}$, since $1 \in \mathcal{O}$. Vice versa, if $\alpha \in \mathcal{O}$, then $\alpha\mathcal{O} \subseteq \mathcal{O}$, i.e. $\alpha \in \text{End}(\mathcal{O})$. $\square$

The same claim holds if we take as Λ any complex lattice homothetic to $\mathcal{O}$, since the set $\{\alpha \in \mathbb{C} \mid \alpha\Lambda \subseteq \Lambda\}$ does not change if we replace Λ with $\Lambda' = \gamma\Lambda$ for any $\gamma \in \mathbb{C} \setminus \{0\}$.

Thanks to what we have seen before we can state the following theorem (for details see [57, Theorem 5.42]), that tells us that there is a correspondence between the ideal class group of an order in an imaginary quadratic field and homothety classes of lattices for which the ring of endomorphisms of the associated elliptic curve is isomorphic to the given order.

Theorem 4.5.30. *Let $\mathcal{O}$ be an order in an imaginary quadratic field $\mathcal{K}$, There is a one-to-one correspondence between elements of the ideal class group $\text{cl}(\mathcal{O})$ and homothety classes of complex lattices $\Lambda \subseteq \mathbb{C}$ for which $\text{End}(E_\Lambda) = \mathcal{O}$.*

Before the next part, we remind that isomorphism classes of elliptic curves over $\mathbb{C}$ are identified by their j -invariant.

Definition 4.5.31. Let $\mathcal{O}$ be an order in an imaginary quadratic field $\mathcal{K}$. We define the set

$$\text{Ell}_{\mathcal{O}}(\mathbb{C}) = \{j(E) \mid E \text{ is defined over } \mathbb{C} \text{ and } \text{End}(E) = \mathcal{O}\}.$$

Remark 4.5.32. The previous discussion tells us that there is a bijection

$$\begin{aligned} \text{cl}(\mathcal{O}) &\xrightarrow{\sim} \text{Ell}_{\mathcal{O}}(\mathbb{C}) \\ [\mathfrak{a}] &\mapsto j(E_{\mathfrak{a}}) = j(\mathfrak{a}). \end{aligned}$$

In particular, since $\text{cl}(\mathcal{O})$ is finite, also the set $\text{Ell}_{\mathcal{O}}(\mathbb{C})$ is finite, with the cardinality equal to the class number $h(\mathcal{O})$.

Let $\mathcal{O}$ be an order in an imaginary quadratic field. We want to define the action of $\text{cl}(\mathcal{O})$ on $\text{Ell}_{\mathcal{O}}(\mathbb{C})$. We do this by defining an action of proper $\mathcal{O}$ -ideals on elliptic curves E defined over $\mathbb{C}$ with complex multiplication by $\mathcal{O}$.

Every elliptic curve $E/\mathbb{C}$ with $\text{End}(E) = \mathcal{O}$ is isomorphic to $E_{\mathfrak{b}}$ for some proper $\mathcal{O}$ -ideal $\mathfrak{b}$, thanks to Theorem 4.5.30. For any proper $\mathcal{O}$ -ideal $\mathfrak{a}$ we define the action of $\mathfrak{a}$ on $E_{\mathfrak{b}}$ via

$$\mathfrak{a}E_{\mathfrak{b}} = E_{\mathfrak{a}^{-1}\mathfrak{b}}.$$

Definition 4.5.33. With the notation as before, we define the action of the equivalence class $[\mathfrak{a}] \in \text{cl}(\mathcal{O})$ on the isomorphism class $j(E_{\mathfrak{b}}) \in \text{Ell}_{\mathcal{O}}(\mathbb{C})$ by

$$[\mathfrak{a}]j(E_{\mathfrak{b}}) = j(E_{\mathfrak{a}^{-1}\mathfrak{b}}),$$

which we can also write as

$$[\mathfrak{a}]j(\mathfrak{b}) = j(\mathfrak{a}^{-1}\mathfrak{b}).$$

We denote the action with $\star$:

$$\begin{aligned} \star : \text{cl}(\mathcal{O}) \times \text{Ell}_{\mathcal{O}}(\mathbb{C}) &\rightarrow \text{Ell}_{\mathcal{O}}(\mathbb{C}) \\ ([\mathfrak{a}], j(E_{\mathfrak{b}})) &\mapsto j(E_{\mathfrak{a}^{-1}\mathfrak{b}}). \end{aligned}$$

One can prove that $\star$ is a group action with certain nice properties.

Proposition 4.5.34. *The action $\star : \text{cl}(\mathcal{O}) \times \text{Ell}_{\mathcal{O}}(\mathbb{C}) \rightarrow \text{Ell}_{\mathcal{O}}(\mathbb{C})$ defined above is a group action of $\text{cl}(\mathcal{O})$ on $\text{Ell}_{\mathcal{O}}(\mathbb{C})$. Moreover it is free and transitive.*

Definition 4.5.35. An action of a group G on a set X that is both free and transitive is said to be regular or simply transitive. In this case, the set X is said to be a G -torsor. Equivalently, the group action is regular if and only if for all $x, y \in X$ there exists a unique $g \in G$ such that $g \cdot x = y$.

In our elliptic curves case, we have that the set $\text{Ell}_{\mathcal{O}}(\mathbb{C})$ is a $\text{cl}(\mathcal{O})$ -torsor.

Now, we look at an example of the action in Proposition 4.5.34.

Example 4.5.36. Consider $\mathcal{K} = \mathbb{Q}(\sqrt{-15})$. The discriminant $d_{\mathcal{K}}$ is -15 and the class number is $h(\mathcal{O}_{\mathcal{K}}) = 2$ ($\mathcal{O}_{\mathcal{K}} = \text{span}_{\mathbb{Z}}(1, \omega_{\mathcal{K}})$). Since $d_{\mathcal{K}} = -15 = 1 \pmod{8}$, we have that

$$\left(\frac{d_{\mathcal{K}}}{2}\right) = 1$$

and then, using Theorem 4.5.26, we see that the ideal (2) splits as

$$(2) = \mathfrak{p}\bar{\mathfrak{p}}, \text{ with } \mathfrak{p} = \text{span}_{\mathbb{Z}}\left(1, \frac{1 + \sqrt{-15}}{2}\right).$$

Since the ideal $\mathfrak{p}$ is not principal, we have that

$$\text{cl}(\mathcal{O}_{\mathcal{K}}) = \{[\mathcal{O}_{\mathcal{K}}], [\mathfrak{p}]\} \cong \mathbb{Z}/2\mathbb{Z}.$$

So, there exists only two elliptic curves over $\mathbb{C}$ up to isomorphism (i.e up to equal j -invariant) with endomorphism ring $\mathcal{O}_{\mathcal{K}}$:

$$E_1 = \mathbb{C}/\mathcal{O}_{\mathcal{K}} \text{ and } E_2 = \mathbb{C}/\mathfrak{p}.$$

So, if $E = E_1 = \mathbb{C}/\mathcal{O}_{\mathcal{K}}$, then $E_2 = E_{\mathfrak{p}}$. Then, the action of $[\mathfrak{p}]$ on $j(E_2)$ is simple to explain:

$$[\mathfrak{p}] \star j(E_2) = [\mathfrak{p}] \star j(E_{\mathfrak{p}}) = j(E_{\mathfrak{p}^{-1}\mathfrak{p}}) = j(E) = j(E_1).$$

The action of $[\mathfrak{p}]$ on $j(E_1)$ is a little more involved. We have that

$$[\mathfrak{p}] \star j(E_1) = [\mathfrak{p}] \star j(E) = j(E_{\mathfrak{p}^{-1}\mathcal{O}_{\mathcal{K}}}).$$

But $[\mathfrak{p}] = [\mathfrak{p}]^{-1} = [\mathfrak{p}^{-1}]$ in $\text{cl}(\mathcal{O}_{\mathcal{K}})$, so

$$E_{\mathfrak{p}^{-1}\mathcal{O}_{\mathcal{K}}} = E_{\mathfrak{p}} = E_2 \text{ and so } [\mathfrak{p}] \star j(E_1) = j(E_2).$$

Now we look at isogenies between elliptic curves with complex multiplication given by the same $\mathcal{O}$. Since any complex isogeny arises from an inclusion of complex lattices, we have the following result.

Proposition 4.5.37. [57, Proposition 5.48] *Let $\mathcal{O}$ be an order in an imaginary quadratic field. Then all elliptic curves over $\mathbb{C}$ with complex multiplication by $\mathcal{O}$ are isogenous. Moreover, up to isomorphism, every isogeny between elliptic curves over $\mathbb{C}$ with complex multiplication by $\mathcal{O}$ is of the form $E_{\mathfrak{b}} \rightarrow \mathfrak{a}E_{\mathfrak{b}} = E_{\mathfrak{a}^{-1}\mathfrak{b}}$, where $\mathfrak{a}, \mathfrak{b}$ are invertible $\mathcal{O}$ -ideals.*

Definition 4.5.38. Let E/K be any elliptic curve with complex multiplication by an imaginary quadratic order $\mathcal{O}$ and let $\mathfrak{a}$ be an $\mathcal{O}$ -ideal. The $\mathfrak{a}$ -torsion subgroup of E is defined by

$$E[\mathfrak{a}] = \{P \in E(\bar{K}) \mid \alpha(P) = 0 \text{ for all } \alpha \in \mathfrak{a}\},$$

where we are viewing each $\alpha \in \mathfrak{a} \subseteq \mathcal{O} \cong \text{End}(E)$ as an endomorphism.

Theorem 4.5.39. [57, Theorem 5.50] *Let $\mathcal{O}$ be an order in an imaginary quadratic field, let $E/\mathbb{C}$ be an elliptic curve with endomorphism ring $\mathcal{O}$, let $\mathfrak{a}$ be an invertible $\mathcal{O}$ -ideal and let $\varphi_{\mathfrak{a}}$ be the corresponding isogeny from E to $\mathfrak{a}E$. Then $\text{Ker}(\varphi_{\mathfrak{a}}) = E[\mathfrak{a}]$ and $\deg(\varphi_{\mathfrak{a}}) = N(\mathfrak{a})$.*

From this theorem we get that if $\mathcal{O}$ is an order in an imaginary quadratic field, $\mathfrak{a}$ is an invertible $\mathcal{O}$ -ideal and $E/\mathbb{C}$ is an elliptic curve with complex multiplication by $\mathcal{O}$. Then the elliptic curves E and $E_{\mathfrak{a}}$ are related by an isogeny $\varphi_{\mathfrak{a}} : E \rightarrow \mathfrak{a}E$ of degree $N(\mathfrak{a})$.

4.6 The Deuring Lifting Theorem and Isogeny graphs

Now, we study two important polynomials, the modular polynomial and the Hilbert class polynomial.

In the following we will use the notation $j(\tau)$ to indicate the j -invariant of the complex lattice $\Lambda_\tau = \text{span}_{\mathbb{Z}}(1, \tau)$. Moreover, we indicate with $j_N(\tau)$ the j -invariant $j(N\tau)$ where $\text{Im}(\tau) > 0$ and $N \in \mathbb{Z}_{\geq 1}$.

Definition 4.6.1 (Modular polynomial). The modular polynomial $\Phi_N \in \mathbb{C}(j)[Y]$ is the minimal polynomial of j_N over $\mathbb{C}(j)$. We can write it as

$$\Phi_N(Y) = (Y - j_N(\tau)) \prod_{k=0}^{N-1} \left(Y - j\left(\frac{\tau + k}{N}\right) \right).$$

One can prove that this definition is well-posed and the coefficients of Φ_N are polynomials in j . This allows us to replace j in it with a new variable X . Thus we obtain a polynomial in $\mathbb{C}[X, Y]$ that we write as $\Phi_N(X, Y)$. It has some properties.

Proposition 4.6.2. *Let $N > 0$ be an integer. The modular polynomial Φ_N satisfies the following properties:*

- (1) $\phi_N \in \mathbb{Z}[X, Y]$.
- (2) $\Phi_N(X, Y) = \Phi_N(Y, X)$.
- (3) Φ_N has degree $N + 1$ in X and Y .
- (4) If N is prime, the leading coefficient of Φ_N is -1 .

The most important theorem for modular polynomials is the following.

Theorem 4.6.3. [57, Theorem 6.4] *Let $N > 0$ be a positive integer and let K be a field such that $\text{char}(K) \nmid N$. For all $j_1, j_2 \in K$ we have that $\Phi_N(j_1, j_2) = 0$ if and only if j_1 and j_2 are the j -invariants of elliptic curves over K that are related by a cyclic isogeny of degree N defined over $\overline{K}$.*

We now describe the Hilbert class polynomial.

Definition 4.6.4 (Hilbert class polynomial). The Hilbert class polynomial associated to an imaginary quadratic order $\mathcal{O}$ is

$$H_{\mathcal{O}}(X) = H_D(X) = \prod_{j \in \text{Ell}_{\mathcal{O}}(\mathbb{C})} (X - j(E)),$$

with $D = \text{disc}(\mathcal{O})$.

Recall that the discriminant D uniquely determines $\mathcal{O}$ and vice versa, so the notation is unambiguous.

Theorem 4.6.5. [57, Theorem 6.10, Corollary 6.11] *Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D . Then $H_D(X) \in \mathbb{Z}[X]$. Moreover, if $E/\mathbb{C}$ is an elliptic curve with complex multiplication, the its j -invariant $j(E)$ is an algebraic integer.*

Definition 4.6.6. Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D . The splitting field $\mathcal{L}$ of the Hilbert class polynomial $H_D(X)$ over $\mathcal{K} = \mathbb{Q}(\sqrt{D})$ is called the ring class field of $\mathcal{O}$.

One can prove that the ring class field is the extension of $\mathcal{K}$ generated by the j -invariant of any elliptic curve $E/\mathbb{C}$ with complex multiplication by $\mathcal{O}$.

We can now generalize the definition of splitting completely of a prime p to any number field, not only quadratic fields.

Definition 4.6.7. Let p be a prime integer and let $\mathcal{L}$ be a number field. We say that p is unramified in $\mathcal{L}$ if the ideal $p\mathcal{O}_{\mathcal{L}}$ factors into distinct prime ideals $\mathfrak{q}$ in $\mathcal{O}_{\mathcal{L}}$. In this case, we say that $\mathfrak{q}$ lies above p . Moreover, we say that p splits completely in $\mathcal{L}$ if the prime ideals $\mathfrak{q} \mid p$ are distinct and have minimal norm $N(\mathfrak{q}) = p$.

Theorem 4.6.8. [57, Theorem 6.14] Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D and ring class field $\mathcal{L}$. Let $p \nmid D$ be an odd prime unramified in $\mathcal{L}$. The following are equivalent:

- (1) p is the norm of a principal $\mathcal{O}$ -ideal.
- (2) $\left(\frac{D}{p}\right) = 1$ and $H_D(X)$ splits into linear factors in $\mathbb{F}_p[X]$.
- (3) p splits completely in $\mathcal{L}$.
- (4) $4p = t^2 - v^2D$ for some integers t, v such that $t \not\equiv 0 \pmod{p}$.

The equation $4p = t^2 - v^2D$ in the theorem above is called the norm equation. When one of the equivalent conditions of the theorem above is met, we say that the norm equation holds. This denomination arises from the principal $\mathcal{O}$ -ideal (λ) of norm p , given by part (1) of the same theorem, generated by a root $\lambda \in \mathcal{O} \subseteq \mathcal{O}_{\mathcal{K}}$ of $x^2 - tx + p$, which has solutions

$$\lambda = \frac{-t \pm \sqrt{t^2 - 4p}}{2} = \frac{-t \pm v\sqrt{D}}{2}.$$

Assume now that the norm equation holds with $p \nmid D$ odd and $D < -4$. The last assumption excludes only the case $\mathcal{O} = \mathbb{Z}[\xi]$ with ξ a non-trivial cubic root of unity, and $\mathcal{O} = \mathbb{Z}[i]$, which corresponds to the j -invariants 0 and 1728, respectively. By Theorem 4.6.8 the prime p splits completely in the ring class field $\mathcal{L}$ for the order $\mathcal{O}$ of discriminant D and we can completely factor $H_D(X)$ in both $\mathcal{O}_{\mathcal{L}}[X]$ and $\mathbb{F}_p[X]$. Now, we fix a prime $\mathfrak{q}$ lying above p . Then $N(\mathfrak{q}) = p$, so we have a reduction map

$$\bar{\cdot} : \mathcal{O}_{\mathcal{L}} \rightarrow \mathcal{O}_{\mathcal{L}}/\mathfrak{q} \cong \mathbb{F}_p$$

that we can apply to $E/\mathbb{C}$ with complex multiplication by $\mathcal{O}$ to obtain the reduction of E , that we denote with $\overline{E}/\mathbb{F}_p$.

Definition 4.6.9 (Good reduction). With the same notation as above, we say that E has good reduction modulo $\mathfrak{q}$ if the reduction $\overline{E}$ is an elliptic curve.

In particular, with the assumptions as before, we can apply the reduction map to the roots of $H_D(X)$, or, equivalently, to the set $\text{Ell}_{\mathcal{O}}(\mathbb{C})$. We have that the j -invariants of any elliptic curve $E/\mathbb{C}$ with complex multiplication by $\mathcal{O}$ can be reduced modulo $\mathfrak{q}$ to the j -invariant of $\overline{E}/\mathbb{F}_p$, which has good reduction.

Proposition 4.6.10. [57, Proposition 6.17] Let $\mathcal{O}$ be an imaginary quadratic order of discriminant $D < -4$ and ring class field $\mathcal{L}$ and let p be an odd prime number unramified in $\mathcal{L}$ such that $p \nmid D$. Suppose that the norm equation holds and let $\mathfrak{q}$ be a prime lying above p . Then the reduction map

$$\bar{\cdot} : \mathcal{O}_{\mathcal{L}} \rightarrow \mathcal{O}_{\mathcal{L}}/\mathfrak{q} \cong \mathbb{F}_p$$

induces an injective ring homomorphism

$$\text{End}(E) \hookrightarrow \text{End}(\overline{E}),$$

where $E/\mathbb{C}$ is an elliptic curve with complex multiplication by $\mathcal{O}$ and $\overline{E}/\mathbb{F}_p$ is its reduction.

Remark 4.6.11. We can generalize everything we have seen to an arbitrary finite field $\mathbb{F}_q$, with q a power of a prime. In Theorem 4.6.8, if $\mathcal{O}$ is an imaginary quadratic order of discriminant D and ring class field $\mathcal{L}$, we can replace $p \nmid D$ with $\gcd(q, D) = 1$, replace $\left(\frac{D}{p}\right) = 1$ with the requirement that D is a square in $\mathbb{F}_q$ and rather than requiring p to split completely in $\mathcal{L}$, we require q to be the norm of a prime ideal $\mathfrak{q}$ in $\mathcal{O}_{\mathcal{L}}$. The norm equation becomes $4q = t^2 - v^2D$ with $\gcd(t, q) = 1$ and, if it satisfies with $D < -4$, the Hilbert class polynomial $H_D(X)$ splits completely in $\mathbb{F}_q[X]$.

Now, we want to be able to use all the tools described on elliptic curves and isogenies defined over $\mathbb{C}$ to elliptic curves and isogenies defined over finite fields, for cryptographic applications.

The following theorems, due to Max Deuring, allows to do exactly this.

Theorem 4.6.12 (Deuring). *Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D with ring class field $\mathcal{L}$ and let $q = p^r$ (p prime) be the norm of a prime ideal in $\mathcal{O}_{\mathcal{L}}$ with $\gcd(q, D) = 1$. Then $H_D(X)$ splits into distinct linear factors in $\mathbb{F}_q[X]$ and its roots form the set*

$$\text{Ell}_{\mathcal{O}}(\mathbb{F}_q) = \{j(E) \in \mathbb{F}_q \mid \text{End}(E) \cong \mathcal{O}\}$$

of j -invariants of elliptic curves $E/\mathbb{F}_q$ with complex multiplication by $\mathcal{O}$.

Proof. It follows from [38, 13.4 Theorem 13] □

Theorem 4.6.13 (Deuring Lifting Theorem). *Let $E/\mathbb{F}_q$ be an elliptic curve over a finite field $\mathbb{F}_q$ ($q = p^r$, p prime) and let $\varphi \in \text{End}(E)$ be non-zero. Then, there exists an elliptic curve E^* over a number field $\mathcal{L}$ with an endomorphism $\varphi^* \in \text{End}(E^*)$ such that E^* has good reduction modulo a prime ideal $\mathfrak{q}$ of $\mathcal{O}_{\mathcal{L}}$ with residue field $\mathcal{O}_{\mathcal{L}}/\mathfrak{q} \cong \mathbb{F}_q$ and E and φ are the reductions modulo $\mathfrak{q}$, respectively, of E^* and φ^* .*

Proof. It follows from [38, 13.5 Theorem 14] □

As a consequence of the two Deuring theorems, we have that the injective ring homomorphism in Proposition 4.6.10 is in fact an isomorphism. Moreover, with the same hypotheses of the proposition, every elliptic curve over $\mathbb{F}_p$ with complex multiplication by $\mathcal{O}$ arises as the reduction of an elliptic curve $E/\mathcal{L}$ and this correspondence is a bijection at the level of j -invariants.

From now on in this section, unless otherwise specified, we assume ℓ to be a prime different from the characteristic of the working field K and all the isogenies we consider are separable.

Proposition 4.6.14. [57, Proposition 6.22] *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ ($q = p^r$, p prime) and let ℓ be a prime different from p . Then, there are $\ell + 1$ distinct isogenies of degree ℓ with domain E defined over the algebraic closure $\overline{\mathbb{F}_q}$. Moreover, there are 0, 1, 2 or $\ell + 1$ isogenies of degree ℓ with domain E defined over $\mathbb{F}_q$.*

We are now able to define ℓ -isogeny graphs.

Definition 4.6.15 (ℓ -isogeny graph). The ℓ -isogeny graph $G_{\ell}(K)$ is the directed graph with vertex set K and edges (j_1, j_2) present with multiplicity equal to multiplicity of j_2 as root of $\Phi_{\ell}(j_1, Y)$. The vertices of $G_{\ell}(K)$ are j -invariants, and its edges correspond to (isomorphism classes of) ℓ -isogenies.

We point out that if $j_1 = j(E_1)$ and $j_2 = j(E_2)$ are the j -invariants of a pair of ℓ -isogenous elliptic curves, the ordered pair (j_1, j_2) does not uniquely determine an ℓ -isogeny $\varphi : E_1 \rightarrow E_2$. There may be multiple non-equivalent ℓ -isogenies from E_1 to E_2 . That is why we need to count edges in $G_\ell(K)$ with multiplicities.

The existence of the dual isogeny guarantees that (j_1, j_2) is an edge in $G_\ell(K)$ if and only if (j_2, j_1) is also an edge. Provided that $j_1, j_2 \neq 0, 1728$, these edges have the same multiplicity, so we can see the subgraph of $G_\ell(K)$ on $K \setminus \{0, 1728\}$ as an undirected graph.

Notice that for any field K the graphs $G_\ell(K)$ all have the same vertex set, but different edge sets, depending on ℓ . So, given an elliptic curve E/K , we can view $j(E)$ as a vertex in any of these graphs.

The types of isogenies that can occur in a graph $G_\ell(K)$ are of three types.

Theorem 4.6.16. [57, Theorem 6.26] *Let $\varphi : E \rightarrow E'$ be an ℓ -isogeny of elliptic curves defined over a field K . Then $\text{End}^0(E) \cong \text{End}^0(E')$. Moreover, if $\text{End}^0(E) = \mathcal{K}$ is an imaginary quadratic field, then $\text{End}(E) = \mathcal{O}$ and $\text{End}(E') = \mathcal{O}'$ are orders in $\mathcal{K}$ such that one of the following holds:*

$$(1) \mathcal{O} = \mathcal{O}' \text{ or } (2) [\mathcal{O} : \mathcal{O}'] = \ell \text{ or } (3) [\mathcal{O}' : \mathcal{O}] = \ell$$

We call these three possibilities in the following way.

Definition 4.6.17. With the notation as in the previous theorem, we distinguish three cases:

- (1) When $\mathcal{O} = \mathcal{O}'$, we say that φ is horizontal.
- (2) When $[\mathcal{O} : \mathcal{O}'] = \ell$, we say that φ is descending.
- (3) When $[\mathcal{O}' : \mathcal{O}] = \ell$, we say that φ is ascending.

We refer to ascending and descending isogenies as vertical isogenies.

If we have a horizontal ℓ -isogeny, the its dual is a horizontal ℓ -isogeny too. If we have a descending ℓ -isogeny, the its dual is an ascending ℓ -isogeny, and vice versa.

Given an elliptic curve $E/\mathbb{C}$, we know exactly how many ℓ -isogenies starting from E of each type there are.

Theorem 4.6.18. [57, Theorem 6.28] *Let $E/\mathbb{C}$ be an elliptic curve with complex multiplication by an order $\mathcal{O}$ of discriminant D in an imaginary quadratic field $\mathcal{K}$ and let ℓ be a prime. If $\ell \nmid [\mathcal{O}_\mathcal{K} : \mathcal{O}]$, then E admits $1 + \left(\frac{D}{\ell}\right)$ horizontal, $\ell - \left(\frac{D}{\ell}\right)$ descending and no horizontal isogenies. Otherwise, E admits no horizontal, ℓ descending and one ascending ℓ -isogeny.*

We can use Deuring Lifting Theorem to extend Theorem 4.6.18 to a finite field $\mathbb{F}_q$ with $\ell \nmid q$.

For an imaginary quadratic order $\mathcal{O}$ with discriminant D and any field K , we define

$$\text{Ell}_\mathcal{O}(K) = \{j(E) \in K \mid \text{End}(E) = \mathcal{O}\}.$$

Lemma 4.6.19. [57, Lemma 6.29] *Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D and let $\mathbb{F}_q$ be a finite field ($q = p^r$, p prime) with $\gcd(q, D) = 1$. Then the set $\text{Ell}_\mathcal{O}(\mathbb{F}_q)$ is either empty or has cardinality $h(\mathcal{O})$. If $\text{Ell}_\mathcal{O}(\mathbb{F}_q)$ is non-empty, so it is $\text{Ell}_{\mathcal{O}'}(\mathbb{F}_q)$ for every imaginary quadratic order $\mathcal{O}'$ containing $\mathcal{O}$.*

This lemma allows us to count how many ℓ -isogenies of each type there are from $E/\mathbb{F}_q$ an elliptic curve defined over a finite field.

Corollary 4.6.20. [57, Corollary 6.30] Let $E/\mathbb{F}_q$ ($q = p^r$, p prime) be an elliptic curve with complex multiplication by an order $\mathcal{O}$ of discriminant D , with $\gcd(q, D) = 1$, in an imaginary quadratic field $\mathcal{K}$ and let $\ell \nmid q$ be a prime. We denote with $\mathcal{O}' \subseteq \mathcal{O}$ the order of index ℓ in $\mathcal{O}$.

(1) If $\text{Ell}_{\mathcal{O}'}(\mathbb{F}_q)$ is non-empty, then E admits

$$\begin{cases} 1 + \left(\frac{D}{\ell}\right) \text{ horizontal}, \ell - \left(\frac{D}{\ell}\right) \text{ descending}, 0 \text{ ascending} & \text{if } \ell \nmid [\mathcal{O}_{\mathcal{K}} : \mathcal{O}] \\ 0 \text{ horizontal}, \ell \text{ descending}, 1 \text{ ascending} & \text{if } \ell \mid [\mathcal{O}_{\mathcal{K}} : \mathcal{O}] \end{cases}$$

ℓ -isogenies.

(2) If $\text{Ell}_{\mathcal{O}'}(\mathbb{F}_q)$ is empty, then E admits

$$\begin{cases} 1 + \left(\frac{D}{\ell}\right) \text{ horizontal}, 0 \text{ descending}, 0 \text{ ascending} & \text{if } \ell \nmid [\mathcal{O}_{\mathcal{K}} : \mathcal{O}] \\ 0 \text{ horizontal}, 0 \text{ descending}, 1 \text{ ascending} & \text{if } \ell \mid [\mathcal{O}_{\mathcal{K}} : \mathcal{O}] \end{cases}$$

ℓ -isogenies.

If $E/\mathbb{F}_q$ is an elliptic curve with complex multiplication by an imaginary quadratic order $\mathcal{O}$ and $\mathfrak{a}$ is a proper $\mathcal{O}$ -ideal, we have that the $\mathfrak{a}$ -torsion subgroup of E is

$$E[\mathfrak{a}] = \{P \in E(\overline{\mathbb{F}_q}) \mid \alpha(P) = 0 \text{ for all } \alpha \in \mathfrak{a}\}.$$

The next result is the extension of Proposition 4.5.34 to finite fields.

Proposition 4.6.21. [57, Corollary 6.31] Let $\mathcal{O}$ be an imaginary quadratic order of discriminant D and let $\mathbb{F}_q$ ($q = p^r$, p prime) be a finite field with $\gcd(q, D) = 1$. If the set $\text{Ell}_{\mathcal{O}}(\mathbb{F}_q)$ is non-empty, then it is a $\text{cl}(\mathcal{O})$ -torsor in which the action of the ideal class of any proper $\mathcal{O}$ -ideal of prime norm $\ell \nmid q$ is given by a horizontal ℓ -isogeny. The inverse of this action is given by the dual isogeny.

We know that every ideal in $\text{cl}(\mathcal{O})$ contains infinitely many proper $\mathcal{O}$ -ideals of prime norm. This means that if we want to compute the action of a given proper $\mathcal{O}$ -ideal $\mathfrak{l}_1$ of prime norm ℓ_1 we can compute it using another proper $\mathcal{O}$ -ideal $\mathfrak{l}_2$ of prime norm ℓ_2 that lies in the same ideal class. This is very useful, because if ℓ_1 is large we can use a much smaller ℓ_2 .

As before we denote the action as

$$\begin{aligned} \star : \text{cl}(\mathcal{O}) \times \text{Ell}_{\mathcal{O}}(\mathbb{F}_q) &\rightarrow \text{Ell}_{\mathcal{O}}(\mathbb{F}_q) \\ ([\mathfrak{a}], j(E_{\mathfrak{b}})) &\mapsto j(E_{\mathfrak{a}^{-1}\mathfrak{b}}). \end{aligned}$$

4.7 An Isogeny-based Cryptographic Construction: CSIDH

At this point, we have all the tools to present a cryptosystems based on the hardness of computing isogenies between elliptic curve, CSIDH [16], first presented in 2018 by Wouter Castryck, Tanja Lange Chloe Martindale, Lorenz Panny and Joost Renes.

CSIDH is a cryptosystem that allows two parties, Alice and Bob, to exchange a common secret key in a public channel.

In the following, where we describe the cryptosystem, we will consider the action $\star$ of the previous section on the representatives. For an elliptic curve E defined over $\mathbb{F}_p$, we let $\text{End}_p(E)$ be the subring of the endomorphism ring $\text{End}(E)$ consisting of endomorphisms defined over $\mathbb{F}_p$. This subring is

isomorphic to an order in a quadratic number field. Conversely, for a given order $\mathcal{O}$ in an imaginary quadratic field, we let

$$\mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$$

denote the set of elliptic curves E defined over $\mathbb{F}_p$ with $\mathcal{O} \xrightarrow{g} \text{End}_p(E)$ such that $\phi_p^E \in \text{End}_P(E)$ corresponds to the p -Frobenius endomorphism of E . In particular, this implies that $\varphi \circ \beta = \beta \circ \varphi$ for all isogenies φ defined over $\mathbb{F}_p$ between two curves in $\mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ and all $\beta \in \mathcal{O}$ interpreted as endomorphisms via θ . Ideals are always assumed to be non-zero.

The following Theorem ([16, Theorem 7]) allows the extension of the $\star$ action to representatives of j -invariant classes of elliptic curves.

Theorem 4.7.1. *Let $\mathcal{O}$ be an order in an imaginary quadratic field such that $\mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ is non-empty. Then the ideal class group $\text{cl}(\mathcal{O})$ acts freely and transitively on the set $\mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, via the map*

$$\begin{aligned} \star : \text{cl}(\mathcal{O}) \times \mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p) &\rightarrow \mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p) \\ ([\mathfrak{a}], E_{\mathfrak{b}}) &\mapsto E_{\mathfrak{a}^{-1}\mathfrak{b}}, \end{aligned}$$

where $\mathfrak{a}$ is chosen as a proper representative and $E = E_{\mathfrak{b}}$ for some unique proper $\mathcal{O}$ -ideal $\mathfrak{b}$.

The following proposition ([16, Proposition 8]) shows that the Montgomery coefficient forms a unique representative for the $\mathbb{F}_p$ -isomorphism class resulting from the $\star$ group action, so can be used as a shared secret without having to take j -invariants.

Proposition 4.7.2. *Let $p \geq 5$ be a prime such that $p \equiv 3 \pmod{8}$, and let $E/\mathbb{F}_p$ be a supersingular elliptic curve. Then $\text{End}_p(E) = \mathbb{Z}[\phi_p^E]$ if and only if there exists $A \in \mathbb{F}_p$ (the Montgomery coefficient) such that E is isomorphic over $\mathbb{F}_p$ to the curve $E_A : y^2 = x^3 + Ax^2 + x$, which, in that case, is said to be in Montgomery form. Moreover, if such A exists it is unique.*

Therefore, by choosing public keys to consist of a Montgomery coefficient $A \in \mathbb{F}_p$, the previous result guarantees that A represents a curve in the correct isogeny class $\mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, where $\mathcal{O} = \mathbb{Z}[\phi_p^E]$, under the assumption that it is smooth (i.e. $A \neq 2$) and supersingular.

We are now ready to describe the scheme.

The setup of the scheme is the following: global parameters of the scheme are a large prime $p = 4 \cdot \ell_1 \cdots \ell_n - 1$, where the ℓ_i s are small pairwise distinct primes > 2 , and the supersingular elliptic curve

$$E_0 : y^2 = x^3 + x \text{ defined over } \mathbb{F}_p,$$

that has endomorphism ring $\text{End}(E) = \mathcal{O} = \mathbb{Z}[\phi_q^{E_0}]$.

The private key is a n -tuple $(e_1, \dots, e_n) \in \mathbb{Z}^n$, with each integer e_i sampled randomly in $\{-m, \dots, m\}$. These integers represent the ideal class $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}] \in \text{cl}(\mathcal{O})$, where $\mathfrak{l}_i = (\ell_i, \phi_q^{E_0} - 1)$. The public key is the coefficient (called Montgomery coefficient) $A \in \mathbb{F}_p$ of the elliptic curve

$$[\mathfrak{a}] \star E_0 : y^2 = x^3 + Ax^2 + x$$

obtained by applying the action of $[\mathfrak{a}]$ to the curve E_0 (where $E_0 = E_{\mathfrak{c}}$ for some proper $\mathcal{O}$ -ideal $\mathfrak{c}$).

Suppose that Alice and Bob have (secret key, public key) pairs, respectively $([\mathfrak{a}], A)$ and $([\mathfrak{b}], B)$. Upon receiving Bob's public key $B \in \mathbb{F}_p \setminus \{\pm 2\}$, Alice verifies that the curve $E_B : y^2 = x^3 + Bx^2 + x$ is indeed in $\text{Ell}_{\mathcal{O}}(\mathbb{F}_p)$. She then applies the action of her secret key $[\mathfrak{a}]$ to E_B and computes the curve

$$[\mathfrak{a}] \star E_B = [\mathfrak{a}] \star [\mathfrak{b}] \star E_{\mathfrak{c}}.$$

Bob proceeds analogously and with his own secret $[\mathfrak{b}]$ and Alice's public key A he computes the curve

$$[\mathfrak{b}] \star E_A = [\mathfrak{b}] \star [\mathfrak{a}] \star E_c.$$

The shared secret is the Montgomery coefficient S of the common secret curve

$$[\mathfrak{a}] \star [\mathfrak{b}] \star E_c = [\mathfrak{b}] \star [\mathfrak{a}] \star E_c,$$

written as $y^2 = x^3 + Sx^2 + x$, which is the same for Alice and Bob thanks to the commutativity of $\text{cl}(\mathcal{O})$ and Proposition 4.7.2.

The security of this scheme is based on the hardness of "inverting" the action $\star$: given $E_0 = E_c, E_A, E_B$ it is really hard to find $\mathfrak{a}$ and $\mathfrak{b}$. This means that it is hard to find the isogeny between E_0 and E_A and between E_0 and E_B . This is the key problem in isogeny-based cryptography when we are dealing with supersingular elliptic curves.

Chapter 5

Overview of a Post-Quantum Lattice-based HIBBE scheme

Identity-based encryption is a public-key cryptographic tool designed to simplify key management and improve the efficiency of secure communications. Unlike traditional public-key cryptography, where users must generate public keys and obtain digital certificates from a certificate authority, identity-based encryption allows a user's identity to function directly as the public key. Information such as an email address, username, or phone number can therefore be used to encrypt messages securely. This concept reduces the complexity associated with certificate management and makes cryptographic systems more practical for large-scale and distributed environments.

The idea of identity-based encryption was first introduced by Shamir in 1984 in [58] as an alternative to conventional public-key infrastructures. In a typical identity-based system, a trusted entity known as the manager M is responsible for generating system parameters and issuing private keys to users based on their identities. When a sender wants to encrypt a message, they simply use the recipient's identity and the public system parameters, without needing access to a separate public certificate. The recipient can then decrypt the message using the private key obtained from the manager.

One of the main advantages of identity-based encryption is the simplification of public-key distribution. Since identities are inherently public and easy to verify, there is no longer a need for complex certificate infrastructures. This makes identity-based encryption particularly suitable for applications such as secure email systems, mobile communication networks, cloud computing, and Internet of Things (IoT) environments, where scalability and ease of management are essential. Additionally, it enables encryption before the recipient has even generated or received a private key, which provides flexibility in asynchronous communication systems.

In this chapter we will deal with an identity-based encryption scheme which is also hierarchical and broadcast: we add a hierarchy in the set of senders and receivers (which are all depending on the manager M) and we allow the communications to be performed over a broadcast channel, where only a selected number of users can decrypt the content that is being sent.

As we pointed out in the introduction, lattice-based cryptography is one of the strongest contenders for the post-quantum transition, with two cryptosystems standardized by NIST in 2024, CRYSTALS - Kyber and CRYSTALS - Dilithium. The goal of this chapter is to give a detailed overview of the construction of a lattice-based hierarchical identity-based broadcast encryption scheme, which is one of the first of its kind to be post-quantum. In particular, we have written down explicitly (for the first time) some of the key algorithms that constitute the cryptosystems, such as **TrapGen**, **SamplePre**, **BasisDel** and the hash function $\mathcal{H}$ that samples from a discrete Gaussian distribution. Moreover, we

have also made a Python implementation of it.

5.1 Additional Lattice Algorithms for the scheme

We now want to investigate the hierarchical identity-based broadcast encryption (HIBBE) scheme found in the article [67] by C. Yang, Zheng, Wang, Lu and Y. Yang. In particular we will be able to write down explicitly as pseudo-code all the algorithms that make this construction work, and we include the results of their correctness. Some of the procedures we use inside the HIBBE scheme were investigated in Chapter 3. In particular, the Micciancio-Goldwasser Algorithm **MG** (Theorem 3.4.2), for which we gave a pseudo-code description and an extended proof of correctness for the first time, will be a key feature in the cryptosystem we are about to describe.

Definition 5.1.1 (q -ary lattice). Let $A \in M_{n,m}(\mathbb{Z}/q\mathbb{Z})$ for some positive integers $q, m, n > 0$. The q -ary lattice associated to A is the lattice

$$\Lambda_q^\perp(A) = \{z \in \mathbb{Z}^m \mid Az = 0 \pmod{q}\}.$$

We also define the coset for any $y \in \mathbb{Z}_q^n$ that is generated by the columns of A as

$$\Lambda_q^y(A) = \{z \in \mathbb{Z}^m \mid Az = y \pmod{q}\}.$$

The previous set is called coset of $\Lambda_q^\perp(A)$ because, if we have that $y = Ax$ for some $x \in \mathbb{Z}^n$, then

$$\Lambda_q^y(A) = \Lambda_q^\perp(A) + x = \{z + x \mid z \in \Lambda_q^\perp(A)\}.$$

Lemma 5.1.2 (TrapGen). [67, Lemma 2.1] *Let $q \geq 3$ be odd and $m = \lceil 6n \log q \rceil$. There is a probabilistic polynomial-time algorithm **TrapGen**(q, n) that outputs a pair $(A, B_A) \in M_{n,m}(\mathbb{Z}/q\mathbb{Z}) \times M_n(\mathbb{Z})$ such that A is a matrix that is statistically close to a uniform matrix in $M_{n,m}(\mathbb{Z}/q\mathbb{Z})$ and B_A is a basis for $\Lambda_q^\perp(A)$ that satisfies*

$$\|\widetilde{B_A}\| \leq O(\sqrt{n \log q}) \text{ and } \|B_A\| \leq \mathcal{O}(n \log q),$$

where $\widetilde{}$ is, as before, the Gram-Schmidt orthogonalization process done on the columns from left to right (without normalization). This bounds work with all but negligible probability in n .

This algorithm is essentially Algorithm 1 in [3] in its first construction, with $r = 2$ and $\delta = 1$.

TrapGen: Trapdoor Generation Algorithm

Input: a random matrix $A_1 \in M_{n,m_1}(\mathbb{Z}/q\mathbb{Z})$ such that $m_1 + m_2 = m$ and a basis B of $\Lambda_q^\perp(A_1)$

Output: a matrix $A_2 \in M_{n,m_2}(\mathbb{Z}/q\mathbb{Z})$, $A = (A_1|A_2) \in M_{n,m}(\mathbb{Z}/q\mathbb{Z})$ and a basis B_A of $\Lambda_q^\perp(A)$

```

1 :  $H, - = \text{HermiteNormalForm}(B)$  ### on the columns; we are only interested in the resulting matrix
2 :  $H' = H - I_{m_1}$ 
3 : parse  $H' = [h_1, \dots, h_{m_1}]$  ### as columns
4 :  $\ell = \log_2(q)$ 
5 : for  $i = 1$  to  $m_1$  do
6 :    $g_\ell^{(i)} = h'_i$ 
7 :   for  $j = \ell - 1$  down to  $1$  do
8 :      $g_j^{(i)} = \left\lfloor \frac{g_{j+1}^{(i)}}{2} \right\rfloor = \left( \left\lfloor \frac{h'_i}{2^{\ell-j}} \right\rfloor \right)$ 
9 :   endfor
10 : endfor
11 : for  $i = 1$  to  $m - 1$  do
12 :    $G^{(i)} = [g_1^{(i)}, \dots, g_\ell^{(i)}] \in M_{m_1, \ell}(\mathbb{Z})$ 
13 : endfor
14 :  $G = (G^{(1)} | \dots | G^{(m_1)} | 0_{m_1 \times m_2 - m_1 \ell})$ 
15 : for  $i = 1$  to  $m_1$  do
16 :    $p_j = e_{j\ell} \in \mathbb{Z}^{m_2}$  ### the  $j\ell$ -th elementary column basis vector
17 : endfor
18 :  $P = (p_1, \dots, p_{m_1}) \in M_{m_2, m_1}(\mathbb{Z})$ 
19 : construct  $T_\ell = (t_{i,j}) \in M_\ell(\mathbb{Z})$  such that
20 :    $t_{i,i} = 1$  for all  $i = 1, \dots, \ell$ 
21 :    $t_{i,i+1} = -2$  for all  $i = 1, \dots, \ell - 1$ 
22 :    $t_{i,j} = 0$  elsewhere
23 :  $U = \text{diag}(\overbrace{T_\ell, \dots, T_\ell}^{m_1 \text{ times}}, I_{m_2 - m_1 \ell}) \in M_{m_2}(\mathbb{Z})$  ### diag is used to concatenate matrices in the diagonal
24 :  $d = 2n \log(q)$ 
25 : construct  $R = (r_{i,j}) \in M_{m_1, m_2}(\mathbb{Z})$  such that
26 :   if  $i > d, r_{i,j} = 0$ 
27 :   if  $i \leq d, \mathbb{P}[r_{i,j} = 0] = \frac{1}{2}, \mathbb{P}[r_{i,j} = 1] = \frac{1}{4}, \mathbb{P}[r_{i,j} = -1] = \frac{1}{4}$ 
28 :  $A_2 = -A_1(R + G), A = (A_1|A_2)$ 
29 :  $B_A = \begin{pmatrix} (G + R)U & RP \\ U & P \end{pmatrix} \in M_m(\mathbb{Z})$ 
30 : return  $(A_2, A, B_A)$ 
```

We denote with $L = \mathcal{O}(\sqrt{n \log q})$ the maximum Gram-Schmidt norm of a basis B_A produced by **TrapGen**(q, n).

To describe the algorithm we need a other specific algorithms to sample on lattices and cosets associated to lattices.

First of all, we define a distribution on the coset $\mathcal{L} + y$ of a lattice $\mathcal{L} \subseteq \mathbb{R}^n$ with respect to $y \in \mathbb{R}^n$

and $s > 0$:

$$\mathcal{D}_{\mathcal{L}+y,s}(x) = \frac{\rho_{s,0}(x)}{\sum_{z \in \mathcal{L}} \rho_{s,0}(y+z)}$$

for all $x \in \mathcal{L} + y$.

Regarding cosets of q -ary lattices, the following lemma guarantees that we can sample from them.

Lemma 5.1.3 (SamplePre). *[67, Lemma 2.2 3.] Let $q \geq 2$ and $A \in \mathbb{M}_{n,m}(\mathbb{Z}/q\mathbb{Z})$ such that $m > n$. Let B_A be a basis for $\Lambda_q^\perp(A)$ and $\sigma \geq \|\widetilde{B}_A\| \omega(\sqrt{\log m})$. Then, for $u \in (\mathbb{Z}/q\mathbb{Z})^n$,*

$$\mathbb{P} \left[\|x\|_2 > \sigma\sqrt{m} \mid x \xleftarrow{\mathcal{D}_{\Lambda_q^u(A),\sigma}} \Lambda_q^u(A) \right] \leq \text{negl}(n),$$

and there is a probabilistic polynomial-time algorithm **SamplePre** that with inputs A, B_A, u and σ returns $x \in \Lambda_q^u(A)$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^u(A),\sigma}$.

The algorithm is a slight modification of **SampleD**, and we present it here.

SamplePre

Input: a matrix $A \in \mathbb{M}_{n,m}(\mathbb{Z}/q\mathbb{Z})$, a basis B_A of $\Lambda_q^\perp(A)$, a vector $t \in (\mathbb{Z}/q\mathbb{Z})^m$ and a parameter $\sigma > 0$

Output: an element $x \in \Lambda_q^t(A)$

```

1:  $c = \text{Solve}(As = t)$  ### in  $\mathbb{Z}/q\mathbb{Z}$ 
2:  $c' =$  Representative of  $c$  over  $\mathbb{Z}^n$  with coefficients in  $\{0, \dots, q-1\}$ 
3:  $d \leftarrow \text{SampleD}(B_A, \sigma, -c')$ 
4:  $x = c + d$ 
5: return  $x$ 

```

The main tool of this HIBBE is the basis delegation technique, that allows to delegate a short lattice basis and has the advantage of maintaining the lattice dimension unchanged.

Before diving into the description of the basis delegation, we introduce an algorithm that allows the randomization of a given lattice basis: **RandBasis**.

Theorem 5.1.4 (RandBasis). *[14, Lemma 3.3] The **RandBasis** algorithm is a probabilistic polynomial-time algorithm that, given a basis B of a m -dimensional q -ary lattice $\Lambda_q^\perp(A)$, outputs a basis B' of $\Lambda_q^\perp(A)$ such that with overwhelming probability $\|\widetilde{B}'\| \leq \sigma\sqrt{m}$, and up to a statistical distance, the distribution of B' does not depend on B . That is, the random variable $\text{RandBasis}(B, \sigma)$ is statistically close to $\text{RandBasis}(C, \sigma)$ for any other basis C of $\Lambda_q^\perp(A)$ satisfying $\|\widetilde{C}\| \leq \sigma/\omega(\sqrt{\log n})$. In the algorithm, the canonical basis BC of $\Lambda_q^\perp(A)$ can be obtained from A using linear algebra methods. In general, this basis consists of very long vectors compared to the ones in useful (and private) short bases.*

RandBasis: Basis Randomization Algorithm

Input: a basis B of an m -dimensional lattice $\Lambda_q^\perp(A)$, a gaussian parameter $\sigma \geq \|\tilde{B}\|\omega(\sqrt{\log n})$

Output: a basis B' for $\Lambda_q^\perp(A)$

```

1 :  $V = []$ 
2 : for  $i = 1$  to  $m$  do
3 :    $v \leftarrow \text{SampleD}(B, \sigma, 0)$ 
4 :   if  $v \notin \text{span}_{\mathbb{R}}(V)$  then
5 :      $V = V \cup [v]$ 
6 :   else
7 :     goto line 3
8 :   endif
9 : endfor
10 : parse  $V = [v_1, \dots, v_m]$ 
11 :  $BC =$  a canonical basis for  $\Lambda_q^\perp(A)$  ### It can be obtained from  $A$  with linear algebra methods
12 :  $B' = \mathbf{MG}(BC, V)$ 
13 : return  $B'$ 

```

If we are given a matrix $A \in M_{n,n}(\mathbb{Z}/q\mathbb{Z})$ and a short basis B for the lattice $\Lambda_q^\perp(A)$ and a matrix $R \in M_m(\mathbb{Z})$ which is $\mathbb{Z}/q\mathbb{Z}$ -invertible (i.e. it is invertible modulo q), the basis delegation algorithm outputs a matrix $B_{AR_q^{-1}} \in M_m(\mathbb{Z})$ that is a short basis for the lattice $\Lambda_q^\perp(AR_q^{-1})$, where $R_q = R \bmod q$.

At the heart of the basis delegation lies the Micciancio-Goldwasser Algorithm (Theorem 3.4.2), which is also used in **RandBasis**.

Let $\sigma_R = L\omega(\sqrt{\log m})$ and let $\mathcal{D}_{m \times m} = (\mathcal{D}_{\mathbb{Z}^m, \sigma_R, 0})^m$ denote a distribution on $\mathbb{Z}/q\mathbb{Z}$ -invertible matrices in $M_m(\mathbb{Z})$ (in particular, on their columns). We can now explain how the basis delegation algorithm works.

BasisDel: Basis Delegation Algorithm

Input: a matrix $A \in M_{n,m}(\mathbb{Z}/q\mathbb{Z})$, a basis B of the m -dimensional lattice $\Lambda_q^\perp(A) \subseteq \mathbb{Z}^m$,
a $\mathbb{Z}/q\mathbb{Z}$ -invertible matrix R sampled from $\mathcal{D}_{m \times m}$, a parameter $\sigma \geq \|\tilde{B}\|\sigma_R\sqrt{m}\omega(\sqrt{\log^{3/2} m})$.

Output: a basis $B_{AR_q^{-1}}$ for the lattice $\Lambda_q^\perp(AR_q^{-1})$, where $R_q = R \bmod q$

```

1 : parse  $B = [b_1, \dots, b_m] \subseteq \mathbb{Z}^m$  ### the columns of  $B$  are parsed
2 :  $S' = [Rb_1, \dots, Rb_m]$  ###  $S'$  consists of linearly independent vectors in  $\Lambda_q^\perp(AR_q^{-1})$ 
3 :  $BC_{AR_q^{-1}} =$  a canonical basis for  $\Lambda_q^\perp(AR_q^{-1})$ 
4 :  $S'' = \mathbf{MG}(BC_{AR_q^{-1}}, S')$ 
5 :  $B_{AR_q^{-1}} \leftarrow \text{RandBasis}(S'', \sigma)$ 
6 : return  $B_{AR_q^{-1}}$ .

```

We have that [1, Theorem 3] by Shweta Agrawal, Dan Boneh and Xavier Boyen guarantees that $B_{AR_q^{-1}}$ is distributed statistically close to the distribution of $\text{RandBasis}(C, \sigma)$, where C is an arbitrary basis of $\Lambda_q^\perp(AR_q^{-1})$ satisfying $\|\tilde{C}\| < \sigma/\omega(\sqrt{\log m})$. Moreover, if R is a product of matrices sampled from $\mathcal{D}_{m \times m}$, then the bound on σ degrades to $\sigma > \|\tilde{B}\| \cdot \sigma_R\sqrt{m}\omega(\log^{1/2} m) \cdot \omega(\log m)$. When R is a product for ℓ matrices sampled from $\mathcal{D}_{m \times m}$, then for the smallest possible σ we obtain that with high

probability

$$\frac{\|\widetilde{B_{AR_q^{-1}}}\|}{\|\widetilde{B}\|} \leq (m\omega(\log m))^\ell \sqrt{m}\omega(\log m).$$

This quantity is the minimum degradation in basis quality as we delegate across levels of the HIBBE hierarchy.

The last algorithm needed for the HIBBE scheme is the generalized preimage sampleable algorithm **GenSamplePre**.

Let $A \in M_{n,km}(\mathbb{Z}/q\mathbb{Z})$ and write $A = (A_1|A_2|\dots|A_k)$, where $A_i \in M_{n,m}(\mathbb{Z}/q\mathbb{Z})$ for all $i = 1, \dots, k$. For $S \subseteq [1, \dots, k]$, $S = [i_1, i_2, \dots, i_j]$ we write A_S to denote $(A_{i_1}|A_{i_2}|\dots|A_{i_j})$.

Lemma 5.1.5 (GenSamplePre). *[13, Theorem 3.4] Let n, q, m, k be positive integers with $q \geq 2$ and $m \geq 2n \log q$. There is a probabilistic polynomial-time algorithm **GenSamplePre** that takes as inputs $A \in M_{n,km}(\mathbb{Z}/q\mathbb{Z})$, a ordered list $S \subseteq [1, \dots, k]$, $S = [i_1 < i_2 < \dots < i_j]$, a basis B_S for $\Lambda_q^\perp(A_S)$, a vector $y \in (\mathbb{Z}/q\mathbb{Z})^n$ and an integer $r \geq \|\widetilde{B_S}\|\omega(\sqrt{\log km})$ and outputs a vector $e \in \mathbb{Z}^{km}$ which is within negligible statistical distance from the distribution $\mathcal{D}_{\Lambda_q^y(A),r}$ for an overwhelming fraction of $A \in M_{n,km}(\mathbb{Z}/q\mathbb{Z})$.*

In the algorithm, without loss of generality, we put $S = [1, \dots, j] \subseteq [1, \dots, k]$.

GenSamplePre

Input: a matrix $A \in M_{n,km}(\mathbb{Z}/q\mathbb{Z})$, an ordered list $S = [1, \dots, j] \subseteq [1, \dots, k]$, a basis B_S of $\Lambda_q^\perp(A_S)$, a vector $y \in (\mathbb{Z}/q\mathbb{Z})^n$ and an integer $r \geq \|\widetilde{B_S}\|\omega(\sqrt{\log km})$

Output: a vector $e \in \mathbb{Z}^{km}$

```

1:  $S^c = [1, \dots, k] \setminus S = [j+1, \dots, k]$ 
2:  $e_{S^c} \leftarrow \mathbf{SampleD}(I_{(k-j)m}, r, 0)$ 
3:  $z = y - A_{S^c}e_{S^c}$ 
4: parse  $e_{S^c} = (e_{j+1}^T, \dots, e_k^T)^T \in \mathbb{Z}^{(k-j)m}$ 
5:  $e_S \leftarrow \mathbf{SamplePre}(A_S, B_S, z)$ 
6: parse  $e_S = (e_1^T, \dots, e_j^T)^T \in \mathbb{Z}^{jm}$ 
7:  $e = (e_1^T, \dots, e_j^T, e_{j+1}^T, \dots, e_k^T)^T \in \mathbb{Z}^{km}$ 
8: return  $e$ 
```

5.2 Construction of the Hash function

The HIBBE scheme we are dealing with needs a particular hash function

$$\begin{aligned} \mathcal{H} : \{0, 1\}^* &\rightarrow M_m(\mathbb{Z}/q\mathbb{Z}) \\ \text{id} &\mapsto \mathcal{H}(\text{id}) \sim \mathcal{D}_{m \times m} = (\mathcal{D}_{\mathbb{Z}^m, \sigma_R, 0})^m \end{aligned}$$

i.e. the columns of $\mathcal{H}(\text{id})$ should be seen as sampled from a discrete Gaussian distribution, conditioned on the fact that $\mathcal{H}(\text{id})$ has to be $\mathbb{Z}/q\mathbb{Z}$ -invertible.

Our construction follows the idea in [1, Section 4]. The idea is to exploit the hash function SHA-256 [46]:

$$\text{SHA-256} : \{0, 1\}^* \rightarrow \{0, 1\}^{256}.$$

We explain how does the hash function work. for an identity string $\text{id} \in \{0, 1\}^\ell$ for some $\ell \in \mathbb{Z}_{\geq 1}$ we perform the following steps.

- (1) Compute $\text{RandBits} = \text{SHA-256}(\text{id}) \in \{0, 1\}^{256}$.
- (2) Run a pseudo-random version of **SampleD**, on inputs a basis I_m for $\mathbb{Z}^m$, the parameter $\sigma_R > 0$ and center 0, where, for randomness, we use the pseudo-random bits in RandBits , until we get m vectors that generate a $\mathbb{Z}/q\mathbb{Z}$ -invertible matrix in $M_m(\mathbb{Z})$.

If we need more randomness, we attach $\text{SHA-256}(\text{RandBits})$ to RandBits to get 512 random bits. If we need more we shall do that again (attaching $\text{SHA-256}(\text{RandBits})$ to RandBits , so the process is all deterministic.

$\mathcal{H}$ Hash Function

Input: a bit string $\text{id} \in \{0, 1\}^*$
Output: a matrix $M \in M_m(\mathbb{Z})$

```

1 : RandBits = SHA-256(id)
2 : IsInvertible = false
3 : while IsInvertible == false do
4 :   for j = 1 to m do
5 :      $v_j = \text{SampleD}(I_m, \sigma_R, 0, \text{RandBits})$  ###Notice that here SampleD is deterministic, given RandBits
       if we need more pseudo-random bits in SampleD then
         RandBits = (RandBits, SHA-256(RandBits)) ###Run it for more pseudo-randomness
       endif
6 :   endfor
7 :    $M = [v_1, \dots, v_m] \in M_m(\mathbb{Z})$ 
8 :   if gcd(det(M), q) = 1 then
9 :     IsInvertible = true
10 :   endif
11 : endwhile
12 : return M
```

We are now ready to describe the post-quantum lattice-based HIBBE scheme. We will follow the outline we used in Section 2.5.

5.3 Description of the HIBBE scheme

Suppose everyone in the scheme knows the hash function of the previous section $\mathcal{H} : \{0, 1\}^* \rightarrow M_m(\mathbb{Z}/q\mathbb{Z})$ such that $\text{id} \mapsto \mathcal{H}(\text{id}) \sim \mathcal{D}_{m \times m}$.

Setup: Given a security parameter n , the maximum hierarchy depth d and the maximum size h of the receiver set, the algorithm performs the following steps. It runs the trapdoor generation algorithm **TrapGen** to generate $A \in M_{n,m}(\mathbb{Z}/q\mathbb{Z})$ with the corresponding trapdoor $B \in M_m(\mathbb{Z})$. Note that B is a short basis for $\Lambda_q^\perp(A)$ and $\|B\| \leq L = O(\sqrt{n \log q})$. It generates a uniform random matrix $V \in M_{n,t}(\mathbb{Z}/q\mathbb{Z})$, where t is the length of the symmetric encryption key. It outputs the public parameters pp and master key mk given by $\text{pp} = (A, V)$ and $\text{mk} = B$.

For an l depth identity $\text{id}_{i_l}^l$, if we write its path in the tree as $[\text{id}_{i_1}^1, \text{id}_{i_2}^2, \dots, \text{id}_{i_l}^l]$, we let $R_{\text{id}_{i_l}^l} = \mathcal{H}(\text{id}_{i_l}^l) \mathcal{H}(\text{id}_{i_{l-1}}^{l-1}) \dots \mathcal{H}(\text{id}_{i_1}^1)$ and let the corresponding matrix $A_{\text{id}_{i_l}^l} = A \cdot R_{\text{id}_{i_l}^l}^{-1}$. The secret key for $\text{id}_{i_l}^l$ is a basis $B_{\text{id}_{i_l}^l}$ for $\Lambda_q^\perp(A_{\text{id}_{i_l}^l})$.

Delegate: Given the public parameters pp , a secret key $B_{\text{id}_{i_l}^l}$ for a parent node identity $\text{id}_{i_l}^l$ as $[\text{id}_{i_1}^1, \text{id}_{i_2}^2, \dots, \text{id}_{i_l}^l]$ and a child node identity $\text{id}_{i_s}^s$ as $[\text{id}_{i_1}^1, \text{id}_{i_2}^2, \dots, \text{id}_{i_l}^l, \dots, \text{id}_{i_s}^s]$ where $l < s \leq d$, the algorithm performs the following steps. Let $R = \mathcal{H}(\text{id}_{i_s}^s) \dots \mathcal{H}(\text{id}_{i_{l+1}}^{l+1})$, then

$$A_{\text{id}_{i_s}^s} = A \cdot R_{\text{id}_{i_s}^s}^{-1} = A \cdot R_{\text{id}_{i_l}^l}^{-1} R^{-1} = A_{\text{id}_{i_l}^l} R^{-1}.$$

Run the basis delegation algorithm $B_{\text{id}_{i_s}^s} \leftarrow \mathbf{BasisDel}(A_{\text{id}_{i_l}^l}, B_{\text{id}_{i_l}^l}, R, \sigma_s)$ to obtain a short random basis for $\Lambda_q^\perp(A_{\text{id}_{i_s}^s})$, where $\sigma_s \geq \|\widetilde{B_{\text{id}_{i_l}^l}}\|(\sigma_R \sqrt{m} \omega(\log^{1/2} m))^{s-l} \omega(\log m)$ and $\sigma_R = L\omega(\sqrt{\log m})$. It outputs the secret key $B_{\text{id}_{i_s}^s}$ for the identity $\text{id}_{i_s}^s$, where

$$\frac{\|\widetilde{B_{\text{id}_{i_s}^s}}\|}{\|\widetilde{B_{\text{id}_{i_l}^l}}\|} \leq (m\omega(\log m))^{s-l} \sqrt{m} \omega(\log m).$$

KeyExt: It works exactly as $\mathbf{Delegate}(\text{pp}, \text{mk}, \text{id}_{i_s}^s)$, where $A_{\text{id}_0} = A$ and $B_{\text{id}_0} = B$.

Enc: Given the public parameters pp and a receiver set S (as paths in the trees) it performs the following steps. It remove the parent nodes from S (the parent nodes know their child node's secret keys and can decrypt ciphertexts). For simplicity of notation, assume that the remainder receiver set is $S = [\text{id}_1, \dots, \text{id}_k]$, where $k \leq h$. Let $A_S = (A_{\text{id}_1} | \dots | A_{\text{id}_k}) \in \mathbb{M}_{n, km}(\mathbb{Z}/q\mathbb{Z})$, where, if id_i has depth j and corresponds to $[\text{id}_{i_1}^1, \dots, \text{id}_{i_j}^j] = \text{id}_i$ with $1 \leq j \leq d, 1 \leq i \leq k, k \leq h$, then

$$A_{\text{id}_i} = A \cdot R_{\text{id}_i}^{-1} = A \cdot (\mathcal{H}(\text{id}_{i_j}^j) \mathcal{H}(\text{id}_{i_{j-1}}^{j-1}) \dots \mathcal{H}(\text{id}_{i_1}^1))^{-1}.$$

Define a label $\text{lab}S$ that contains information about the way in which A_S is associated with the sequence of the receivers $[\text{id}_1, \dots, \text{id}_k]$. Choose a vector $s \in (\mathbb{Z}/q\mathbb{Z})^n$ uniformly at random. Compute $p = A_S^T \cdot s + x$, where $x \xleftarrow{(\psi_{\alpha k})^{km}} (\mathbb{Z}/q\mathbb{Z})^{km}$ (see Remark 3.6.2). Choose a symmetric encryption key $K \in \{0, 1\}^t$. For $j = 1, \dots, t$, let K_j be the j -th bit of K . Compute $c = V^T \cdot s + y + K \lfloor \frac{q}{2} \rfloor$, where $y \leftarrow \chi^t$ and $\chi = \psi_{\alpha k}$. Output $\text{Hdr} = (p, c, \text{lab}S)$.

Dec: Given the public parameters pp , the receiver set S and the header Hdr , a receiver identity id_i with secret key B_{id_i} wants to obtain the symmetric encryption key. If $\text{id}_i \in S$, the depth of id_i is l , then the algorithm performs the following steps. Let $r = \sigma_l \sqrt{km} \omega(\sqrt{\log km}) (\geq \|\widetilde{B_{\text{id}_i}}\| \omega(\log km))$. By the information in $\text{lab}S$, set $A_S = (A_{\text{id}_1} | \dots | A_{\text{id}_k}) \in \mathbb{M}_{n, km}(\mathbb{Z}/q\mathbb{Z})$. Parse $V = [v_1, \dots, v_t]$. For $j = 1, \dots, t$ run the algorithm **GenSamplePre** to generate

$$e_j \leftarrow \mathbf{GenSamplePre}(A_S, S, B_{\text{id}_i}, v_j, r),$$

where $r \geq \|\widetilde{B_{\text{id}_i}}\| \omega(\log km)$. We have that e_j is within negligible statistical distance from the distribution $\mathcal{D}_{\Lambda_q^{v_j}(A_S), r}$ and $\|e_j\| \leq r \sqrt{km}$ (see [67]). Parse c as $[c_1, \dots, c_t] \subseteq \mathbb{Z}/q\mathbb{Z}$. For $j = 1, \dots, t$ compute $K'_j = c_j - e_j^T \cdot p \in \mathbb{Z}/q\mathbb{Z}$. Select K''_j as the representative of K'_j in $\{0, \dots, q-1\}$. If $|K''_j - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$ we have $K_j = 1$, otherwise $K_j = 0$. Output $K = (K_1, K_2, \dots, K_t)$.

The HIBBE scheme presented above is correct. This is due to the following lemma.

Lemma 5.3.1. [67, Lemma 2.3] Let $e \in \mathbb{Z}^m$ be a vector and let $y \xleftarrow{(\overline{\psi_\alpha})^m} (\mathbb{Z}/q\mathbb{Z})^m$ be a sample (with $q \in \mathbb{Z}_{>1}$ a modulus). Then $|e^T \cdot y| \bmod q \in \{0, \dots, q-1\}$ (i.e. regarded as an integer in $\{0, \dots, q-1\}$) satisfies

$$|e^T \cdot y| \leq \|e\|_2 \cdot q \cdot \alpha\omega(\sqrt{\log m}) + \|e\|_2 \frac{\sqrt{m}}{2}$$

with probability 1 minus a negligible function in m .

Proposition 5.3.2. [67, Section V.] The HIBBE scheme presented in this chapter is correct.

Chapter 6

Post-Quantum Adaptor Signatures from Cryptographic Group Actions

Adaptor signatures, originally introduced under the concept of scriptless scripts, extend digital signature schemes by adding a pre-signing phase (see Definition 2.6.2). In this phase, a user generates a pre-signature that can later be converted into a valid signature once a witness y for a statement Y in an NP language $\mathcal{L}$ becomes available. A canonical application is the atomic swap, which enables two mutually distrustful parties to securely exchange digital assets without relying on a trusted escrow intermediary.

Security properties ensuring the suitability of adaptor signatures for atomic swaps have been formalized through notions such as adaptive existential unforgeability, pre-signature adaptability, and witness extractability, that we saw in Section 2.6.

Recently adaptor signatures have found applications across numerous blockchain protocols, including payment [56] and generalized channels [4], deterministic wallets [22, 25], private coin-mixing systems [65, 31], oracle-based payments [43], and multiparty fair exchange mechanisms [6]. Subsequent research [21, 29, 30] demonstrated that the original security framework does not provide sufficient guarantees for many of these broader applications. This motivated the introduction of stronger security notions, including extractability, unique extractability, unlinkability, and pre-verify soundness, in order to support these emerging use cases.

Despite these advances in security definitions, most existing adaptor signature constructions face a significant long-term challenge: vulnerability to quantum attacks.

Cryptographic group actions have emerged as a promising foundation for post-quantum digital signatures. A major advantage of this approach is that it often permits relatively straightforward extensions to more advanced cryptographic functionalities, such as ring signatures, threshold signatures, and blind signatures.

However, only two adaptor signature constructions based on group actions currently exist in the literature: Isogeny Adaptor Signatures (IAS) [66] and adaptor-modified CSI-FiSh [11]. Both approaches exhibit important drawbacks. IAS relies on a large and computationally expensive non-interactive zero-knowledge proof, causing pre-signatures to be substantially larger than standard signatures and significantly more expensive to generate. Meanwhile, adaptor-modified CSI-FiSh produces signatures in a format that differs slightly from standard CSI-FiSh signatures, preventing them from passing the standard CSI-FiSh verification procedure. As a consequence, the scheme can only be deployed on blockchains that explicitly support both signature formats.

To fill this gap, with our construction One-Round Cheating Adaptor Signatures (ORCAS) [12]

we create an efficient group action-based adaptor signature without modification to the underlying signature verification algorithm.

6.1 Sigma Protocols from Group Actions and CSI-FiSh

We introduced the concept of hard relation in Definition 2.6.1.

If we have a group G acting on a set X with an action

$$* : G \times X \rightarrow X,$$

we consider relations $\mathcal{R}$ of the following form. For any fixed $x_0 \in X$,

$$\mathcal{R}_{G,x_0} = \{(Y, y) \in X \times G \mid Y = y * x_0 \text{ for } y \in G\}.$$

The corresponding language $\mathcal{L}$ is

$$\mathcal{L}_{G,x_0} = \{Y \in X \mid \exists y \in G \text{ s.t. } Y = y * x_0\}.$$

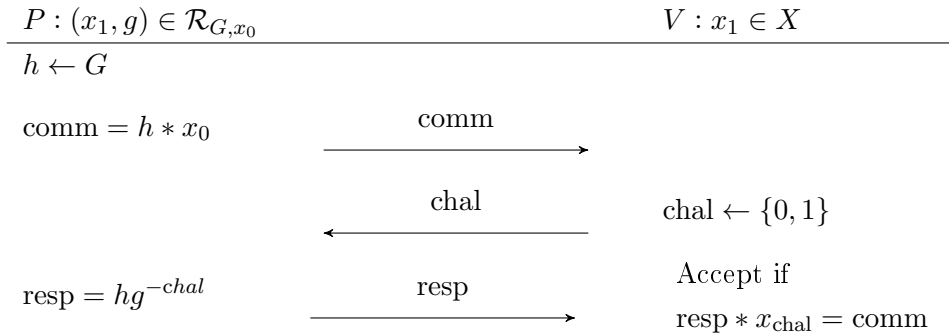
If we require that G admits a uniform sampling algorithm, we can derive a sampling algorithm for the relation $\mathcal{R}_{G,x_0}$:

$$\text{sample } g \leftarrow G \text{ and set } Y = g * x_0.$$

If $\mathcal{R}_{G,x_0}$ is equipped with the previous sampling algorithm, then $\mathcal{R}_{G,x_0}$ is a hard relation under the group action inverse assumption for G and X . If $x_0 \in X$, we denote with $G * x_0$ the orbit of X under the action $*$ given by G .

Definition 6.1.1 (Group Action Inverse Problem, Group Action Inverse assumption). Let G be a group that acts on a set X , and let $x_0 \in X$. The group action inverse problem (GAIP) asks, given $x_1 \in G * x_0$ to find $g \in G$ such that $x_1 = g * x_0$. The group action inverse assumption is that there exists no probabilistic polynomial-time adversary $\mathcal{A}$ that solves the GAIP with non-negligible probability.

Remark 6.1.2. Any cryptographic group action yields an ID scheme between a prover P and a verifier V for the relation $\mathcal{R}_{G,x_0}$.



One can use multiple public keys to reduce the soundness error of an identification scheme. This idea leads to the multi-target GAIP.

Definition 6.1.3 (Multi-target Group Action Inverse Problem/assumption). Let G be a group that acts on a set X , and let $x_0 \in X$. The multi-target Group Action Inverse Problem (MT-GAIP) asks, given $x_1, \dots, x_t \in G * x_0$ to find $g \in G$ such that $x_i = g * x_j$ for some $i, j \in \{0, \dots, t\}$ with $i \neq j$. The multi-target group action assumption is that there exists non probabilistic polynomial-time adversary $\mathcal{A}$ that solves the MT-GAIP with non-negligible probability.

From the ID scheme of Remark 6.1.2 we can derive the basic description of the digital signature CSI-FiSh.

Let $S - 1$ be the number of public keys, t the number of rounds and

$$\mathcal{H} : \{0, 1\}^* \rightarrow \{-(S - 1), -(S - 2), \dots, 0, \dots, S - 1\}^t$$

a hash function (modeled as if it was truly random, i.e modeled as a random oracle) with output distribution $\mathcal{D}_{\mathcal{H}}$ uniform over $\{-(S - 1), -(S - 2), \dots, 0, \dots, S - 1\}^t$.

We consider the transitive action of a group G on a set X with starting element $x_0 \in X$, for which the GAIP is hard.

We denote a public key by $\text{pk} = (x_1, \dots, x_{S-1}) \in X^{S-1}$ with corresponding secret key $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1}) \in G^{S-1}$ such that $x_i = \text{sk}_i * x_0$ for all $i = 1, \dots, S - 1$. According to our notation, we set $\text{sk}_0 = e_G$ the neutral element of the group G , so that $\text{sk}_0 * x_0 = x_0$.

Extrapolating from the particular group action the CSI-FiSh signature scheme can be described as follows.

KeyGen: for all $i = 1, \dots, S - 1$, according to the security parameter 1^λ the key generation algorithm samples $\text{sk}_i \leftarrow G$ and puts $x_i = \text{sk}_i * x_0$. Then $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$ and $\text{pk} = (x_1, \dots, x_{S-1})$. It returns (sk, pk) .

Sign: with a secret key $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$ and a message m , the signing algorithm samples $g_1, \dots, g_t \leftarrow G$ uniformly from G and computes

$$\text{comm}_i = g_i * x_0 \text{ for all } i = 1, \dots, t.$$

Then, via the hash function $\mathcal{H}$ it computes

$$\text{chal} = \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t) \in \{-(S - 1), -(S - 2), \dots, 0, \dots, S - 1\}^t,$$

where $\cdot || \cdot$ denotes concatenation (as strings). Then it parses chal as $(\text{chal}_1, \dots, \text{chal}_t)$. For $i = 1, \dots, t$ the responses are computed as

$$\text{resp}_i = g_i \cdot \text{sk}_{\text{chal}_i}^{-1}.$$

The signing algorithm returns the signature

$$\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t).$$

Verify: given a public key $\text{pk} = (x_1, \dots, x_{S-1})$, the message m and the signature $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$, it parses chal as $(\text{chal}_1, \dots, \text{chal}_t)$. Then it computes

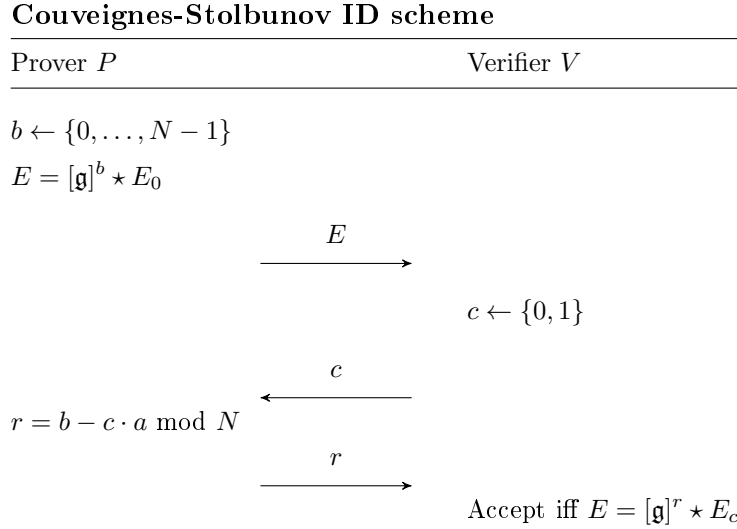
$$\text{comm}'_i = \text{resp}_i * x_{\text{chal}_i} \text{ for all } i = 1, \dots, t.$$

Finally, it computes $\text{chal}' = \mathcal{H}(m || \text{comm}'_1 || \dots || \text{comm}'_t)$. If $\text{chal}' = \text{chal}$ the signature is accepted, otherwise it is rejected.

The action used in CSI-FiSh is the one of CSIDH presented in Section 4.7: according to Theorem 4.7.1, if $\mathcal{O}$ is an order in an imaginary quadratic field such that $\mathcal{E}\ell\ell_{\mathcal{O}}(\mathbb{F}_p)$ is non-empty, then the ideal class group $\text{cl}(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}\ell\ell_{\mathcal{O}}(\mathbb{F}_p)$. So $G = \text{cl}(\mathcal{O})$ and $X = \mathcal{E}\ell\ell_{\mathcal{O}}(\mathbb{F}_p)$.

From our description of the backbone of CSI-FiSh we can see that it is realized as a Fiat-Shamir transformation of an identification scheme, developed independently by Couveignes [17] and Stolbunov [63]. Thus we call it Couveignes-Stolbunov ID scheme.

The identification scheme works as follows. The public key of the prover P consists of $E_1 = [\mathbf{a}] \star E_0$ with $[\mathbf{a}] \in \text{cl}(\mathcal{O})$ a random element and E_0 the basic elliptic curve specified by the system parameters. If we assume that $\text{cl}(\mathcal{O})$ is cyclic with generator $[\mathbf{g}]$, we can write $[\mathbf{a}] = [\mathbf{g}]^a$ with $a \in \{0, \dots, N-1\}$ and $N = |\text{cl}(\mathcal{O})|$. The prover P samples a random element $b \leftarrow \{0, \dots, N-1\}$, computes $[\mathbf{b}] = [\mathbf{g}]^b$ and commits to the (isomorphism class of the) curve $E = [\mathbf{g}]^b \star E_0$. Then, the verifier V chooses a random bit $c \in \{0, 1\}$ and sends it to the prover P. If $c = 0$, the prover P responds with $r = b$ and the verifier V checks that $E = [\mathbf{g}]^r \star E_0$. If $c = 1$, the prover P responds with $r = b - a \bmod N \in \{0, \dots, N-1\}$ and the verifier V checks that $E = [\mathbf{g}]^r \star E_1$.



Theorem 6.1.4. [11, Theorem 4] *The Couveignes-Stolbunov ID scheme is complete and secure, proving knowledge of a solution of a GAIP instance.*

In order to optimize the Couveignes-Stolbunov ID scheme we can do many things. First, it is sufficient that the prover P sends $\mathcal{H}(E)$ rather than E for a collision-resistant hash function $\mathcal{H}$. The thing that the verifier V has to do is to compute $\mathcal{H}([\mathbf{g}]^r \star E_c)$ and check that it is equal to the hash value sent by the prover. If we do t rounds of the protocol in parallel to amplify soundness, we send a single hash of the concatenations of all the $E^{(i)}$ s.

In order to lower the soundness error, one can increase the challenge space. In our case, we move from GAIP to MT-GAIP. We use $S-1$ public keys instead of one and the public key now consists of the tuple

$$(E_0, E_1 = [\mathbf{g}]^{a_1} \star E_0, \dots, E_{S-1} = [\mathbf{g}]^{a_{S-1}} \star E_0).$$

The prover P proves to the verifier V that he knows $s \in \{0, \dots, N-1\}$ such that $E_j = [\mathbf{g}]^s \star E_i$ for a pair of curves in the public key with $i \neq j$. The prover P chooses a random exponent $b \leftarrow \{0, \dots, N-1\}$ and computes $E^{(i)} = [\mathbf{g}]^b \star E_0$.

The verifier V sends a challenge $c \in \{0, \dots, S-1\}$ and the response consists of $r = b - a_c \bmod N$ (with $a_0 = 0$). The verifier then recomputes $[\mathbf{g}]^r \star E_c$ and verifies that is equal to $E^{(i)}$.

Proposition 6.1.5. [11, Theorem 5] *The adapted version of the Couveignes-Stolbunov ID scheme is complete and secure, proving knowledge of a solution of a MT-GAIP instance.*

Now, the soundness error of the adapted Couveignes-Stolbunov ID scheme is $\frac{1}{S}$

In CSI-FiSh the authors make another modification in order to decrease again the soundness error. They exploit the fact that, given a curve $E = [\mathfrak{g}]^a \star E_0$, its quadratic twist E^{qt} can be computed efficiently and is isomorphic over $\mathbb{F}_p$ to $[\mathfrak{g}]^{-a} \star E_0$. So, the set of public keys is almost doubled to

$$E_{-(S-1)}, \dots, E_0, \dots, E_{S-1},$$

where $E_{-i} = E_i^{qt}$. Here the communication cost is not augmented, and the soundness error becomes $\frac{1}{2S-1}$. The inclusion of this modification does not affect security: solving this "twisted" MT-GAIP, can be reduced to solving GAIP.

So, with all the new modification to the Couveignes-Stolbunov ID scheme, the authors of CSI-FiSh transform the ID scheme into a digital signature using the Fiat-Shamir paradigm: the challenges $c_i \in \{-(S-1), \dots, S-1\}$ are obtained by hashing the message m together with the curves $E^{(i)}$, $i = 1, \dots, t$:

$$(c_1, \dots, c_t) = \mathcal{H}(m \| E^{(1)} \| \dots \| E^{(t)}).$$

Therefore, the signature consists of $(r_1, \dots, r_t, c_1, \dots, c_t)$; the verifier V recomputes $E^{(i)} = [\mathfrak{g}]^{r_i} \star E_{c_i}$ for $i = 1, \dots, t$ and checks that $(c_1, \dots, c_t)$ and $\mathcal{H}(m \| E^{(1)} \| \dots \| E^{(t)})$ coincide.

KeyGen(1^λ)	Sign(sk, m)
101 : for $i = 1, \dots, S-1$ do	201 : parse $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$
102 : $\text{sk}_i \leftarrow \{0, \dots, N-1\}$	202 : $b_1, \dots, b_t \leftarrow \{0, \dots, N-1\}$
103 : $E_i = [\mathfrak{g}]^{\text{sk}_i} \star E_0$	203 : for $i = 1, \dots, t$ do
104 : endfor	204 : $\text{comm}_i = [\mathfrak{g}]^{b_i} \star E_0$
105 : $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$	205 : endfor
106 : $\text{pk} = (E_1, \dots, E_{S-1})$	206 : $\text{chal} = \mathcal{H}(m \ \text{comm}_1 \ \dots \ \text{comm}_t)$
107 : return (sk, pk)	207 : parse $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$
Verify(pk, m , σ)	208 : for $i = 1, \dots, t$ do
301 : parse $\text{pk} = (E_1, \dots, E_{S-1})$	209 : $\text{resp}_i = b_i - \text{sk}_{\text{chal}_i} \bmod N (\in \{0, \dots, N-1\})$
302 : parse $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$	210 : endfor
303 : parse $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$	211 : return $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$
304 : for $i = 1, \dots, t$ do	
305 : $\text{comm}'_i = [\mathfrak{g}]^{\text{resp}_i} \star E_{\text{chal}_i}$	
306 : endfor	
307 : $\text{chal}' = \mathcal{H}(m \ \text{comm}'_1 \ \dots \ \text{comm}'_t)$	
308 : Accept iff $\text{chal}' = \text{chal}$	

Figure 6.1: CSI-FiSh digital signature scheme, without twists

The hash function $\mathcal{H}$ maps into $\{0, \dots, S-1\}^t$

From CSI-FiSh, we built ORCAS, i.e. One Round "Cheating" Adaptor Signatures, that inherits its name from the fact that its pre-signing algorithm resembles how a cheating signer might try to forge a signature in the Fiat-Shamir paradigm, but with a caveat that renders pre-signing computationally feasible.

In the following we will provide the description of the base algorithm, which we call Base ORCAS.

6.2 ORCAS

In Section 2.6 we gave all the adaptor signature definitions. In this one we explain how the (base) ORCAS scheme is constructed.

As we saw in Section 2.6, an adaptor signature scheme with respect to a digital signature scheme $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify})$ and an hard relation $\mathcal{R}$ with language $\mathcal{L}$ consists of four additional algorithms: PreSign, PreVerify, Adapt and Ext. We consider CSI-FiSh as our base digital signature scheme on which we build the adaptor signature.

Here's a description of the basic version of ORCAS. For the notation we keep the action of the group G on the set X (instead of directly writing the action of the ideal class group on the elliptic curves).

First, we have $x_0 \in X$ and $y \leftarrow G$. Then $Y = y * x_0$; Y becomes the statement for the witness y .

PreSign: it is a probabilistic polynomial-time algorithm that on input the secret key $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$ the message m and the statement $Y \in \mathcal{L}_{G, x_0}$ performs the following steps. It selects $r \leftarrow \{1, \dots, t\}$ and $g_1, \dots, g_t \leftarrow G$. For all $i = 1, \dots, t$

$$\text{comm}_i = \begin{cases} g_i * x_0 & \text{if } i \neq r \\ g_i * Y & \text{if } i = r. \end{cases}$$

It computes

$$\text{chal} = (\text{chal}_1, \dots, \text{chal}_t) = \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t).$$

If $\text{chal}_r \neq 0$ then it restarts form sampling r , otherwise, if $\text{chal}_r = 0$ it passes to the next step.

For all $i = 1, \dots, t$ it puts

$$\widehat{\text{resp}}_i = g_i \cdot \text{sk}_{\text{chal}_i}^{-1}.$$

It outputs $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$.

PreVerify: it is a probabilistic polynomial-time algorithm that with inputs the public key $\text{pk} = (x_1, \dots, x_{S-1})$, the message m , the statement Y and the pre-signature $\hat{\sigma}$, it performs the following steps. If Y is not in the language $\mathcal{L}$ it fails. It parses the pre-signature $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$ and the challenge $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$. For all $i = 1, \dots, t$ it puts

$$\text{comm}'_i = \begin{cases} \widehat{\text{resp}}_i * x_{\text{chal}_i} & \text{if } i \neq r \\ \widehat{\text{resp}}_i * Y & \text{if } i = r. \end{cases}$$

Then it computes $\text{chal}' = \mathcal{H}(m || \text{comm}'_1 || \dots || \text{comm}'_t)$. The pre-signature is accepted if and only if $\text{chal}' = \text{chal}$ and $\text{chal}_r = 0$.

Adapt: it is a probabilistic polynomial-time algorithm that on input a pre-signature $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$ and the witness y for the statement Y it performs the following. It parses $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$.

For $i = 1, \dots, t$ it puts

$$\text{resp}_i = \begin{cases} \widehat{\text{resp}}_i & \text{if } i \neq r \\ \widehat{\text{resp}}_i \cdot y & \text{if } i = r. \end{cases}$$

It returns the adapted signature $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$.

Ext: it is a probabilistic polynomial-time algorithm that with inputs the signature $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$, the pre-signature $(r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$ and the statement Y , performs the following. If $\text{chal} \neq \text{chal}'$ it fails. If it does not fail, it returns $y = \widehat{\text{resp}}_r^{-1} \cdot \text{resp}_r$.

The key idea of our construction is to wait for a challenge of 0 to appear honestly. We guess a round r for which we hope to have $\text{chal}_r = 0$ and construct our commitments as $\text{comm}_i = g_i * x_0$ for each $i \neq r$ and $\text{comm}_r = g_r * Y$. The challenges are, then, constructed using the Fiat-Shamir paradigm:

$$(\text{chal}_1, \dots, \text{chal}_t) = \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t).$$

Then, if $\text{chal}_r = 0$ we construct the responses $(\widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$. The resulting pre-signature

$$\hat{\sigma} = (r, \text{comm}_1, \dots, \text{comm}_t, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$$

can later be adapted by setting $\text{resp}_r = \widehat{\text{resp}}_r \cdot y$. If, instead, $\text{chal}_r \neq 0$ we generate new fresh commitments and try again. This explains why the "C" in ORCAS stands for "Cheating": the procedure described before resembles the way that an adversary might try to forge a signature in the Fiat-Shamir construction, i.e. guess the challenge randomly, construct a compatible commitment and response and hope that the hash function (modeled as a random oracle) will output the guessed challenge. The key difference is that while a forger has to guess the challenge in every position (and he can do it only with probability $\frac{1}{(2S-1)^t}$), we only have to guess the position of a single 0; for that reason, after $2S - 1$ tries we expect to successfully produce an adaptable pre-signature.

In Figure 6.2 we present the pseudo-code for Base ORCAS adapting CSI-FiSh. We recall that $\text{sk}_0 = 0$.

We have that, given $E_0 \in \mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ and $[\mathbf{g}]$ a generator of $\text{cl}(\mathcal{O})$, if we take $y \in \{0, \dots, N - 1\}$ (with $N = \text{cl}(\mathcal{O})$), a statement Y for the witness y is $Y = [\mathbf{g}]^y \star E_0$.

In our paper [12, Theorem 4.16] we prove the following theorem, which ensures that our adaptor signature base ORCAS has all the needed properties in order to be a good adaptor signature.

Theorem 6.2.1. *Suppose that MT-GAIP is hard for the action of the group G on the set X (for example the action of $G = \text{cl}(\mathcal{O})$ on $X = \mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$). Base ORCAS is a correct (Definition 2.6.2), pre-signature adaptable (Definition 2.6.4), strongly full extractable (Definition 2.6.10), uniquely extractable (Definition 2.6.12), unlinkable (Definition 2.6.5) and pre-verify sound (Definitions 2.6.13 and 2.6.14) adaptor signature for CSI-FiSh in the random oracle model.*

For exposition reasons, we will prove all the above properties of Base ORCAS for a modified version of CSI-FiSh, called Filtered CSI-FiSh, in Theorem 6.3.8. The two signature schemes have identical key generation and verification. The only difference is in their signing algorithms. Even the signing algorithms are not too different: every Filtered CSI-FiSh signature is also CSI-FiSh signature. In fact, signing in Filtered CSI-FiSh is equivalent to generating a sequence of CSI-FiSh signatures and choosing one to output with probability proportional to the number of zeros in its challenge. We refer to this process as filtering.

In our paper we also provide an optimization of Base ORCAS, called Fast ORCAS. The most time-consuming part of CSI-FiSh signing consists in computing the isogenies to find the commitment curves. In Base ORCAS the pre-signing algorithm must recompute in succession until it finds a challenge with $\text{chal}_r = 0$. So the expected runtime of PreSign is approximately $2S - 1$ times that of Sign and this is impractical. In fact, we do not need to recompute all the commitments from scratch: if we have generated one commitment vector we can re-randomize by permuting the commitments until we get a favorable challenge. This is exactly what Fast ORCAS does.

Again, this new adaptor signature has all the good properties.

Theorem 6.2.2. [12, Theorem 5.5] *Suppose that MT-GAIP is hard for the action of the group G on the set X . Fast ORCAS is a correct, pre-signature adaptable, strongly full extractable, uniquely extractable, unlinkable and pre-verify sound adaptor signature for CSI-FiSh in the random oracle model.*

PreSign(sk, m, Y)	PreVerify(pk, m, Y, $\hat{\sigma}$)
101 : parse sk = (sk ₁ , ..., sk _{S-1})	201 : if Y $\notin \mathcal{L}_{\{0, \dots, N-1\}, E_0}$
102 : chal _r = 1	202 : return 0
103 : while chal _r $\neq$ 0 do	203 : endif
104 : r $\leftarrow$ {1, ..., t}	204 : parse pk = (E ₁ , ..., E _{S-1})
105 : c ₁ , ..., c _t $\leftarrow$ {0, ..., N-1}	205 : parse $\hat{\sigma}$ = (r, chal, $\widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t$)
106 : for i = 1, ..., t do	206 : parse chal = (chal ₁ , ..., chal _t)
107 : if i $\neq$ r then	207 : for i = 1, ..., t do
108 : comm _i = [g] ^{c_i} $\star$ E ₀	208 : if i $\neq$ r then
109 : elseif i = r then	209 : comm' _i = [g] ^{$\widehat{\text{resp}}_i$} $\star$ E _{chal_i}
110 : comm _i = [g] ^{c_i} $\star$ Y	210 : elseif i = r then
111 : endif	211 : comm' _i = [g] ^{$\widehat{\text{resp}}_i$} $\star$ Y
112 : endfor	212 : endif
113 : chal = $\mathcal{H}(m \text{comm}_1 \dots \text{comm}_t)$	213 : endfor
114 : parse chal = (chal ₁ , ..., chal _t)	214 : chal' = $\mathcal{H}(m \text{comm}'_1 \dots \text{comm}'_t)$
115 : endwhile	215 : Accept iff chal' = chal and chal _r = 0
116 : for i = 1, ..., t do	
117 : $\widehat{\text{resp}}_i = c_i - \text{sk}_{\text{chal}_i} \bmod N$	
118 : return $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$	
Ext($\sigma, \hat{\sigma}, Y$)	Adapt($\hat{\sigma}, y$)
301 : parse $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$	401 : parse $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$
302 : parse $\hat{\sigma} = (r, \text{chal}', \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$	402 : parse chal = (chal ₁ , ..., chal _t)
303 : if chal $\neq$ chal' then	403 : for i = 1, ..., t do
304 : return $\perp$	404 : if i $\neq$ r then
305 : endif	405 : resp _i = $\widehat{\text{resp}}_i$
306 : return y = resp _r - $\widehat{\text{resp}}_r \bmod N$	406 : elseif i = r then
	407 : resp _i = $\widehat{\text{resp}}_i + y \bmod N$
	408 : endif
	409 : endfor
	410 : return $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$

Figure 6.2: Base ORCAS adapting CSI-FiSh (without twists)

6.3 Properties of Base ORCAS for Filtered CSI-FiSh

As we mentioned in the previous section, we prove the Theorem 6.2.1 for Base ORCAS for Filtered CSI-FiSh, which is described in Figure 6.3.

KeyGen(1^λ)	Sign(sk, m)
101 : for $i = 1, \dots, S-1$ do	201 : parse $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$
102 : $\text{sk}_i \leftarrow \{0, \dots, N-1\}$	202 : $b_1, \dots, b_t \leftarrow \{0, \dots, N-1\}$
103 : $E_i = [\mathbf{g}]^{\text{sk}_i} \star E_0$	203 : for $i = 1, \dots, t$ do
104 : endfor	204 : $\text{comm}_i = [\mathbf{g}]^{b_i} \star E_0$
105 : $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$	205 : endfor
106 : $\text{pk} = (E_1, \dots, E_{S-1})$	206 : $\text{chal} = \mathcal{H}(m \text{comm}_1 \dots \text{comm}_t)$
107 : return (sk, pk)	207 : parse $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$
	208 : $r \leftarrow \{1, \dots, t\}$
Verify(pk, m, σ)	209 : if $\text{chal}_r \neq 0$ then
301 : parse $\text{pk} = (E_1, \dots, E_{S-1})$	210 : goto line 202
302 : parse $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$	211 : endif
303 : parse $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$	212 : for $i = 1, \dots, t$ do
304 : for $i = 1, \dots, t$ do	213 : $\text{resp}_i = b_i - \text{sk}_{\text{chal}_i} \bmod N (\in \{0, \dots, N-1\})$
305 : $\text{comm}'_i = [\mathbf{g}]^{\text{resp}_i} \star E_{\text{chal}_i}$	214 : endfor
306 : endfor	215 : return $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$
307 : $\text{chal}' = \mathcal{H}(m \text{comm}'_1 \dots \text{comm}'_t)$	
308 : Accept iff $\text{chal}' = \text{chal}$	

Figure 6.3: Filtered CSI-FiSh digital signature scheme, without twists

The lines in grey do not appear in regular CSI-FiSh

The hash function $\mathcal{H}$ maps into $\{0, \dots, S-1\}^t$

We consider Filtered CSI-FiSh instead of standard CSI-FiSh because our adaptor signature scheme has (strongly) canonical signing with reference to this scheme: we can generate a "correctly-distributed" signature by adapting a pre-signature generated with respect to any statement-witness pair (even one chosen by an adversary). This property greatly simplifies the remaining proofs.

Throughout this section let KeyGen, Sign, and Verify denote the components of Filtered CSI-FiSh.

Lemma 6.3.1 (Canonical Signing). *Base ORCAS has strongly canonical signing for Filtered CSI-FiSh.*

Proof. Let $(Y, y) \in \mathcal{R}$ and $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$, and fix a message m . A signature $\sigma \leftarrow \text{Sign}(\text{sk}, m)$ is uniquely determined by the tuple of commitments $(\text{comm}_1, \dots, \text{comm}_t)$, which is distributed uniformly over $C_1 \sqcup \dots \sqcup C_t$ (the disjoint union means we count multiplicity), where $C_r = \{(\text{comm}_1, \dots, \text{comm}_t) \in X^t : \text{chal}_r = 0\}$. In fact $(\text{comm}_1, \dots, \text{comm}_t)$ is sampled uniformly from X^t . It may, however, be rejected depending on the value of $\text{chal} = \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t)$: a commitment is used if and only if $\text{chal}_r = 0$, where $r \leftarrow \{1, \dots, t\}$. Thus it is clear that the final commitment is sampled uniformly from $C_1 \sqcup \dots \sqcup C_t$. A pre-signature $\hat{\sigma} \leftarrow \text{getsPreSign}(\text{sk}, m, Y)$ is uniquely determined by its own tuple of commitments $(\text{comm}'_1, \dots, \text{comm}'_t)$, along with its choice of r . Since two pre-signatures with the same commitments but different choices of r will both adapt to the same signature, the output of $\text{CanonicalSign}_{(Y, y)}$ is uniquely determined by $(\text{comm}'_1, \dots, \text{comm}'_t)$. Therefore, it suffices to show that $(\text{comm}'_1, \dots, \text{comm}'_t)$ is uniform over $C_1 \sqcup \dots \sqcup C_t$, which follows from the previous reasoning done for $(\text{comm}_1, \dots, \text{comm}_t)$. $\square$

Lemma 6.3.2 (Pre-Signature Correctness). *Base ORCAS is pre-signature correct for Filtered CSI-FiSh.*

Proof. For pre-signature correctness we will show three properties:

1. For a given key pair (sk, pk) , message m , tuple $(Y, y) \in \mathcal{R}$, and pre-signature $\widehat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$, we have $\text{PreVerify}(\text{pk}, m, Y, \widehat{\sigma}) = 1$.
2. After adapting the pre-signature $\widehat{\sigma}$ into the signature $\sigma = \text{Adapt}(\widehat{\sigma}, y)$, we have $\text{Verify}(\text{pk}, m, \sigma) = 1$.
3. After extracting $y' = \text{Ext}(\sigma, \widehat{\sigma}, Y)$, we have $(Y, y') \in \mathcal{R}$.

Let $\lambda \in \mathbb{N}$, $m \in \{0, 1\}^*$ and $(Y, y) \in \mathcal{R}$. Given $\text{sk}_i \in \{0, \dots, N-1\}$ for $i = 1, \dots, S-1$, then we compute $E_i = [\mathbf{g}]^{\text{sk}_i} \star E_0$, for all $i = 1, \dots, S-1$. Consider keys $\text{sk} = (\text{sk}_1, \dots, \text{sk}_{S-1})$ and $\text{pk} = (E_1, \dots, E_{S-1})$, where each $E_i = [\mathbf{g}]^{\text{sk}_i} \star E_0$. To show property 1, let $\widehat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t) \leftarrow \text{PreSign}(\text{sk}, m, Y)$. Notice that asking the couple (Y, y) is in $\mathcal{R}$ implies in particular that Y is in $\mathcal{L}$. Then the first PreVerify check is satisfied. Moreover, for each $i \neq r$, the pre-signer computes $\text{comm}_i = [\mathbf{g}]^{c_i} \star E_0$ and $\widehat{\text{resp}}_i = c_i - \text{sk}_{\text{chal}_i} \bmod N$, while the pre-verifier computes $\text{comm}'_i = [\mathbf{g}]^{\widehat{\text{resp}}_i} \star E_{\text{chal}_i}$. So, $\text{comm}'_i = ([\mathbf{g}]^{c_i - \text{sk}_{\text{chal}_i}} \star [\mathbf{g}]^{\text{sk}_{\text{chal}_i}}) \star E_0 = [\mathbf{g}]^{c_i} \star E_0 = \text{comm}_i$. For $i = r$, the pre-signer computes $\text{comm}_r = [\mathbf{g}]^{c_r} \star Y$ and $\widehat{\text{resp}}_r = c_r - \text{sk}_{\text{chal}_r} \bmod N$, while the pre-verifier computes $\text{comm}'_r = [\mathbf{g}]^{\widehat{\text{resp}}_r} \star Y = [\mathbf{g}]^{c_r} \star Y = \text{comm}_r$. Having shown that $\text{comm}'_i = \text{comm}_i$ for each $i \in 1, \dots, t$, we know

$$\text{chal}' = \mathcal{H}(m || \text{comm}'_1 || \dots || \text{comm}'_t) = \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t) = \text{chal}.$$

The honesty of the pre-signature $\widehat{\sigma}$ guarantees us that $\text{chal}_r = 0$. Therefore $\text{PreVerify}(\text{pk}, m, Y, \widehat{\sigma}) = 1$. To show property 2, let $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t) = \text{Adapt}(\widehat{\sigma}, y)$. For each $i \neq r$, the adaptor computes $\text{resp}_i = \widehat{\text{resp}}_i$ while the verifier computes $\text{comm}'_i = [\mathbf{g}]^{\text{resp}_i} \star E_{\text{chal}_i}$. Once again, we have $\text{comm}'_i = \text{comm}_i$. For $i = r$, the adaptor computes $\text{resp}_r = \widehat{\text{resp}}_r + y \bmod N$ while the verifier computes $\text{comm}_r = [\mathbf{g}]^{\text{resp}_r} \star E_{\text{chal}_r}$. Since $\text{chal}_r = 0$, we have $\text{comm}'_r = [\mathbf{g}]^{\widehat{\text{resp}}_r + y} \star E_0 = [\mathbf{g}]^{\widehat{\text{resp}}_r} \star Y = \text{comm}_r$. Again,

$$\text{chal}' = \mathcal{H}(m || \text{comm}'_1 || \dots || \text{comm}'_t) = \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t) = \text{chal}.$$

This is the verifier's only check, so $\text{Verify}(\text{pk}, m, \sigma) = 1$. To show property 3, let $y' = \text{Ext}(\sigma, \widehat{\sigma}, Y)$. Since σ and $\widehat{\sigma}$ have the same challenge chal , the extractor will not return $\perp$; instead, it will return

$$y' = \text{resp}_r - \widehat{\text{resp}}_r = \text{resp}_r - (\text{resp}_r - y)y.$$

Therefore $(Y, y') = (Y, y) \in \mathcal{R}$. □

Lemma 6.3.3 (Pre-Signature Adaptability). *Base ORCAS is pre-signature adaptable for Filtered CSI-FiSh.*

Proof. Let $(Y, y) \in \mathcal{R}$, $m \in \{0, 1\}^*$, and $\text{pk} = (E_1, \dots, E_{S-1})$. Consider a pre-signature $\widehat{\sigma} = (r, \text{chal}, \text{resp}_1, \dots, \text{resp}_t)$ such that $\text{PreVerify}(\text{pk}, m, Y, \widehat{\sigma}) = 1$, and let $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t) \leftarrow \text{Adapt}(\widehat{\sigma}, y)$. For all $i \neq r$, since $\text{resp}_i = \widehat{\text{resp}}_i$ we have the same comm'_i in both PreVerify and Verify . For $i = r$, we know from pre-verification that $\text{chal}_r = 0$ and thus $[\mathbf{g}]^{\text{resp}_r} \star E_{\text{chal}_r} = (\widehat{\text{resp}}_r + y) \star E_0 = [\mathbf{g}]^{\widehat{\text{resp}}_r} \star Y$. So comm'_r is also the same in both PreVerify and Verify , meaning so is chal' and thus $\text{Verify}(\text{pk}, m, \sigma) = 1$. □

Lemma 6.3.4. *Assuming $\mathcal{H}$ is a collision-resistant hash-function, Base ORCAS satisfies unique extractability for Filtered CSI-FiSh.*

Proof. Suppose $\mathcal{A}$ outputs a tuple $(m, Y, \widehat{\sigma}, \sigma, \sigma')$ to win the UniqueExtractability game. Expand $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$, $\sigma' = (\text{chal}', \text{resp}'_1, \dots, \text{resp}'_t)$, and $\widehat{\sigma} = (r, \text{chal}'', \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$. For each $i \in$

$\{1, \dots, t\}$, define $\text{comm}_i = [\mathbf{g}]^{\text{resp}_i} \star E_{\text{chal}_i}$ and $\text{comm}'_i = [\mathbf{g}]^{\text{resp}'_i} \star E_{\text{chal}'_i}$. Since both extractions succeed and Ext checks that the challenges match, we have $\text{chal} = \text{chal}' = \text{chal}''$. Moreover $\text{chal} = \text{chal}'$ and $\sigma \neq \sigma'$ imply that there exists i such that $\text{resp}_i \neq \text{resp}'_i$, thus $\text{comm}_i \neq \text{comm}'_i$. Since $\text{Verify}(\text{pk}, m, \sigma) = \text{Verify}(\text{pk}, m, \sigma') = 1$, we have found a hash collision:

$$\mathcal{H}(m \parallel \text{comm}_1 \parallel \dots \parallel \text{comm}_t) = \text{chal} = \text{chal}' = \mathcal{H}(m \parallel \text{comm}'_1 \parallel \dots \parallel \text{comm}'_t).$$

If $\mathcal{H}$ is collision resistant, then $\mathcal{A}$ can output a winning tuple $(m, Y, \hat{\sigma}, \sigma, \sigma')$ only with negligible probability. Therefore Base ORCAS satisfies unique extractability for Filtered CSI-FiSh. $\square$

Lemma 6.3.5. *If $\mathcal{H}$ is a collision-resistant hash-function, Base ORCAS satisfies strong full extractability for Filtered CSI-FiSh in the random oracle model.*

Proof. We will show that the scheme satisfies extractability. Since Base ORCAS has canonical signing for Filtered CSI-FiSh, $\mathcal{R}$ is a hard relation (under the GAIP assumption), and (by Lemma 6.3.4) Base ORCAS satisfies unique extractability, by [21, Theorem 3.4] this implies strong full extractability.

We will construct an algorithm $\mathcal{B}$ to attack MT-GAIP by simulating the security game

$$\text{Extractability}_{\mathcal{A}, \text{Base ORCAS}}.$$

The simulator acts identically to the honest challenger, except it does not know sk and must instead program the random oracle in order to answer pre-signature queries. When asked to pre-sign message m with respect to statement Y , it first samples a random challenge chal and an index r in $\{1, \dots, t\}$. If $\text{chal}_r \neq 0$, then it repeats the random sampling of chal and r until it gets $\text{chal}_r = 0$. The simulator also samples at random a response $\widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t$. It then computes the (unique) commitment $\text{comm}_1, \dots, \text{comm}_t$ consistent with r , chal and $\widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t$, setting $\text{comm}_r = [\mathbf{g}]^{\text{resp}_r} \star Y$ and $\text{comm}_i = [\mathbf{g}]^{\widehat{\text{resp}}_i} \star E_{\text{chal}_i}$ for each $i \neq r$. Finally, it programs the random oracle by setting $\mathcal{H}[m \parallel \text{comm}_1 \parallel \dots \parallel \text{comm}_t] = \text{chal}$ and outputs the completed pre-signature $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$. Naturally random oracle programming fails if the random oracle has already been queried on $m \parallel \text{comm}_1 \parallel \dots \parallel \text{comm}_t$. Since the commitment is sampled at random from $C_1 \sqcup \dots \sqcup C_t$, where $C_r = \{(\text{comm}_1, \dots, \text{comm}_t) \in X^t : \text{chal}_r = 0\}$, this will happen with negligible probability. Notice that the honest pre-signing algorithm chooses r at random before computing chal , while the simulator samples iteratively at random chal and r until it finds an acceptable pair with $\text{chal}_r = 0$. However, this slight difference does not effect the distribution of the output of pS , since both chal and r are sampled at random from the distribution of $\mathcal{H}(\mathcal{D}_{\mathcal{H}})$ and $\{1, \dots, t\}$, respectively.

$\mathcal{B}(E_1, \dots, E_{S-1})$	$\mathcal{O}_{\text{pS}}(m, Y)$
101 : $T, \mathcal{Q} = \emptyset$	301 : do
102 : parse $\text{pk} = (E_1, \dots, E_{S-1})$	302 : $\text{chal} \leftarrow \mathcal{D}_{\mathcal{H}}$
103 : $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pS}}(\cdot, \cdot)}(\text{pk})$	303 : parse $\text{chal} = (\text{chal}_1, \dots, \text{chal}_t)$
104 : assert $\text{Verify}(\text{pk}, m, \sigma) = 1$	304 : $r \leftarrow \{1, \dots, t\}$
105 : for $(Y, \hat{\sigma}) \in T[m]$	305 : until $\text{chal}_r = 0$
106 : assert $(Y, \text{Ext}(\sigma, \hat{\sigma}, Y)) \notin \mathcal{R}$	306 : $\widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t \leftarrow \{0, \dots, N-1\}$
107 : endfor	307 : for $i = 1, \dots, t$ do
108 : return (m, σ)	308 : if $i \neq r$ then
	309 : $\text{comm}_i = [\mathbf{g}]^{\widehat{\text{resp}}_i} \star E_{\text{chal}_i}$
	310 : elseif $i = r$ then
	311 : $\text{comm}_i = [\mathbf{g}]^{\widehat{\text{resp}}_i} \star Y$
	312 : endif
	313 : endfor
	314 : assert $\mathcal{H}[m \text{comm}_1 \dots \text{comm}_t] \neq \perp$
	315 : $H[m \text{comm}_1 \dots \text{comm}_t] \leftarrow \text{chal}$
	316 : $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$
	317 : $T[m] \leftarrow T[m] \cup \{(Y, \hat{\sigma})\}$
	318 : return $\hat{\sigma}$
$\mathcal{O}_{\mathcal{H}}(x)$	
201 : $\mathcal{Q} = \mathcal{Q} \cup \{x\}$	
202 : if $\mathcal{H}[x] = \perp$ then	
203 : $\mathcal{H}[x] \leftarrow \mathcal{D}_{\mathcal{H}}$	
204 : endif	
205 : return $\mathcal{H}[x]$	

Figure 6.4: Our MT-GAIP algorithm $\mathcal{B}$ built from an adversary $\mathcal{A}$ which attacks the extractability of our scheme.

Recall the security game Extractability as described in Definition 2.6.11. We now make the reduction from Extractability $_{\mathcal{A}, \text{Base ORCAS}}$ to the simulator $\mathcal{B}$.

$\mathbf{G}_0$	$\mathcal{O}_{\mathcal{H}}(x)$
101 : $T, \mathcal{Q} = \emptyset$	201 : $\mathcal{Q} = \mathcal{Q} \cup \{x\}$
102 : $\mathcal{H} \leftarrow [\perp]$	202 : if $\mathcal{H}[x] = \perp$ then
103 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	203 : $\mathcal{H}[x] \leftarrow \mathcal{D}_{\mathcal{H}}$
104 : $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pS}}(\cdot, \cdot)}(\text{pk})$	204 : endif
105 : assert $\text{Verify}(\text{pk}, m, \sigma) = 1$	205 : return $\mathcal{H}[x]$
106 : for $(Y, \hat{\sigma}) \in T[m]$ do	
107 : assert $(Y, \text{Ext}(\sigma, \hat{\sigma}, Y)) \notin \mathcal{R}$	
108 : endfor	
109 : return 1	
	$\mathcal{O}_{\text{pS}}(m, Y)$
	301 : $\hat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
	302 : $T[m] \leftarrow T[m] \cup \{(Y, \hat{\sigma})\}$
	303 : return $\hat{\sigma}$

Figure 6.5: **Game $\mathbf{G}_0$** : This game corresponds to the original Extractability game, in which the adversary $\mathcal{A}$ is given access to a pre-signature oracle $\mathcal{O}_{\text{pS}}$ and an oracle $\mathcal{O}_{\mathcal{H}}$ and, to win, must produce a valid signature (m, σ) such that $(Y, \text{Ext}(\sigma, \hat{\sigma}, Y)) \notin \mathcal{R}$ for any previous queries to $\mathcal{O}_{\text{pS}}$.

Thus $\mathbb{P}[\mathbf{G}_0 = 1] = \mathbb{P}[\text{Extractability}_{\mathcal{A}, \text{Base ORCAS}}(\lambda) = 1]$.

Assuming the pre-signing oracle does not abort, in which case $\text{chal} \leftarrow \mathcal{D}_{\mathcal{H}}$, we claim the pre-signature $\hat{\sigma}$ output by $\mathcal{O}_{\text{pS}}$ is distributed the same in $\mathbf{G}_1$ as in $\mathbf{G}_0$. Fix m and Y and define $p_n(\hat{\sigma}) = \mathbb{P}[\hat{\sigma}' = \hat{\sigma} | \hat{\sigma}' \leftarrow \mathbf{G}_n. \mathcal{O}_{\text{pS}}]$ for $n \in \{0, 1\}$. Since each $\widehat{\text{resp}}_i$ is distributed uniformly over $\{0, \dots, N-1\}$,

$\mathbf{G}_1$	$\mathcal{O}_{\text{pS}}(m, Y)$
101 : $T, \mathcal{Q} = \emptyset$	301 : do
102 : $\mathcal{H} \leftarrow [\perp]$	302 : $\text{chal} \leftarrow \mathcal{D}_{\mathcal{H}}$
103 : $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda)$	303 : $\text{parse chal} = (\text{chal}_1, \dots, \text{chal}_t)$
104 : $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pS}}(\cdot, \cdot)}(\text{pk})$	304 : $r \leftarrow [t]$
105 : assert $\text{Verify}(\text{pk}, m, \sigma) = 1$	305 : until $\text{chal}_r = 0$
106 : for $(Y, \hat{\sigma}) \in T[m]$ do	306 : $\widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t \leftarrow \{0, \dots, N-1\}$
107 : assert $(Y, \text{Ext}(\sigma, \hat{\sigma}, Y)) \notin \mathcal{R}$	307 : for $i = 1, \dots, t$ do
108 : endfor	308 : if $i \neq r$ then
109 : return 1	309 : $\text{comm}_i = [\mathbf{g}]^{\text{resp}_i} \star E_{\text{chal}_i}$
	310 : elseif $i = r$ then
	311 : $\text{comm}_i = [\mathbf{g}]^{\text{resp}_i} \star Y$
$\mathcal{O}_{\mathcal{H}}(x)$	312 : endif
201 : $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{x\}$	313 : endfor
202 : if $\mathcal{H}[x] = \perp$ then	314 : assert $\mathcal{H}[m \text{comm}_1 \dots \text{comm}_t] = \perp$
203 : $\mathcal{H}[x] \leftarrow \mathcal{D}_{\mathcal{H}}$	315 : $\mathcal{H}[m \text{comm}_1 \dots \text{comm}_t] \leftarrow \text{chal}$
204 : endif	316 : $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$
205 : return $\mathcal{H}[x]$	317 : $T[m] \leftarrow T[m] \cup \{(Y, \hat{\sigma})\}$
	318 : return $\hat{\sigma}$

Figure 6.6: **Game $\mathbf{G}_1$** .

for any valid pre-signature $\hat{\sigma} = (r, \text{chal}, \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$ we have $p_0(\hat{\sigma}) = p_1(\hat{\sigma})$, and both are equal to

$$\frac{1}{N^t} \mathbb{P} \left[r' = r \wedge \text{chal}' = \text{chal} \mid \text{chal}'_{r'} = 0 \wedge \text{chal}' \leftarrow \mathcal{D}_{\mathcal{H}} \wedge r' \leftarrow \{1, \dots, t\} \right].$$

Since the non-abort distributions are identical, the adversary's probability of success in $\mathbf{G}_1$ is

$$\mathbb{P}[\mathbf{G}_1 = 1] \leq \mathbb{P}[\mathbf{G}_0 = 1] + \text{negl}(\lambda).$$

Notice that we no longer use the secret key sk . This completes the reduction from $\text{Extractability}_{\mathcal{A}, \text{Base ORCAS}}$ to $\mathcal{B}$.

Suppose there exists a probabilistic polynomial-time adversary $\mathcal{A}$ winning game $\mathbf{G}_0$ with non-negligible probability. So $\mathcal{A}$ also wins game $\mathbf{G}_1$ with non-negligible probability, which we will call $\epsilon(\lambda)$. This is the probability that, instead of aborting, $\mathcal{B}$ returns a message and signature (m, σ) , where $\sigma = (\text{chal}, \text{resp}_1, \dots, \text{resp}_t)$. We consider two cases, based on whether $\mathcal{A}$ has or has not queried the random oracle $\mathcal{O}_{\mathcal{H}}$ to construct σ . Let $\epsilon_0(\lambda)$ be the probability $\mathcal{B}$ outputs (m, σ) such that $\text{chal} \neq \mathcal{H}(x)$ for all $x \in \mathcal{Q}$, and $\epsilon_1(\lambda)$ the probability that $\text{chal} = \mathcal{H}(x^*)$ for some $x^* \in \mathcal{Q}$. If $\text{chal} = \mathcal{H}(x^*)$ we run $\mathcal{B}$ again with the same randomness but a fresh response $\mathcal{H}'(x^*) \leftarrow \mathcal{D}_{\mathcal{H}}$ to the random oracle query x^* . By the Forking Lemma [8, Theorem 1], if $\epsilon_1(\lambda)$ is non-negligible, then with non-negligible probability the second run of $\mathcal{B}$ outputs (m', σ') such that $\text{chal}' = \mathcal{H}'(x^*)$. Suppose both runs of $\mathcal{B}$ output message-signature pairs as desired, with $\text{chal} = \mathcal{H}(x^*)$ and $\text{chal}' = \mathcal{H}'(x^*)$. Define $\text{comm}_i = [\mathbf{g}]^{\text{resp}_i} \star E_{\text{chal}_i}$ and $\text{comm}'_i = [\mathbf{g}]^{\text{resp}'_i} \star E_{\text{chal}'_i}$. Since neither run of $\mathcal{B}$ aborted, we know that $\text{Verify}(\text{pk}, m, \sigma) = \text{Verify}(\text{pk}, m', \sigma') = 1$. This gives us two equalities:

$$\begin{aligned} \mathcal{H}(m || \text{comm}_1 || \dots || \text{comm}_t) &= \text{chal} = \mathcal{H}(x^*), \\ \mathcal{H}'(m' || \text{comm}'_1 || \dots || \text{comm}'_t) &= \text{chal}' = \mathcal{H}'(x^*). \end{aligned}$$

Since the probability of a collision is negligible, with overwhelming probability $m || \text{comm}_1 || \dots || \text{comm}_t = m' || \text{comm}'_1 || \dots || \text{comm}'_t = x^*$. Since $\mathcal{D}_{\mathcal{H}}$ is large, with overwhelming probability $\mathcal{H}(x^*) \neq \mathcal{H}'(x^*)$. So

$\text{comm}_i = \text{comm}'_i$ for each round i , but there is at least one round j such that $\text{chal}_j \neq \text{chal}'_j$. We can now compute a path between two distinct public keys E_{chal_j} and $E_{\text{chal}'_j}$. Indeed,

$$[\mathbf{g}]^{\text{resp}_j} \star E_{\text{chal}_j} = \text{comm}_j = \text{comm}'_j = [\mathbf{g}]^{\text{resp}'_j} \star E_{\text{chal}'_j}.$$

Therefore $s = \text{resp}'_j - \text{resp}_j \pmod N$ is a solution to MT-GAIP, with $[\mathbf{g}]^s \star E_{\text{chal}'_j} = E_{\text{chal}_j}$. To complete the proof, we show that $\epsilon_1(\lambda)$ is non-negligible. Since $\epsilon(\lambda) = \epsilon_0(\lambda) + \epsilon_1(\lambda)$ is non-negligible by assumption, it suffices to show that $\epsilon_0(\lambda)$ is negligible; that is, $\mathcal{A}$ cannot win with non-negligible probability unless $\text{chal} = \mathcal{H}(x^*)$ for some $x^* \in \mathcal{Q}$. Suppose $\mathcal{B}$ outputs (m, σ) such that $\text{chal} \neq \mathcal{H}(x)$ for all $x \in \mathcal{Q}$. Define $x^* = m || \text{comm}_1 || \dots || \text{comm}_t$. Since $\text{Verify}(\text{pk}, m, \sigma) = 1$, we know that $\text{chal} = \mathcal{H}(x^*)$ and thus $x^* \notin \mathcal{Q}$. In fact, we claim that at the time $\mathcal{A}$ returns (m, σ) the random oracle is undefined at x^* ($\mathcal{H}[x^*] = \perp$). So $\mathcal{H}(x^*)$ is sampled from $\mathcal{D}_{\mathcal{H}}$ at the time of verification, meaning $\text{Verify}(\text{pk}, m, \sigma) = 1$ only if $\mathcal{A}$ succeeds in blindly guessing $\mathcal{H}(x^*)$, which happens with probability $\frac{1}{|\mathcal{D}_{\mathcal{H}}|}$. Therefore $\epsilon_0(\lambda) \leq \frac{1}{|\mathcal{D}_{\mathcal{H}}|}$ is negligible. To prove the claim, suppose for sake of contradiction that $\mathcal{H}[x^*] \neq \perp$ when $\mathcal{A}$ returns. Since $\mathcal{A}$ has never queried $\mathcal{O}_{\mathcal{H}}(x^*)$, it must be that $\mathcal{H}[x^*]$ was defined during some pre-signing query $\mathcal{O}_{\text{PS}}(m', Y)$. Let $\widehat{\sigma} = (r, \widehat{\text{chal}}', \widehat{\text{resp}}_1, \dots, \widehat{\text{resp}}_t)$ be the pre-signature output by this computation, and let $(\text{comm}'_1, \dots, \text{comm}'_t)$ be the commitment used to generate $\widehat{\sigma}$. Assuming $\mathcal{H}[x^*]$ was defined here, we must have $x^* = m' || \text{comm}'_1 || \dots || \text{comm}'_t$ and therefore $\widehat{\text{chal}}' = \text{chal} = \mathcal{H}(x^*)$. Since $\text{PreVerify}(\text{pk}, m, Y, \widehat{\sigma}) = 1$, we know that $\widehat{\text{chal}}'_r = \text{chal}_r = 0$ and thus $[\mathbf{g}]^{\text{resp}_r} \star E_0 = \text{comm}_r = \text{comm}'_r = [\mathbf{g}]^{\widehat{\text{resp}}_r} \star Y$. Define $y = \text{Ext}(\sigma, \widehat{\sigma}, Y) = \text{resp}_r - \widehat{\text{resp}}_r$. So $[\mathbf{g}]^y \star E_0 = Y$, meaning $(Y, y) \in \mathcal{R}$. Since $(\widehat{\sigma}, Y) \in T[m]$ this would cause $\mathcal{B}$ to abort during its extraction checks, but we assume that $\mathcal{B}$ does not abort. $\square$

Lemma 6.3.6. *Base ORCAS satisfies unlinkability for Filtered CSI-FiSh.*

Proof. We will show that unlinkability is implied by strongly canonical signing, which we have already shown the scheme to have. Recalling the algorithm $\text{CanonicalSign}_{(Y,y)}$ from the definition of strongly canonical signing, we can rewrite the algorithm Challenge from the security game as follows.

$\text{Challenge}_b(m, (Y, y))$	$\text{CanonicalSign}_{(Y,y)}(\text{sk}, m)$
101 : assert $(Y, y) \in \mathcal{R}$	201 : $\widehat{\sigma} \leftarrow \text{PreSign}(\text{sk}, m, Y)$
102 : $\sigma_0 \leftarrow \text{CanonicalSign}_{(Y,y)}(\text{sk}, m)$	202 : $\sigma \leftarrow \text{Adapt}(\widehat{\sigma}, y)$
103 : $\sigma_1 \leftarrow \text{Sign}(\text{sk}, m)$	203 : return σ
104 : return σ_b	

For a scheme with strongly canonical signing, by definition σ_0 is distributed identically to σ_1 for every $(Y, y) \in \mathcal{R}$. Therefore the output of Challenge is independent of b , meaning $\mathcal{A}$ succeeds with probability exactly $\frac{1}{2}$. $\square$

Lemma 6.3.7. *Base ORCAS has pre-verify soundness for Filtered CSI-FiSh.*

Proof. Consider a statement $Y \notin \mathcal{L}$, a key pair (sk, pk) , a message m , and a pre-signature $\widehat{\sigma}$. Since the first step of pre-verification is to check whether $Y \in \mathcal{L}$, we are guaranteed that $\text{PreVerify}(\text{pk}, m, Y, \widehat{\sigma}) = 0$. $\square$

The previous lemmas of this section contain the proof of the fact that Base ORCAS is correct and secure for Filtered CSI-FiSh.

Theorem 6.3.8. *Suppose that MT-GAIP is hard for the action of the group G on the set X (for example the action of $G = \text{cl}(\mathcal{O})$ on $X = \mathcal{Ell}_{\mathcal{O}}(\mathbb{F}_p)$). Base ORCAS is a correct, pre-signature adaptable, strongly full extractable, uniquely extractable, unlinkable and pre-verify sound adaptor signature for Filtered CSI-FiSh in the random oracle model.*

In our paper [12, Sections 4.2,4.3], we reduce the security of Base ORCAS for CSI-FiSh to its security for Filtered CSI-FiSh, using the language of filtered signatures, proving Theorem 6.2.1.

Bibliography

- [1] S. AGRAWAL, D. BONEH, AND X. BOYEN, *Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE*, in Advances in cryptology—CRYPTO 2010, vol. 6223 of Lecture Notes in Comput. Sci., Springer, Berlin, 2010, pp. 98–115.
- [2] M. AJTAI AND C. DWORK, *A public-key cryptosystem with worst-case/average-case equivalence*, in STOC '97 (El Paso, TX), ACM, New York, 1999, pp. 284–293.
- [3] J. ALWEN AND C. PEIKERT, *Generating shorter bases for hard random lattices*, in STACS 2009: 26th International Symposium on Theoretical Aspects of Computer Science, vol. 3 of LIPIcs. Leibniz Int. Proc. Inform., Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2009, pp. 75–86.
- [4] L. AUMAYR, O. ERSOY, A. ERWIG, S. FAUST, K. HOSTÁKOVÁ, M. MAFFEI, P. MORENO-SANCHEZ, AND S. RIAHI, *Generalized Channels from Limited Blockchain Scripts and Adaptor Signatures*, in Advances in cryptology—ASIACRYPT 2021. Part II, vol. 13091 of Lecture Notes in Comput. Sci., Springer, Cham, [2021] ©2021, pp. 635–664.
- [5] L. BABAI, *On Lovász' lattice reduction and the nearest lattice point problem*, Combinatorica, 6 (1986), pp. 1–13.
- [6] R. BAECKER, P. GERHART, J. KATZ, AND D. SCHRÖDER, *Fair Exchange for Decentralized Autonomous Organizations via Threshold Adaptor Signatures*. Cryptology ePrint Archive, Paper 2025/388, 2025. <https://eprint.iacr.org/2025/388>.
- [7] S. BAI, L. DUCAS, E. KILTZ, T. LEPOINT, V. LYUBASHEVSKY, P. SCHWABE, G. SEILER, AND D. STEHLÉ, *CRYSTALS-Dilithium, Algorithm Specifications and Supporting Documentation (Version 3.1)*, 2021. NIST Round 3 Submission Package.
- [8] M. BELLARE AND G. NEVEN, *Multi-signatures in the plain public-key model and a general forking lemma*, in Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006, A. Juels, R. N. Wright, and S. D. C. di Vimercati, eds., ACM, 2006, pp. 390–399.
- [9] D. J. BERNSTEIN, J. BUCHMANN, AND E. DAHMEN, *Post-Quantum Cryptography*, Springer Berlin Heidelberg, 2009.
- [10] M. BERTUZZO, *Lattice-based Cryptography*, 2023. Università di Genova, Master Thesis (in Italian).
- [11] W. BEULLENS, T. KLEINJUNG, AND F. VERCAUTEREN, *CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations*, in Advances in Cryptology – ASIACRYPT 2019, S. D. Galbraith and S. Moriai, eds., Cham, 2019, Springer International Publishing, pp. 227–247.

- [12] R. CARTOR, N. DALY, G. GAGGERO, J. T. LEGROW, A. SANGUINETI, AND S. SCONZA, *Post-Quantum Adaptor Signatures with Strong Security from Cryptographic Group Actions*, in 17th International Workshop, PQCrypto 2026, Saint-Malo, France, April 14–16, 2026, Proceedings, Part I, M. Bardet and R. Niederhagen, eds., vol. 16491 of Lecture Notes in Computer Science, Springer Cham, 2026, pp. 370–405.
- [13] D. CASH, D. HOFHEINZ, AND E. KILTZ, *How to delegate a lattice basis*. Cryptology ePrint Archive, Paper 2009/351, 2009. <https://eprint.iacr.org/2009/351>.
- [14] D. CASH, D. HOFHEINZ, E. KILTZ, AND C. PEIKERT, *Bonsai trees, or how to delegate a lattice basis*, J. Cryptology, 25 (2012), pp. 601–639.
- [15] J. W. S. CASSELS, *An Introduction to the Geometry of Numbers*, vol. Bd. 99 of Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete, Springer-Verlag, Berlin-Göttingen-Heidelberg, 1959.
- [16] W. CASTRYCK, T. LANGE, C. MARTINDALE, L. PANNY, AND J. RENES, *CSIDH: an efficient post-quantum commutative group action*, in Advances in cryptology—ASIACRYPT 2018. Part III, vol. 11274 of Lecture Notes in Comput. Sci., Springer, Cham, 2018, pp. 395–427.
- [17] J.-M. COUVEIGNES, *Hard Homogeneous Spaces*. Cryptology ePrint Archive, Paper 2006/291, 2006.
- [18] D. A. COX, *Primes of the form $x^2 + ny^2$ —Fermat, class field theory, and complex multiplication*, AMS Chelsea Publishing, Providence, RI, third ed., [2022] ©2022. With contributions by Roger Lipsett.
- [19] R. CRAMER AND V. SHOUP, *A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack*, in Advances in cryptology—CRYPTO '98 (Santa Barbara, CA, 1998), vol. 1462 of Lecture Notes in Comput. Sci., Springer, Berlin, 1998, pp. 13–25.
- [20] W. DAI, T. OKAMOTO, AND G. YAMAMOTO, *Stronger security and generic constructions for adaptor signatures*, in Progress in cryptology—INDOCRYPT 2022, vol. 13774 of Lecture Notes in Comput. Sci., Springer, Cham, [2022] ©2022, pp. 52–77.
- [21] ———, *Stronger Security and Generic Constructions for Adaptor Signatures*, in Progress in cryptology—INDOCRYPT 2022, vol. 13774 of Lecture Notes in Comput. Sci., Springer, Cham, [2022] ©2022, pp. 52–77.
- [22] P. DAS, S. FAUST, AND J. LOSS, *A Formal Treatment of Deterministic Wallets*, in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11–15, 2019, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, eds., ACM, 2019, pp. 651–668.
- [23] W. DIFFIE AND M. E. HELLMAN, *New directions in cryptography*, IEEE Trans. Inform. Theory, IT-22 (1976), pp. 644–654.
- [24] L. DUCAS, T. LEPOINT, V. LYUBASHEVSKY, P. SCHWABE, G. SEILER, AND D. STEHLÉ, *CRYSTALS-Dilithium: Digital Signatures from Module Lattices*, 2017. NIST Submission Package.

- [25] A. ERWIG AND S. RIAHI, *Deterministic Wallets for Adaptor Signatures*, in Computer Security - ESORICS 2022 - 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26-30, 2022, Proceedings, Part II, V. Atluri, R. D. Pietro, C. D. Jensen, and W. Meng, eds., vol. 13555 of Lecture Notes in Computer Science, Springer, 2022, pp. 487–506.
- [26] A. FIAT AND A. SHAMIR, *How to prove yourself: practical solutions to identification and signature problems*, in Advances in cryptology—CRYPTO '86 (Santa Barbara, Calif., 1986), vol. 263 of Lecture Notes in Comput. Sci., Springer, Berlin, 1987, pp. 186–194.
- [27] C. GENTRY, C. PEIKERT, AND V. VAIKUNTANATHAN, *Trapdoors for Hard Lattices and New Cryptographic Constructions*. Cryptology ePrint Archive, Paper 2007/432, 2007. <https://eprint.iacr.org/2007/432>.
- [28] C. GENTRY, C. PEIKERT, AND V. VAIKUNTANATHAN, *Trapdoors for hard lattices and new cryptographic constructions [extended abstract]*, in STOC'08, ACM, New York, 2008, pp. 197–206.
- [29] P. GERHART, D. SCHRÖDER, P. SONI, AND S. A. THYAGARAJAN, *Foundations of adaptor signatures*, in Advances in cryptology—EUROCRYPT 2024. Part II, vol. 14652 of Lecture Notes in Comput. Sci., Springer, Cham, [2024] ©2024, pp. 161–189.
- [30] P. GERHART, D. SCHRÖDER, P. SONI, AND S. A. THYAGARAJAN, *Foundations of adaptor signatures*. Cryptology ePrint Archive, Paper 2024/1809, 2024.
- [31] N. GLAESER, M. MAFFEI, G. MALAVOLTA, P. MORENO-SANCHEZ, E. TAIRI, AND S. A. K. THYAGARAJAN, *Foundations of Coin Mixing Services*, in Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022, H. Yin, A. Stavrou, C. Cremers, and E. Shi, eds., ACM, 2022, pp. 1259–1273.
- [32] O. GOLDBREICH, S. GOLDWASSER, AND S. HALEVI, *Eliminating decryption errors in the Ajtai-Dwork cryptosystem*, in Advances in cryptology—CRYPTO '97 (Santa Barbara, CA, 1997), vol. 1294 of Lecture Notes in Comput. Sci., Springer, Berlin, 1997, pp. 105–111.
- [33] —, *Public-key cryptosystems from lattice reduction problems*, in Advances in cryptology—CRYPTO '97 (Santa Barbara, CA, 1997), vol. 1294 of Lecture Notes in Comput. Sci., Springer, Berlin, 1997, pp. 112–131.
- [34] J. HOFFSTEIN, J. PIPHER, AND J. H. SILVERMAN, *An Introduction to Mathematical Cryptography*, Undergraduate Texts in Mathematics, Springer, New York, 2008.
- [35] D. HUSEMÖLLER, *Elliptic curves*, vol. 111 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 2004. With appendices by Otto Forster, Ruth Lawrence and Stefan Theisen.
- [36] IBM QUANTUM COMPUTING, *IBM Quantum Computation Roadmap*. <https://www.ibm.com/quantum/hardware>, 2026.
- [37] P. JANA, S. SHAW, AND R. DUTTA, *Compact Adaptor Signature from Isogenies with Enhanced Security*, in Cryptology and network security. Part I, vol. 14905 of Lecture Notes in Comput. Sci., Springer, Singapore, [2025] ©2025, pp. 77–100.
- [38] S. LANG, *Elliptic functions*, vol. 112 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 1987. With an appendix by J. Tate.

- [39] A. LANGLOIS AND D. STEHLÉ, *Worst-case to average-case reductions for module lattices*, Des. Codes Cryptogr., 75 (2015), pp. 565–599.
- [40] A. K. LENSTRA, H. W. LENSTRA, JR., AND L. LOVÁSZ, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.
- [41] A. K. LENSTRA, H. W. LENSTRA, JR., M. S. MANASSE, AND J. M. POLLARD, *The number field sieve*, in The development of the number field sieve, vol. 1554 of Lecture Notes in Math., Springer, Berlin, 1993, pp. 11–42.
- [42] V. LYUBASHEVSKY, C. PEIKERT, AND O. REGEV, *On ideal lattices and learning with errors over rings*, J. ACM, 60 (2013), pp. Art. 43, 35.
- [43] V. MADATHIL, S. A. K. THYAGARAJAN, D. VASILOPOULOS, L. FOURNIER, G. MALAVOLTA, AND P. MORENO-SANCHEZ, *Cryptographic Oracle-based Conditional Payments*, in 30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023, The Internet Society, 2023.
- [44] D. MICCIANCIO AND S. GOLDWASSER, *Complexity of lattice problems*, vol. 671 of The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Boston, MA, 2002. A cryptographic perspective.
- [45] D. MICCIANCIO AND O. REGEV, *Worst-case to average-case reductions based on Gaussian measures*, SIAM J. Comput., 37 (2007), pp. 267–302.
- [46] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, *FIPS 180-4: Secure Hash Standard (SHS)*, Tech. Rep. NIST Computer Security Resource Center FIPS 180-4, U.S. Department of Commerce, Washington, D.C., 2012.
- [47] —, *Post-Quantum Cryptography Standardization Process*. <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization>, 2017.
- [48] —, *Post-Quantum Cryptography: Additional Digital Signature Schemes*. <https://csrc.nist.gov/projects/pqc-dig-sig>, 2022.
- [49] —, *Module-Lattice-Based Digital Signature Standard*, Tech. Rep. NIST Computer Security Resource Center FIPS 204, U.S. Department of Commerce, Washington, D.C., 2024.
- [50] P. NGUYEN, *Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97 - Advances in cryptology—CRYPTO '99*, in Proceedings of the 19th Annual International Cryptology Conference held in Santa Barbara, CA, August 15–19, 1999, M. Wiener, ed., vol. 1666 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1999, pp. 288–304.
- [51] O. REGEV, *Lattices in Computer Science*, Tel Aviv University, 2004. Lecture Notes.
- [52] —, *New lattice-based cryptographic constructions*, J. ACM, 51 (2004), pp. 899–942. Preliminary Version in STOC 2005.
- [53] —, *On lattices, learning with errors, random linear codes, and cryptography*, J. ACM, 56 (2009), pp. Art. 34, 40.

- [54] R. L. RIVEST, A. SHAMIR, AND L. ADLEMAN, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM, 21 (1978), pp. 120–126.
- [55] R. SAKAI AND J. FURUKAWA, *Identity-based broadcast encryption*. Cryptology ePrint Archive, Paper 2007/217, 2007. <https://eprint.iacr.org/2007/217>.
- [56] C. SCHNORR, *Efficient Signature Generation by Smart Cards*, J. Cryptol., 4 (1991), pp. 161–174.
- [57] S. SCONZA, *Isogney Volcanoes*, 2022. Università di Genova, Master Thesis.
- [58] A. SHAMIR, *Identity-based cryptosystems and signature schemes*, in Advances in cryptology (Santa Barbara, Calif., 1984), vol. 196 of Lecture Notes in Comput. Sci., Springer, Berlin, 1985, pp. 47–53.
- [59] P. W. SHOR, *Algorithms for quantum computation: discrete logarithms and factoring*, in 35th Annual Symposium on Foundations of Computer Science (Santa Fe, NM, 1994), IEEE Comput. Soc. Press, Los Alamitos, CA, 1994, pp. 124–134.
- [60] ———, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., 26 (1997), pp. 1484–1509.
- [61] J. H. SILVERMAN, *The arithmetic of elliptic curves*, vol. 106 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1986.
- [62] ———, *Advanced topics in the arithmetic of elliptic curves*, vol. 151 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1994.
- [63] A. STOLBUNOV, *Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves*, Adv. Math. Commun., 4 (2010), pp. 215–235.
- [64] A. SUTHERLAND, *MIT Lecture Notes, 18.783 - Spring 2017*, tech. rep., MIT, 217. <https://math.mit.edu/classes/18.783/2017/index.html>.
- [65] E. TAIRI, P. MORENO-SANCHEZ, AND M. MAFFEI, *A^2l : Anonymous Atomic Locks for Scalability in Payment Channel Hubs*, in 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021, IEEE, 2021, pp. 1834–1851.
- [66] E. TAIRI, P. MORENO-SANCHEZ, AND M. MAFFEI, *Post-Quantum Adaptor Signature for Privacy-Preserving Off-Chain Payments*, in FC 2021 Part II, N. Borisov and C. Díaz, eds., vol. 12675 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2021, pp. 131–150.
- [67] C. YANG, S. ZHENG, L. WANG, X. LU, AND Y. YANG, *Hierarchical Identity-based Broadcast Encryption Scheme from LWE*, Journal of Communications and Networks, 16 (2014).