UNIVERSITY OF TRENTO

DEPARTMENT OF MATHEMATICS
Master Degree in Mathematics

Thesis

Post-Quantum Cryptography: Towards Commutative Supersingular Isogeny Key Exchange

Supervisor:
Nadir Murru

Candidate:
Giovanni Tognolini

Academic Year 2019–2020

“Agnese mi parlava nella sabbia infuocata
ed io non so perchè, non l’ho dimenticata”

Contents

Introduction

The Diffie-Hellman key-exchange protocol is, both literally and figuratively, at the foundation of public-key cryptography. The goal is for two parties, Alice and Bob, to derive a shared secret from each other's public keys and their own private keys. Diffie and Hellman's original solution [29] is beautifully and brutally simple: given a fixed prime p and a primitive element g in the finite field $\mathbb{F}_p$ (that is, a generator of the multiplicative group $\mathbb{F}_p^*$), Alice and Bob choose secret keys a and b, respectively, in $\mathbb{Z}/(p-1)\mathbb{Z}$. Alice computes and publishes her public key $A = g^a$, and Bob his public key $B = g^b$; the shared secret value is $S = g^{ab}$, which Alice computes as $S = B^a$, Bob as $S = A^b$. The security of the shared secret depends on the hardness of the computational Diffie-Hellman Problem (CDHP), which is to compute S given only A, B, and the public data of the structures that they belong to. For finite-field Diffie-Hellman, this means computing g^{ab} given only g, g^a, and $g^b \pmod p$. The principal approach to solve the CDHP is to solve the Discrete Logarithm Problem (DLP), which is to compute x from g and g^x. We thus recover a from $A = g^a$ (or, equivalently, b from $B = g^b$), then power B by a (or A by b) to recover S. Attacking the DLP means directly attacking one of the public keys, regardless of any particular shared secret they may be used to derive. Over the past four decades, the Diffie-Hellman protocol has been generalized from multiplicative groups of finite fields to a range of other algebraic groups, most notably elliptic curves [68, 55]. Partly motivated by this cryptographic application, there has been great progress in discrete logarithm algorithms for some groups.

The most stunning development in discrete logarithm algorithms came with the rise of the quantum computation paradigm: Shor's quantum algorithm [84] solves the discrete logarithm problem (and thus breaks Diffie-Hellman) in any group in polynomial time and space on a quantum computer. The development of quantum computers of even modest capacity capable of running Shor's algorithm remains a big challenge in experimental physics: at the time of writing, the largest number factored by Shor's algorithm is 35, using IBM's Q System One. Quantum computers are developing fast, and cryptographic research has already bent itself to the construction of post-quantum cryptosystems, designed to be used on conventional computers while resisting known quantum attacks.

The common use of Diffie-Hellman makes the search for a drop-in post-quantum replacement for this protocol particularly relevant today. While many promising post-quantum candidates for public-key encryption and signatures have been developed, finding a simple post-quantum drop-in replacement for Diffie–Hellman (as opposed to a KEM) has proven to be surprisingly complicated. Perhaps surprisingly, given the loudly trumpeted quantum destruction of elliptic curve cryptography by Shor's algorithm, the most serious candidates for post-quantum Diffie–Hellman come from isogeny-based cryptography, which is founded in the deeper theory of elliptic curves. The key idea in moving from conventional elliptic-

curve cryptography to isogeny-based cryptography is that points on curves are replaced with entire curves, and relationships between points (scalars and discrete logarithms) are replaced with relationships between curves (isogenies). Isogeny classes have just enough algebraic structure to define efficient asymmetric cryptosystems, but not enough to make them vulnerable to Shor's algorithm.

In this discussion we describe the process that led to the cryptographic community to find a scheme that naturally replace the Diffie-Hellman protocol, therefore we study in detail its safety properties and specifics.

The Plan. In the first chapter we will give a general overview of our framework, introducing the well-known discrete logarithm problem and the classical Diffie-Hellman protocol as a scheme on a group of the form $\mathbb{F}_p^*$. We will then see how the cryptographic community has modified the underlying group in order to make the scheme more secure. In particular, we will focus on ECDH, but we will not fail to mention other instances, such as algebraic curves with genus greater than 1. We will then introduce the threat represented by Shor and Grover's algorithms, and how the NIST is moving to standardize a family of algorithms able to resist not only the classical computation model, but also the quantum one. The result of this effort is the post quantum standardization process which goes by the name *NIST PQC* (Post Quantum Competition). We will therefore show how, despite the large amount of new schemes submitted to the NIST attention, none of these naturally replace Diffie-Hellman. Up to now, the only known scheme that plays this role is CSIDH, and it was not presented to NIST solely because it was discovered after the submissions' end line. This scheme is based on isogenies of supersingular curves in finite fields. Unlike other post quantum algorithms, isogeny-based cryptosystems require a deep understanding of arithmetic of elliptic curves in finite fields. The following chapters lay the basis for understanding these schemes.

The second chapter is entirely devoted to elliptic curves: in this regard the elementary results are briefly summarized. We will then describe the main kind of maps between elliptic curves, with particular emphasis on isogenies. Finally, we will recall some non-trivial results that will nevertheless be of fundamental importance throughout our discussion.

The third chapter joins the theory of elliptic curves with that of graphs. We will begin by describing the ring structure of the endomorphisms set of an elliptic curve, showing how this can take essentially three forms: an order in an imaginary quadratic field, an order in a quaternion algebra ramified at p and ∞, or simply the integer ring $\mathbb{Z}$. In the case of curves defined over a finite field the last case is excluded. In the first case the curve will be called ordinary, in the second case supersingular. We will describe in detail isogeny graphs, i.e. those (multi)graphs whose vertices are classes of isomorphism of elliptic curves, and whose edges are isogenies between these classes. We will observe how the structure of the endomorphism ring $\mathrm{End}(E)$ of a curve is decisive for the structure of the resulting graph. In the case of ordinary curves the graph will have a structure very regular, which takes the name of isogeny volcano; in the case of supersingular curves in general we will have a k-regular graph, but under some restrictions we will show how to get an isogeny volcano in this case as well.

In section four we will show why we were looking for a structure like that of isogeny volcanoes: in fact we will show how it is possible to define a quantum resistant protocol

above. The first to realize this possibility was Couveignes, in 1997, thanks to the use of ordinary elliptical curves [23]. However, the protocol he proposed remained unpublished for about ten years, until Rostovtsev and Stolbunov independently proposed a modified (and improved) version. Nevertheless, the algorithm is still impracticable. A few years ago, however, a way was found to make this protocol feasible. The key is to use supersingular curves, and to impose some limitations to the respective isogeny graph, for which the resulting graph is an isogeny volcano, exactly like that of ordinary curves. This observation has therefore allowed contemporary research to pick up again a protocol that was in danger of falling into disuse, and gave it back its strength. The result is the algorithm that goes under the name of CSIDH, on which we focus all the remaining part of our discussion.

Chapter 1

The Framework

Modern cryptography bases its security on some difficult math problems, such as factoring large integers, the discrete logarithm in $\mathbb{Z}_p^*$ and the discrete logarithm on the group of elliptic curves. Up to now, for these problems there is no classical algorithm capable of breaking them in polynomial time. The situation is different if we extend the analysis not only to algorithms implemented with a classical computer, but we also include those designed for a quantum computer, which works over a different computational model. In this chapter we introduce the threat represented by quantum computers, with particular emphasis on Shor and Grover's algorithms, and the implications that their implementation would have for modern forms of communication. We will see how the cryptographic community and the NIST (National Institute of Standards and Technology) are moving to create quantum resistant schemes, in order to replace those currently in use. We will focus in particular on the Diffie-Hellman key exchange, given its importance and centrality within almost all cryptographic protocols, and we will analyze how a post quantum analogue of this primitive can be found. In particular we will show how isogeny-based cryptography is the only type of cryptography up to now that provides a post quantum algorithm analogous to Diffie Hellman. For additional details, we refer to [16], [5] and [88].

1.1 Pre-Quantum Diffie Hellman

The Discrete Logarithm Problem

Let G a finite abelian group of order N; clearly G is the product of cyclic groups. For a generic $m \in \mathbb{N}^+$, $P \in G$, define $[m]P$ as $P + \cdots + P$ where P appears m times. Define $[0]P := O$ and $[-m]P := [m](-P)$. We call this function the *scalar multiplication map*. Obviously it is an endomorphism, since it respects the underlying group structure on which it acts. We can compute this maps in $O(\log m)$ operations in G, with the well known classical algorithms. The fundamental hard algorithmic problem in G is to compute the inverse of the scalar multiplication operation: that is, computing discrete logarithms.

Definition 1.1 (DLP)**.** The *Discrete Logarithm Problem* in G is, given P and Q in $\langle P \rangle \subseteq G$, compute $x \in \mathbb{Z}$ such that $Q = [x]P$.

This problem, given its importance and centrality in cryptography, has been a subject of deep study by the cryptographic community. We present now the best known attacks on this problem. Any DLP instance in any group G can always be solved using $O(\sqrt{N})$ operations in G, using (for example) Shanks' baby-step giant-step algorithm (BSGS), which also requires $O(\sqrt{N})$ space [82]; Pollard's ρ algorithm reduces the space requirement

to $O(1)$ [78]. If N is composite and its (partial) factorization is known, then we can do better using the Pohlig-Hellman algorithm [77], which solves the DLP by reducing to the DLP in subgroups of G. Observe that the DLP enjoys random self-reducibility: if we have an algorithm that solves DLPs for a large fraction $1/M$ of all possible inputs, then we can solve DLPs for all possible inputs after an expected M random attempts. Suppose we want to solve an arbitrary DLP instance $Q = [x]P$. We choose a random integer r, try to solve $Q = Q + [r]P = [x + r]P$ for $x + r$, and if we succeed then we recover $x = (x + r) - r$. After M randomizations, we expect to find an r for which Q lands in the set of inputs to which the algorithm applies.

The Diffie-Hellman Protocol

Now we briefly recall Diffie-Hellman in the abstract. Let G be a cyclic group of order N, and fix a public generator P of G. Public keys are elements of G; private keys are bit strings, interpreted as elements of $\mathbb{Z}/N\mathbb{Z}$. The protocol goes as follows:

- (Setup) Global parameter of the scheme is a large cyclic group $G = \langle P \rangle$ of order N.
- (Key generation) Alice samples an element $a \in G$, which becomes her private key. Her public key is the element $A := [a]P$. Bob does the same getting $b \in G$ as private key, and $B := [b]P$ as public key.
- (Key exchange) Alice receives B from Bob and computes $[a]B$. Bob receives A from Alice and computes $[b]A$.

Alice and Bob have the same value S because $[a]B = [ab]P = [ba]P = [b]A$. Clearly each (public,private)-keypair $(Q = [x]P, x)$ presents a DLP instance in G. The protocol is summarized in Table 1.1.

Public parameters	A group $G = \langle P \rangle$ of order N.	
	Alice	**Bob**
Pick random secret	$a \in G$	$b \in G$
Compute public data	$A = [a]P$	$B = [b]P$
Exchange data	$A \longrightarrow$	$\longleftarrow B$
Compute shared secret	$S = [a]B$	$S = [b]A$

Table 1.1: Diffie-Hellman key-exchange protocol.

The secret S shared with the Diffie-Hellman key exchange is normally not directly used as a key for symmetric cryptographic systems; rather, it should be treated with a key derivation function (KDF) to produce a correct symmetric key K. This map essentially hashes the secret S, by evenly distributing the entropy of S in K. In this way an attacker who wants to infer any information on K must first compute S.

The security of the (entire) shared secret depends on the hardness of the Computational Diffie-Hellman Problem (CDHP) in G.

Definition 1.2 (CDHP)**.** The *Computational Diffie-Hellman Problem* in G is, given $P, A = [a]P$, and $B = [b]P$ in G, to compute $S = [ab]P$.

Remark 1.3. The two problems described above, DLP and CDHP, are deeply connected: on one hand, an algorithm that can solve DLP in a group G is able to solve CDHP in the same group G by simply considering it as a DLP instance with parameters $([a]P, [ab]P)$. On the other hand, an algorithm able to solve CDHP is not proven to be able to solve DLP. This direction is a subject of many studies, however it is now generally believed that the DLP and CDHP are equivalent for the kinds of G that cryptographers use in practice. Since solving DLP instances is the only way we know to solve CDHP instances, Diffie–Hellman is generally considered to be a member of the DLP-based family of cryptosystems.

The lifespan of keypairs is crucial in Diffie–Hellman-based cryptosystems. We distinguish two modalities in which the key can be used: static and ephemeral mode. Static Diffie-Hellman key exchanges always use the same Diffie-Hellman private keys. So, each time the same parties do a DH key exchange, they end up with the same shared secret. If both DH private keys are reused, the term *static-static* is used. If only one side uses the same key, the term is *ephemeral-static*. Alice may obtain Bob's long-term public key and complete a Diffie-Hellman key exchange with him (and start using the shared secret) without any active involvement on his part. Static Diffie–Hellman is therefore an important example of a Non-Interactive Key Exchange (NIKE) protocol. In some implementations, it might make sense to have one static DH private key, especially on the server side, for performance reasons. Ephemeral Diffie-Hellman generates a new temporary DH private key for every connection: Alice and Bob's keypairs are unique to each execution of the protocol and thus the same key is never used twice. This enables Forward Secrecy (FS), which means that if the long-term private key of the server gets leaked, past communication is still secure. When both sides always create new DH private keys for new connections, this is called *ephemeral-ephemeral*. Ephemeral Diffie–Hellman is therefore essentially interactive.

Remark 1.4 (Public-key validation)**.** Efficient public-key validation[1] is an important, and often overlooked, requirement for many Diffie-Hellman systems, particularly those where keys are re-used. Suppose Alice derives a shared secret key K from a Diffie-Hellman exchange with Bob's public key B, and then uses K to communicate with Bob. A malicious Bob might construct an invalid public key B in such a way that K reveals information about Alice's secret key a. If (a, A) is ephemeral then Bob has learned nothing useful about a, since it will never be used again; but if the keypair (A, a) is to be reused, as in static Diffie-Hellman, then secret information has been leaked, and Alice thus becomes vulnerable to active attacks[2]. Public key validation is simple in a finite field: it usually suffices to check the order of the element. Antipa, Brown, Menezes, Struik, and Vanstone describe the process for elliptic-curve public keys [2]. We will see that this is a more serious problem in post-quantum systems.

Concrete Groups

Everything we have described so far has been presented in the abstract, however if we want to instantiate this scheme, we must choose a concrete group G. The difficulty of solving the DLP, and therefore the CDHP, varies according to the representation of G; in general we can quantify this complexity as $O(\sqrt{N})$. We briefly present the most commonly used groups to instantiate the Diffie-Hellman protocol.

- The original algorithm uses the multiplicative group $\mathbb{F}_p^*$ of a finite field $\mathbb{F}_p$. The DLP in a finite field is subexponential: the General Number Field Sieve [62] solves

[1] That is, checking that a public key was honestly generated.

[2] Essentially, they consist in changing the information in some way by conducting some process on the information itself, for example though the modification of transmitted or stored data.

a DLP instances in $\mathbb{F}_p$ in time[3] $L_p[1/3, (64/9)^{1/3}]$. However, the complexity can vary if the underlying field takes on particular forms: in extension fields of large characteristic, or when the characteristic has a special form, the complexity is lower, while still subexponential (see [44]); in the extreme case of extension fields of tiny characteristic, the DLP is quasipolynomial in the field size [14].

- The generalization to groups other than that proposed by the original protocol is immediate: less than ten years after the proposal of Diffie and Hellman, Miller [68] and Koblitz [55], independently and almost simultaneously, propose an analog of this protocol based on the group of points of an elliptic curve. At first glance, elliptic-curve cryptography is just finite-field cryptography with a different algebraic group seamlessly swapped in, and no theoretical modification. But Miller's original article [68] ends with an interesting observation that departs from the multiplicative group perspective:

  > *"Finally, it should be remarked, that even though we have phrased everything in terms of points on an elliptic curve, that, for the key exchange protocol (and other uses as one-way functions), that only the x-coordinate needs to be transmitted. The formulas for multiples of a point cited in the first section make it clear that the x-coordinate of a multiple depends only on the x-coordinate of the original point."*

  Miller is talking about elliptic curves in Weierstrass models $y^2 = x^3 + ax + b$, where $-(x, y) = (x, -y)$, so x-coordinates correspond to group elements modulo sign. The mapping $(m, x(P)) \to x([m]P)$ is mathematically well-defined, because every $[m]$ commutes with $[-1]$. In particular Miller proposes to use as keys not the points of an elliptical curve, but the x-coordinates of these points.

  Remark 1.5. Clearly, we lose nothing in terms of security by doing this: the x-coordinate CDHP reduces immediately to the CDHP in the elliptic curve. Given a general CDHP oracle for E, we can compute $\pm[ab]P$ from $(\pm P, \pm[a]P, \pm[b]P)$ by choosing arbitrary lifts to signed points on E and calling the oracle there; conversely, given an x-coordinate CDHP oracle, we can solve CDHP instances on E by projecting to the x-line, calling the oracle there, and then guessing the sign on S.

  The idea to transmit only the x-coordinates may seem advantageous in terms of reducing bandwidth, but in reality this is not the main advantage, in fact we could encode the point (x_P, y_P) taking into account the coordinate x, in addition to a bit indicating the sign of the coordinate y. It is clear that there is very little to be gained. The real practical value in Miller's idea is that working only with x coordinates is faster, and requires less memory: $x([a]P)$ can be computed from a and $x(P)$ using less field operations than it would be necessary to compute $[a]P$ from a and P. Another important aspect of elliptic curve Diffie-Hellman is that subexponential finite-field DLP algorithms do not apply to general elliptic curves, and, to a general prime-order elliptic curve, there is no algorithm with complexity better than $O(\sqrt{N})$. Indeed, the only way to make use of the geometric structure for general curves over prime fields is to run generic $O(\sqrt{N})$ algorithms on equivalence classes modulo ± 1, that is considering (x, y) and $(x, -y)$ equal. In practise this only improves the running time by a factor of roughly $\sqrt{2}$ [6]. We can do better for some

[3] Recall that $L_X[\alpha, c] = \exp\left((c + o(1))(\log X)^{\alpha}(\log\log X)^{1-\alpha}\right)$.

elliptic curves defined over some extension fields [40, 97, 38], and for some small special classes of curves [66, 14, 33, 86]; but in the more than thirty years since Miller and Koblitz introduced elliptic curve cryptography, this speedup represents the only real non-quantum algorithmic improvement for the general elliptic-curve DLP.

- Going beyond elliptic curves, a range of other algebraic groups have been proposed in cryptography. Koblitz proposed cryptosystems in hyperelliptic curves as a generalization of elliptic curves [56]. Generalizing this group structure to the hyperelliptic case is not straightforward: we cannot define the same group law on the set of points lying on a hyperelliptic curve, instead a group structure can be defined on the so-called Jacobian of a hyperelliptic curve. Indeed for elliptic curves the Jacobian turns out to simply be isomorphic to the usual group on the set of points on this curve (this is basically a corollary of the Abel-Jacobi theorem). Others have suggested Jacobians of general algebraic curves, and abelian varieties [69, 81]; but as the genus of the curve (or the dimension of the abelian variety) grows, index-calculus algorithms become more effective, quickly outperforming generic DLP algorithms. At best, the DLP for curves of fixed genus ≥ 3 is exponential, but easier than $O(\sqrt{N})$ [87, 39, 28, 41]; at worst, as the genus and field size both tend to infinity, the DLP becomes subexponential [30].

- The groups mentioned above are all algebraic groups: elements are represented by tuples of field elements, and group operations are computed using polynomial formulæ. Algebraic groups are well-suited to efficient computation on real-world computer architectures, but they are not the only such groups: another kind consists of class groups of number fields. Buchmann and Williams proposed Diffie–Hellman schemes in class groups of quadratic imaginary orders [13], leading to a series of DLP-based cryptosystems set in more general rings (see [12] for a survey); but ultimately these are all vulnerable to subexponential index-calculus attacks.

In the classical world, therefore, elliptic curves over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$ and genus-2 Jacobians over $\mathbb{F}_{p^2}$ present the hardest known DLP instances with respect to group size (and hence key size). Elliptic curves over prime fields have become, in a sense, the gold standard to which all other groups are compared.

1.2 The Quantum Threat

In 1996 Peter Shor, of Bell Laboratories, designs an algorithm for a quantum computer able to solve the integer factorization problem and discrete logarithm in polynomial time [84]. Starting from Shor's work, in the same laboratories and in the same year, Lov Grover draws a quantum algorithm for attacking symmetric ciphers [43]. In this case the time required is not polynomial in the input dimension and only gives a quadratic speedup over the classical brute force method.

The impact that the implementation of these algorithms would have on all modern cryptography would be significant: public key cryptography bases its security on problems that Shor's algorithm is able to solve easily; the situation is different for private key cryptography: since Grover provides a quadratic speedup over normal brute force, the algorithm could force a 128-bit symmetric key in about 2^{64} iterations, or a 256-bit key in about 2^{128}

iterations. To make these ciphers also resistant to Grover's algorithm, it would therefore be sufficient to double the key length. In any case, it is still clear that a quantum computer would put many forms of modern communication in danger. We refer to Table 1.2 for a summary of the large-scale impact a quantum computer would have on common cryptographic algorithms, such as RSA and AES.

Algorithm	*Purpose*	*Impact from quantum computer*
AES	Encryption	Larger key sized needed
SHA-2, SHA-3	Hash Function	Larger output needed
ECDSA, ECDH	Signatures, key exchange	No longer secure
RSA	Signatures, key establishment	No longer secure
DSA	Signatures, key exchange	No longer secure

Table 1.2: Impact of quantum computing on common cryptographic algorithms

To date there are only prototypes of this type of computer, that is machines that are not yet able to do run the algorithms described above with numbers of cryptographic relevance, however, large companies are investing heavily for make this happen. When that happens, quantum computers will then have reached what is now only their potential ability to solve the aforementioned problems, and will therefore be able to effectively break fragile ciphers. This concept is called *quantum supremacy*, or "quantum advantage", to emphasize the fact that this technology allows to go beyond the normal limits of classical computers. In computational-complexity-theoretic terms, this generally means providing a superpolynomial speedup over the best known or possible classical algorithm.

Given the scale of the problem, it is useful at this point to outline the development of quantum computers over the last decade: Google had already announced plans to demonstrate quantum supremacy by the end of 2017 by solving the problem with a superconducting 49 qubits-array. In October 2017, IBM demonstrated the 56 qubits simulation on a conventional supercomputer, increasing the number of qubits needed for quantum supremacy. In March 2018 Google announced Bristiecone, a new 72 qubits quantum computer processor, but it is still trying to make it work: most qubits must be used to do error correction and routine operations of this type.

It is not easy to predict how soon a large-scale quantum computer will be built, although it is very likely that within the next 20 years or so, quantum computers large enough to break all the public key encryption schemes currently in use will exist. The important fact is that, a priori of the exact year in which quantum computers will represent a real risk, we must ensure a secure migration to new cryptographic protocols, which guarantee us protection from these threats. Migrating to new protocols smoothly and securely is an inherently long process, just think that it took almost 20 years to implement our modern public key cryptographic infrastructure, so we have to start working in this direction right now.

1.3 Quantum and Post-Quantum Cryptography

There are basically two approaches for the creation of quantum resistant cryptosystems: quantum cryptography and post-quantum cryptography.

Quantum Cryptography

Unlike conventional cryptography, which is mainly based on number theory, quantum cryptography uses the laws of quantum physics to generate keys and transfer information. However, this approach collides with the numerous physical limitations of the case, as the quantum cryptography will affect the whole cryptographic infrastructure, which involves software, crypto-processors, hardware customization and communications infrastructure. Any changes to this infrastructure must be carefully planned and tested to ensure compatibility with existing components.

Regarding crypto-processors, which are hardware chips carrying several crypto functions such as encryption, signature generation and hashing: they are embedded in many devices such as point-of-sale systems, automated teller machines and smartphones (SIM cards). These cryptosystems rely on the current cryptographic standards, depending on the applications, for example, for lightweight applications with lower amounts of data these crypto-processors mostly rely on small crypto algorithms such as AES. Moving to quantum-resistant crypto primitives will affect the performance of these crypto-processors, since they involve more computations, and they could even render some processors obsolete. Therefore companies may be required to buy new hardwares to handle the increased workload.

Regarding communications infrastructure: with quantum cryptography, we will need very direct channels, and we will also need to amplify them often: quantum key distribution is possible using optical fiber over few tens of kilometers but not more due to single photons (during quantum communication, information is encoded into photons) getting absorbed by the fiber. In addition, quantum cryptography require a technology that will force us to rebuild all various underground connections between servers.

Furthermore this technology does not have enough algorithms to perform the various routines needed in cryptography, for example up to now it is not possible to have an authentication scheme.

Post Quantum Cryptography

Post quantum cryptography, on the other hand, involves the use of classical algorithms, which base their security on problems that both normal and quantum computers can not break. Compared to the first approach, the latter has the advantage of providing algorithms that work efficiently on a classical computer, both in terms of time and memory.

We list below the most important families into which the various post-quantum algorithms are usually divided. For each of these families we briefly state the mathematical problem on which they base their safety, and we describe their strengths and weaknesses.

- **Lattice-based cryptography**: Lattice-based cryptosystems base their security on the well known *shortest vector problem* (SVP), for which, given a lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ we are asked to find the shortest non-zero vector v, i.e. such that

  $$\|v\| = \min_{w \in \mathcal{L} \setminus \{\mathbf{0}\}} \|w\|$$

  We briefly present the best known lattice-based algorithm, the NTRUEncrypt public key cryptosystem, also known as the NTRU encryption algorithm. Each NTRU's instance is specified by three integer parameters (N, p, q) which represent the maximal

degree $N-1$ for all polynomials in the truncated ring $R := \mathbb{Z}[X]/(X^N-1)$, a small modulus and a large modulus, respectively, where it is assumed that N is prime, q is always larger than p, and p and q are co-prime; and four sets of polynomials $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_m$ and $\mathcal{L}_\varphi$ (a polynomial part of the private key, a polynomial for generation of the public key, the message and a blinding value, respectively), all of degree at most $N-1$. Along with these parameters we add some additional constraints that allow this cipher to work correctly. We refer to [47] for further details.

We start with the key generation. The private key is a pair (f, g) defined in the following way: first we choose $f \in \mathcal{L}_f$ such that is is invertible modulo p and q, i.e. over the rings

$$R_p := \mathbb{Z}_p[X]/(X^N-1) \qquad R_q := \mathbb{Z}_q[X]/(X^N-1)$$

We denote with f_p, f_q its inverses, respectively. We also choose an element $g \in \mathcal{L}_g$. The public key is h, obtained as $h := f_q \cdot g \pmod q$. The encryption of a message $m \in \mathcal{L}_m$ is done as follows: chosen a random element $\varphi \in \mathcal{L}_\varphi$, the encrypted message c is computed as $c := p\varphi \cdot h + m \pmod q$. For the decryption it is computed $a \equiv f \cdot c \pmod q$, choosing the representatives of $\mathbb{Z}_q$ in $[-q/2, q/2]$, after which the original message m is recovered as $m = f_p \cdot a \pmod p$, indeed

$$\begin{aligned} f_p \cdot a \pmod q &= f_p \cdot (f \cdot c) \pmod q \\ &= f_p \cdot (f \cdot (p\varphi h + m)) \pmod q \end{aligned}$$

Since the coefficients of a are chosen in the interval $[-q/2, q/2]$ the original message can be properly recovered: the coefficients of the message m has been chosen in the interval $[-p/2, p/2]$, and so all coefficients of $(f \cdot (p\varphi h + m))$ already lie within the interval $[-q/2, q/2]$ because the polynomials h, f, f_p, m and prime p all have coefficients that are small compared to q. This means that all coefficients are left unchanged during reduction modulo q and that

$$\begin{aligned} f_p \cdot a \pmod p &= f_p \cdot (f \cdot (p\varphi h + m)) \pmod p \\ &= f_p \cdot f \cdot m \pmod p \\ &= m \pmod p \end{aligned}$$

In this cryptosystem, searching for the private key f is equivalent to solve the SVP in the lattice defined by the matrix

$$\begin{pmatrix} \alpha & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & \alpha & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha & h_1 & h_2 & \cdots & h_0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{pmatrix}$$

where $h_0, h_1, ..., h_{N-1}$ are the coefficients of h and α is a suitable parameter.

NTRU, and more generally most lattice-based key creation algorithms, are relatively simple, efficient, and highly parallelizable, they also offer security proofs that

are not possible in other families of algorithms. However, usually, in the implementation of these ciphers, more particularly in the parameters' choice, these proofs are not taken into account, because if they were considered the parameters would become too large. The problems of this family lie in the fact that the complexity of the attacks is not yet well understood by the cryptographic community, and that attacks on these ciphers often experience dramatic improvements.

- **Code-based cryptography**: This family provides cryptosystems in which the cryptographic primitive uses an error correcting code $\mathcal{C}$. Code-based cryptosystems base their security on the difficulty of inverting a map such as

$$\begin{aligned} \{\text{Message Space}\} &\longrightarrow \{\text{Key Space}\} \\ m &\longmapsto mG + e \end{aligned}$$

  where G is a generator matrix of a code where correction of errors is very slow, and with e an error vector. The protocol requires G to be of the form $S \cdot \hat{G} \cdot P$, for some invertible matrices S, P and a secret matrix $\hat{G}$. The decoding of a word $c + e$ can be done efficiently only knowing these three matrices. Since the structure of G determines the type of code that is used by the scheme, by changing the structure of the private matrix we can use new codes, which allow different trade-off in the size of the keys and in the efficiency of the scheme. For example, the well-known MCEliece cryptographic scheme is based on Goppa's codes, so that the private key is a random binary irreducible Goppa code and the public key is a random generator matrix of a randomly permuted version of that code. First proposed in 1978, McEliece is a very efficient and safe scheme [65], just think that it has received more than 40 years of studies by cryptologists, without anyone ever being able to violate it. Although fast, this scheme suffers from very large key sizes, due to the type of codes on which it is based. To overcome the problem of large keys, the more recent variants of this scheme have introduced a more articulated structure within the codes, however, this added structure usually leads to critical attacks on these ciphers, suggesting that McEliece is still the best choice.

- **Multivariate polynomial cryptography**: These schemes are based on the difficulty of solving a system of multivariate polynomials defined on a finite field and with degree greater than 1. Several multivariate cryptosystems have been proposed over the past few decades, with many having been broken. While there have been some proposals for multivariate encryption schemes, multivariate cryptography has historically been more successful as an approach to signatures, primarily because multivariate schemes provide the shortest signature among post-quantum algorithms.

- **Hash-based cryptography**: Hash-based cryptography is an alternative post quantum cryptographic scheme that is primarily focused on digital signatures which verify that the document or message originated from the initial sender of the document. There are currently no hash-based cryptographic schemes for encrypting and decrypting messages using asymmetric public key exchange (PKE), so additional cryptographic methods would be necessary for the remaining cryptographic services.

  In this family, each cryptographic primitive bases its strength on the security of a given hash function. The concept of security for a hash function is equivalent to make some assumptions about the properties that this map possesses. In particular, for each secure hash function it is possible to create a new scheme that uses it. As

a result, each suitable map produces a different corresponding hash-based signature scheme. Even if a certain hash function becomes insecure, we can simply replace it with a different and safe one to obtain a secure instance of the hash-based signature scheme under consideration. Minimality of security assumptions is another feature of these signature schemes: generally, they only require a secure cryptographic hash function to ensure the overall security of the scheme.

- **Isogeny-based cryptography**: The last family contains algorithms which get nourishment from the deeper theory of elliptic curves, as we will see throughout this discussion. The key idea in moving from conventional elliptic-curve cryptography to isogeny-based cryptography is that points on curves are replaced with entire curves, and relationships between points (scalars and discrete logarithms) are replaced by relationships between curves (isogenies). Curves that fullfill this relation are said to belong to the same isogeny class. As we have already mentioned, isogeny classes have just enough algebraic structure to define efficient asymmetric cryptosystems, but not enough to make them vulnerable to Shor's algorithm.

  The underlying problem of this family of algorithms is the so called isogeny path problem, which can be stated as follows: given two elliptic curves E,E' defined over a finite field $\mathbb{F}_q$, and such that $|E| = |E'|$, find an isogeny φ of smooth degree that maps E to E'. This schemes use the well studied mathematics of elliptic curves to create a Diffie-Hellman like key exchange that can serve as a straightforward quantum computing resistant replacement for the DH and ECDH key exchange methods. There are three main isogeny-based algorithms: the CRS (Couveignes, Rostovtsev–Stolbunov), SIDH (Supersingular Isogeny Key Exchange), and CSIDH (Commutative Supersingular Isogeny Key Exchange). One of the advantages of this primitive family is to have very small keys compared to the others; on the other hand, the computational cost required to run each scheme is very high, and since this is a research field entirely new in cryptography, there has not been enough analysis yet for the community to have too much confidence in his safety.

1.4 Post-Quantum Cryptography Standardization

To ensure information security against quantum attacks, we will have to migrate to some of these new cryptographic schemes.

Previous transitions from weaker to stronger ciphers were based on the so called security bit paradigm, which measures the security of an algorithm based on the time complexity of a attack done by a classical computer, for example an algorithm is said to have 128 bits of security if the difficulty of attacking it with a classical computer is comparable to the time and resources required to brute-force search for a 128-bit cryptographic key. In January 2016, NIST classified the standardized algorithms into groups with 80, 112, 128, 192 and 256 bits of security [3], and defined the guidelines for the transition to more secure ciphers as follows: the family of ciphers with 80 bits of security is not considered secure enough, furthermore, 112-bit security ciphers must be phased out gradually by 2031. Unfortunately this type of paradigm does not take into account the security of algorithms against quantum cryptanalysis, therefore it is an inadequate tool to guide the transition to quantum-resistant cryptography.

With these premises, NIST in December 2016 began a process of standardization of post-

quantum algorithms announcing a public call for proposals. Since most symmetric primitives are relatively easy to modify in a way that makes them quantum resistant, efforts have focused on public-key cryptography, namely digital signatures and key encapsulation mechanisms. The competition is now in its third round out of expected four, where in each round some algorithms are discarded and others are studied more carefully. NIST hopes to publish the standardization documents by 2024, but may speed up the process if major breakthroughs in quantum computing are made. Below is the list of proposals submitted to NIST in the first round, grouped by type. The counting does not take into account candidates who have withdrawn, and if more candidates have been merged into a single scheme, only the latter has been taken into account.

Family	*Signature*	*PKE/KEM*	*Sum*
Lattice	5	23	28
Codes	3	17	20
Multivariate	7	3	10
Hash	2	0	2
Isogenies	0	1	1
Others	3	5	8

Table 1.3: NIST candidates

We informally remember the difference between *key encapsulation mechanism* (KEM) and *public key encryption* (PKE). These notions define classes of encryption techniques, and are tightly related: PKE's encryption procedure, on input plaintext m and receiver R's public-key pk_R, outputs ciphertext c, while KEM's encryption procedure, on input receiver R's public-key pk_R, outputs ciphertext c and key k, where c is sent to R, and k is kept secret inside the sender, and employed in a subsequent process of data encryption. PKE's decryption procedure, on input c and secret-key sk_R, outputs plaintext m, while KEM's decryption procedure, on input c and secret-key sk_R, outputs key k [70]. It is possible to show that these algorithms can be turned one into the other, and for this reason in the table above they are grouped in a single column.

One of the most important challenges for the NIST competition, and more generally for the post quantum world, is to find a suitable candidate to replace the Diffie-Hellman protocol, given its importance and centrality for all public-key cryptography. Nowadays this primitive is often an elementary component of some other more complicated protocol, for example, the TLS protocol, used to establish secure Internet connections, includes an ephemeral Diffie-Hellman protocol, or the X3DH protocol, used to establish connections in Signal and WhatsApp, includes four simple Diffie-Hellman instances between various short and long term key pairs. The common use of classical Diffie-Hellman therefore makes the search for a post-quantum substitute particularly relevant today. Ideally we do not want to limit ourselves to any substitute, which only achieves the same result, i.e. the exchange of a common key. We would like to have a *drop-in* replacement instead, that is a protocol so suitable to replace the old one that an outside observer does not notice the difference.

Although many promising post-quantum candidates have been developed for public key cryptography and signatures (see Table 1.3) finding a simple replacement for Diffie-Hellman,

as opposed to a KEM, has proven to be surprisingly complicated. The only post-quantum protocol truly similar to Diffie-Hellman is CSIDH, from isogeny-based cryptography. Unfortunately, this scheme was not developed until later, when the NIST process was already underway, and therefore it could not take part in the competition. We will now provide the theoretical background to fully understand this scheme.

Chapter 2

Elliptic Curves

We set out below the most relevant facts regarding elliptic curves. We start by stating some elementary results, after which we analyze the principal type of maps between curves, dwelling on those that preserve both their group and abelian variety structure. We will call these maps *isogenies*. With this new knowledge we will be able to understand further results regarding elliptic curves, which we will state in the final section. For a more detailed description, we refer to [35], [95], [85], [25] and [24].

2.1 Elementary Results

We initially provide the definition of an elliptic curve in its general form. For our purposes this definition turns out not to be very practical, so we replace it with a more usable one. However, we show that, limited to the cases of our interest, these two definitions coincide. We then define a group structure over the set of points of a generic curve, and try to characterize the elements of this group.

Definition 2.1 (Projective space)**.** Let $\mathbb{K}$ be a field and denote with $\bar{\mathbb{K}}$ its algebraic closure, the *projective space of dimension* n, denoted by $\mathbb{P}^n$ or $\mathbb{P}^n(\bar{\mathbb{K}})$, is the set of all $(n+1)$-tuples

$$(x_0, \dots, x_n) \in \bar{\mathbb{K}}^{n+1}$$

such that $(x_0, \dots, x_n) \neq (0, \dots, 0)$, taken modulo the relation $R \subseteq \bar{\mathbb{K}}^{n+1} \times \bar{\mathbb{K}}^{n+1}$ defined by

$$\big((x_0, \dots, x_n), (y_0, \dots, y_n)\big) \in R \iff \exists \lambda \in \bar{\mathbb{K}} \text{ such that } x_i = \lambda y_i \text{ for all } i.$$

It is easy to prove that R is an equivalence relation. We will denote the equivalence class of a projective point $(x_0, \dots, x_n)$ with $(X_0 : \cdots : X_n)$. The set of the $\mathbb{K}$-*rational points*, denoted by $\mathbb{P}^n(\mathbb{K})$, is defined as

$$\mathbb{P}^n(\mathbb{K}) = \{(X_0 : \dots : X_n) \in \mathbb{P}^n \mid X_i \in \mathbb{K} \text{ for all } i\}.$$

By fixing arbitrarily the coordinate $X_n = 0$, we define a projective space of dimension $n-1$, which we call the *hyperplane at infinity*; its points are called *points at infinity*.

Formally, elliptic curves are projective curves of genus 1 with a distinguished point. From now on we suppose that the field $\mathbb{K}$ has characteristic different from 2 and 3. This has the merit of greatly simplifying the representation of an elliptic curve.

Definition 2.2 (Weierstrass equation)**.** Let $\mathbb{K}$ be a field. An *elliptic curve* defined over $\mathbb{K}$ is the locus in $\mathbb{P}^2(\bar{\mathbb{K}})$ of an equation

$$Y^2Z = X^3 + AXZ^2 + BZ^3 \tag{2.1}$$

with $A, B \in \mathbb{K}$ and $4A^3 + 27B^2 \neq 0$. The point $O := (0 : 1 : 0)$ is the only point on the line $Z = 0$; it is called the *point at infinity* of the curve.

In some ambiguous cases, to avoid confusion, we will also add to the point at infinity the curve to which this element refers. For example, for two elliptic curves E_1, E_2 their points at infinity will be denoted respectively as O_{E_1}, O_{E_2}.

It is customary to write Eq. (2.1) in *affine form*: by defining the coordinates $x = X/Z$ and $y = Y/Z$, we equivalently define the elliptic curve as the locus of the equation

$$y^2 = x^3 + Ax + B$$

plus the point at infinity O. In characteristic different from 2 and 3, we can show that any projective curve of genus 1 with a distinguished point O is isomorphic to a Weierstrass equation by sending O onto the point at infinity $(0 : 1 : 0)$.

If we want to consider points with coordinates in some extension field $\mathbb{K}' \supseteq \mathbb{K}$ we write $E(\mathbb{K}')$, referring to the following definition

$$E(\mathbb{K}') := \{(x, y) \in \mathbb{K}' \times \mathbb{K}' \mid y^2 = x^3 + Ax + B\} \cup \{O\}.$$

In order to count the number of points on a field $\mathbb{K}'$ that satisfy the relation defined by the elliptic curve, we clearly just have to compute the cardinality of $E(\mathbb{K}')$. For example, we can consider the elliptic curve $E/\mathbb{F}_{11}$ defined by the equation $y^2 = x^3 - 8x - 1$. In this case $E(\mathbb{F}_{11})$ consists of the neutral element O and all the points emphasized in Figure 2.1.

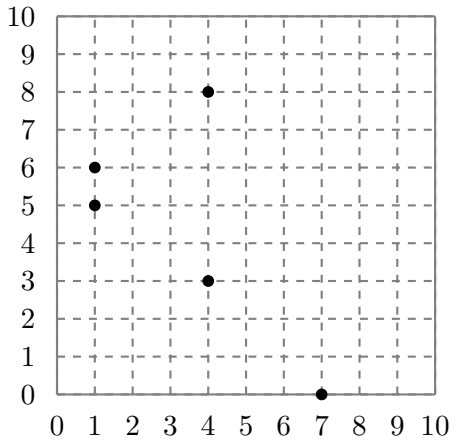


Figure 2.1: $E(\mathbb{F}_{11}) \setminus \{O\}$, with $E/\mathbb{F}_{11} := \left(y^2 = x^3 - 8x - 1\right)$.

It is possible to informally define a binary operation on the points of an elliptic curve through the well-known chord-tangent law, which we briefly recall for completeness.

Definition 2.3 (Chord-tangent law)**.** The chord-tangent composition of two points P and Q of an elliptic curve E is denoted by $P + Q$ and is defined as the opposite of the third point of intersection of the line through P and Q with E. If P and Q are the same point, the line through them is given by the tangent to the curve at P.

The point $P + Q$ determined by this definition is easily shown to exist and may be determined through algebraic manipulations of the elliptic curve equation. Its coordinates can be expressed in terms of the coordinates of P and Q and the coefficients of E. It is useful to take note of a geometric special case. When one of the points involved is O, which can not be represented graphically, we just draw a vertical line through the other point, and check for intersection with E. Consider in this regard the curve $y^2 = x^3 - 8x - 1$ defined over $\mathbb{R}$. The sum operation described above can be given geometrically as illustrated in Figure 2.2.

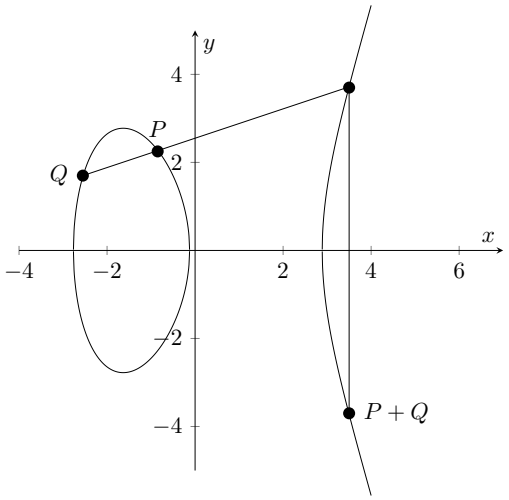


Figure 2.2: $E(\mathbb{R}) \setminus \{O\}$, with $E/\mathbb{R} := \left(y^2 = x^3 - 8x - 1\right)$.

The transposition of this operation to curves defined over finite fields remains the same at the level of definition, but the geometric intuition is faded. In this regard, compare Figure 2.1 and Figure 2.2, where the same curve, defined on different fields, takes a radically different appearance. Figure 2.3 shows the chord-tangent construction in the finite field case.

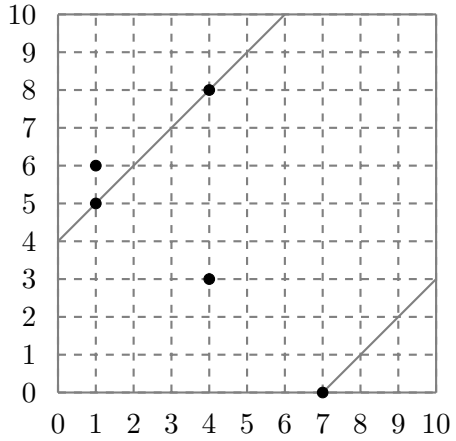


Figure 2.3: $E(\mathbb{F}_{11}) \setminus \{O\}$, with $E/\mathbb{F}_{11} := \left(y^2 = x^3 - 8x - 1\right)$.

In the example just shown, we see how the sum of $P := (1, 5)$ and $Q := (4, 8)$ gives $(0, 7)$ as a result. This operation, defined in a purely intuitive and geometric way, finds in its algebraic formalization the justification of all the properties it possesses, as shown hereafter.

Let $E/\mathbb{F}_q$ be an elliptic curve defined by $y^2 = x^3 + Ax + B$. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on E with $P_1, P_2 \neq O$. Define $P_1 + P_2 = P_3 = (x_3, y_3)$ as follows:

- If $x_1 \neq x_2$,then

$$(x_3, y_3) = \left(\left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2, \left(\frac{y_2 - y_1}{x_2 - x_1} \right)(x_1 - x_3) - y_1 \right).$$

- If $x_1 = x_2$ but $y_1 \neq y_2$,then $P_1 + P_2 = O$.
- If $P_1 = P_2$ and $y_1 \neq 0$, then

$$(x_3, y_3) = \left(\left(\frac{3x_1^2 + A}{2y_1} \right)^2 - 2x_1, \left(\frac{3x_1^2 + A}{2y_1} \right)(x_1 - x_3) - y_1 \right).$$

- If $P_1 = P_2$ and $y_1 = 0$, then $P_1 + P_2 = O$.

Moreover, define

$$P + O = O + P = P$$

for all points P on E.

It is possible to prove that $(E(\mathbb{K}), +, O)$ forms a group with respect to the sum previously defined, with O as a neutral element.

Remark 2.4. It is a known fact in literature that it is possible to define a group law not only on elliptic curves, but also on any irreducible cubic or on a conic. The way in which two points are added is slightly different from what happens with elliptic curves. We start by defining the sum of two points on a conic defined on an infinite field, more precisely the field $\mathbb{R}$ of real numbers, however, the result we obtain is also valid for any other field. Let Γ be a non-degenerate conic. Since, unlike what happens with elliptic curves, we do not have an immediate candidate which plays the role of neutral element with respect to the sum, we fix a generic point $N \in \Gamma$, which will fit the role. We then define the sum operator $+ : \Gamma \times \Gamma \to \Gamma$ in the following manner: given $P, Q \in \Gamma$, we consider the line passing through these points, we then trace a straight line through N, parallel to the one obtained above. This last line must intersect Γ at another point, called R. We define R as the result of the operation. Figure 2.4 illustrates the operation.

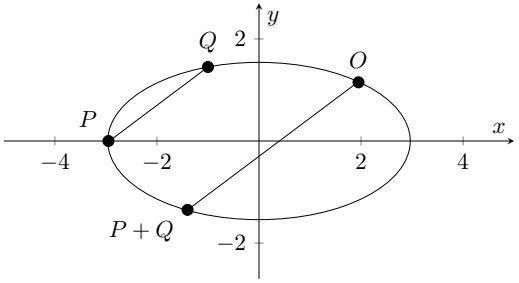


Figure 2.4: $\Gamma(\mathbb{R})$, with $\Gamma/\mathbb{R} := \left(\left(\frac{2}{5}\right)^2 x^2 + \left(\frac{10}{13}\right)^2 y^2 = 1 \right)$.

As with elliptic curves, if $P = Q$, then the line for these two points is the tangent to Γ in P, and therefore to compute the sum $P + P$, it is sufficient to draw the line passing

through N and parallel to the tangent to Γ in P; the second point of intersection with Γ will therefore be $P + P$. Similarly, if the line for N, parallel to that for P and Q, is tangent to Γ in N, we take as sum $P + Q = N$. We also have $N + N = N$ and $P + N = P$ for all $P \in \Gamma$. We refer to [83] for further details.

If the definition field of our conic is a finite field, the group associated with it is cyclic. This fact allows us to use this structure in cryptographic applications. The cases of application are typically limited to those protocols involving cyclic groups, more specifically those for which it is necessary to compute the product $n \cdot P$ of an element P of the group. Using conics we do not avoid this computation, however the computational cost may be significantly smaller, while maintaining the same resistance against external attacks. Unlike elliptic curves, the product $n \cdot P$ is defined in the same way for all points of the curve, and the result of this operation is still a proper point of the latter. A sketch of an RSA-like cryptosystem on general conics is given in [4].

The fact that points on an elliptic curve form an abelian group is behind most of the interesting properties and applications. We will make extensive use of the group structure defined above.

Definition 2.5 (n-torsion subgroup)**.** Let E be an elliptic curve over a field $\mathbb{K}$ and let $P \in E(\mathbb{K})$. The *n-torsion subgroup* is

$$E[n] := \{P \in E(\overline{\mathbb{K}}) : [n]P = O\}$$

Example 2.6. Let us consider again the elliptic curve defined in Figure 2.1. In Table 2.1 we summarize the order of each point.

Point	Order
O	1
$(1, 5)$	3
$(1, 6)$	3
$(4, 3)$	6
$(4, 8)$	6
$(7, 0)$	2

Table 2.1: Points (and respectively orders) of $E/\mathbb{F}_{11} := \left(y^2 = x^3 - 8x - 1\right)$.

It follows that the interesting torsion subgroups are given by

$$\begin{aligned}
E[1] &= \{O\} \\
E[2] &= \{O, (7, 0)\} \\
E[3] &= \{O, (1, 5), (1, 6)\} \\
E[6] &= \{O, (7, 0), (1, 5), (1, 6), (4, 3), (4, 8)\} \\
E[0] &= \{O, (7, 0), (1, 5), (1, 6), (4, 3), (4, 8)\}
\end{aligned}$$

More generally, for every elliptic curve E we have that $E[0] = E$ and $E[1] = \{O\}$.

Proposition 2.7. *Let $\mathbb{K}$ be an algebraically closed field of characteristic p, let E be an elliptic curve defined over $\mathbb{K}$, and let $m \neq 0$ be an integer. The m-torsion group of E, denoted by $E[m]$, has the following structure:*

- *if p does not divide m, then $E[m] \cong \mathbb{Z}_m \times \mathbb{Z}_m$.*
- *If p divides m, then*

$$E[p^n] \cong \begin{cases} \mathbb{Z}_{p^n} & \text{for any } n \geq 0, \text{ or} \\ \{O\} & \text{for any } n \geq 0. \end{cases}$$

Proof. See [85, Cor. 6.4]. □

For curves defined over a field of positive characteristic p, the case $E[p] \cong \mathbb{Z}/p\mathbb{Z}$ is called *ordinary*, while the case $E[p] \cong \{O\}$ is called *supersingular*. We shall see later some alternative characterizations of supersingular curves.

2.2 Maps Between Elliptic Curves

Finally, we focus on different type of maps between elliptic curves. As we said, we are mostly interested in isogenies. We give particular emphasis to isogenies defined from a curve to itself. These maps will be called endomorphisms. We study how this set of functions can be equipped with a ring structure, and we characterize the structure of this ring.

Morphisms

Definition 2.8 (Function field)**.** Let $\mathbb{K}$ be a field and $C/\mathbb{K}$ a projective curve defined by $f(x, y, z) = 0$, where $f \in \mathbb{K}[x, y, z]$ is irreducible[1] in $\overline{\mathbb{K}}[x, y, z]$. The *function field* of C consists of rational functions g/h such that the following conditions hold:

- g and h are homogeneous elements of $\mathbb{K}[x, y, z]$ of the same degree.
- h does not lie in the ideal (f).
- the functions g_1/h_1 and g_2/h_2 are considered equivalent whenever $g_1h_2 - g_2h_1 \in (f)$.

The function field of C is denoted $\mathbb{K}(C)$, which should not to be confused with $C(\mathbb{K})$, the set of $\mathbb{K}$-rational points on C. The fact that f is irreducible and $\mathbb{K}[x, y, z]$ is a unique factorization domain (so every irreducible element is prime) makes it clear that $\mathbb{K}(C)$ is, in fact, a field. The field $\overline{\mathbb{K}}(C)$ is defined analogously, with $g, h \in \overline{\mathbb{K}}[x, y, z]$. Alternatively, if C is defined by an affine equation $f(x, y) = 0$, one can define $\mathbb{K}(C)$ as the fraction field of the ring $\mathbb{K}[x, y]/(f)$; in this case the degree of the function $r = g/h$ is $max\{\deg g, \deg h\}$. We can now define a rational map.

Definition 2.9 (Rational map)**.** Let $\mathbb{K}$ be a field and let C_1, C_2 be projective curves defined over $\mathbb{K}$. A *rational map* $\varphi : C_1(\overline{\mathbb{K}}) \to C_2(\overline{\mathbb{K}})$ is a map of the form $(\varphi_x : \varphi_y : \varphi_z)$, with $\varphi_x, \varphi_y, \varphi_z \in \overline{\mathbb{K}}(C_1)$, such that for every point $P \in C_1(\overline{\mathbb{K}})$ where φ_x, φ_y and φ_z are all defined, the point $(\varphi_x(P) : \varphi_y(P) : \varphi_z(P))$ lies in $C_2(\overline{\mathbb{K}})$.

What exactly does it mean for a map, for example φ_x, to be defined at a point? The concept is clear when we consider functions in a space like $\mathbb{K}[x]$, but we are now working in a quotient space. We observe in fact that $\varphi = (\varphi_x : \varphi_y : \varphi_z)$ is defined only up to scalar equivalence: for any $\lambda \in \overline{\mathbb{K}}$ the triple $(\lambda\varphi_x : \lambda\varphi_y : \lambda\varphi_z)$ defines exactly the same rational map φ. There may be points $P \in C_1(\overline{\mathbb{K}})$ where one of φ_x, φ_y, or φ_z is not defined, but in this case it may still be possible to evaluate the map φ at P after re-scaling φ by an element of $\mathbb{K}(C)$.

[1] It can not be factored into the product of two non-constant polynomials with coefficients in $\overline{\mathbb{K}}$.

Definition 2.10 (Regularity at P)**.** In the notation of the previous definition, we say that $\varphi : C_1(\overline{\mathbb{K}}) \to C_2(\overline{\mathbb{K}})$ is *defined* (or *regular*) at a point $P \in C_1(\overline{\mathbb{K}})$ if there exists a function $g \in \overline{\mathbb{K}}(C_1)$ such that $g\varphi_x$, $g\varphi_y$, $g\varphi_z$ are all defined at P and at least one is nonzero at P. We use $g\varphi$ to denote the map $(g\varphi_x : g\varphi_y : g\varphi_z)$.

There may be many choices of representative for the equivalence class of φ_x, and only some of them may be defined at P, as the following example shows.

Example 2.11. Let $\mathbb{K}$ be a field of characteristic not equal to 2. Let C be the elliptic curve defined by the algebraic set $V(y^2 - x(x-1)(x+1)) \subseteq A^2(\mathbb{K})$. Consider the functions

$$f_1 := \frac{x(x-1)}{y} \qquad \text{and} \qquad f_2 := \frac{y}{x+1}$$

We can check that f_1 is equivalent to f_2 . Note that f_1 is not defined at $(0,0), (1,0)$ or $(-1,0)$ while f_2 is defined at $(0,0)$ and $(1,0)$ but not at $(-1,0)$. The equivalence class of f_1 is therefore regular at $(0,0)$ and $(1,0)$.

Definition 2.12 (Morphism)**.** A rational map that is defined everywhere is called a *morphism.*

For elliptic curves, distinguishing rational maps from morphisms is unnecessary: every rational map between elliptic curves is a morphism. More generally, we have the following.

Theorem 2.13. *If C_1 is a smooth projective curve then every rational map from C_1 to a projective curve C_2 is a morphism.*

Proof. See [85, II, Prop. 2.1]. □

Remark 2.14. We observe that in general a morphism between elliptic curves does not respect the group structure of these curves: suppose E an elliptic curve over a field $\mathbb{K}$, and take $Q \in E(\mathbb{K})$. We define the translation map

$$\begin{aligned} \tau_Q : E(\overline{\mathbb{K}}) &\longrightarrow E(\overline{\mathbb{K}}) \\ P &\longmapsto P + Q \end{aligned}$$

Clearly, τ_Q is a rational map that is defined everywhere on E and so it is a morphism. However, this map does not respect the group structure of its domain and co-domain, in fact, for example, $O \mapsto Q$.

Isogenies

Elliptic curves have both an algebraic structure, as an abelian group, and a geometric structure, as an algebraic curve. We now introduce isogenies, which are maps that respect both the algebraic and the geometric structure of these curves.

Definition 2.15 (Isogeny)**.** Let $\mathbb{K}$ be a field, and let E, E' be elliptic curves defined over $\mathbb{K}$. An *isogeny* is a morphism $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ such that $\varphi(O_E) = O_{E'}$. We call the *zero isogeny* the constant map $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ given by $\varphi(P) = O_{E'}$ for all $P \in E(\overline{\mathbb{K}})$.

Unless otherwise stated, we assume that the isogeny φ is itself defined over $\mathbb{K}$ (meaning that it can be represented by a rational map whose coefficients lie in $\mathbb{K}$). Isogenies can be characterized in different ways, all of which are equivalent. We have defined them in the way that will be more natural to use later, but we also list other definitions that will come in handy in other situations.

Theorem 2.16. *Let $\mathbb{K}$ be a field, and let E, E' be two elliptic curves defined over $\mathbb{K}$. Let $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ be a map between them. The following are equivalent:*

- *φ is a non-constant isogeny.*
- *φ is a surjective group morphism.*
- *φ is a group morphism with finite kernel.*

Proof. See [85, III]. □

Two curves are called *isogenous* if there exists an isogeny between them. We shall see later that this is an equivalence relation. We now look more in detail at isogenies of elliptic curves. We start with some basic definitions.

Definition 2.17 (Degree, separability)**.** Let $\mathbb{K}$ be a field, and let $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ be an isogeny defined over $\mathbb{K}$. Let $\mathbb{K}(E), \mathbb{K}(E')$ be the function fields of E, E'. By composing φ with the functions of $\mathbb{K}(E')$, we obtain a sub-field of $\mathbb{K}(E)$ that we denote by $\varphi^*(\mathbb{K}(E'))$.

1. The *degree* of φ is defined as $\deg \varphi = [\mathbb{K}(E) : \varphi^*(\mathbb{K}(E'))]$.
2. φ is said to be *separable*, *inseparable*, or *purely inseparable* if the extension of function fields is respectively separable, inseparable or purely inseparable.

Proposition 2.18. *In the same notation as above*

1. *$\deg \varphi$ is always finite.*
2. *If φ is separable, then $\deg \varphi = |\ker \varphi|$.*
3. *If φ is purely inseparable, then $\deg \varphi$ is a power of the characteristic of $\mathbb{K}$.*
4. *Any isogeny can be decomposed as a product of a separable and a purely inseparable isogeny.*

Proof. See [85, II]. □

In practice, most of the time we will be considering separable isogenies, and we can take $\deg \varphi = |\ker \varphi|$ as the definition of the degree. Notice that in this case $\deg \varphi$ is the size of any fiber[2] of φ.

Example 2.19. Consider the map φ from $E : y^2 = x^3 + x$ to $E' : y^2 = x^3 - 4x$ defined by

$$\begin{aligned} \varphi(x, y) &= \left(\frac{x^2 + 1}{x}, y \frac{x^2 - 1}{x^2} \right), \\ \varphi(0, 0) &= \varphi(O_E) = O_{E'}. \end{aligned} \tag{2.2}$$

This is a separable isogeny between curves defined over $\mathbb{Q}$. It has degree 2, and its kernel is generated by the point $(0, 0)$. Plotting the isogeny (2.2) over $\mathbb{R}$ would be cumbersome, however, since the curves are defined by integer coefficients, we may reduce the equations modulo a prime p, then the isogeny descends to an isogeny of curves over $\mathbb{F}_p$. Figure 2.5 plots the action of the isogeny after reduction modulo 11.

[2] That is, the preimage of any element of the codomain.

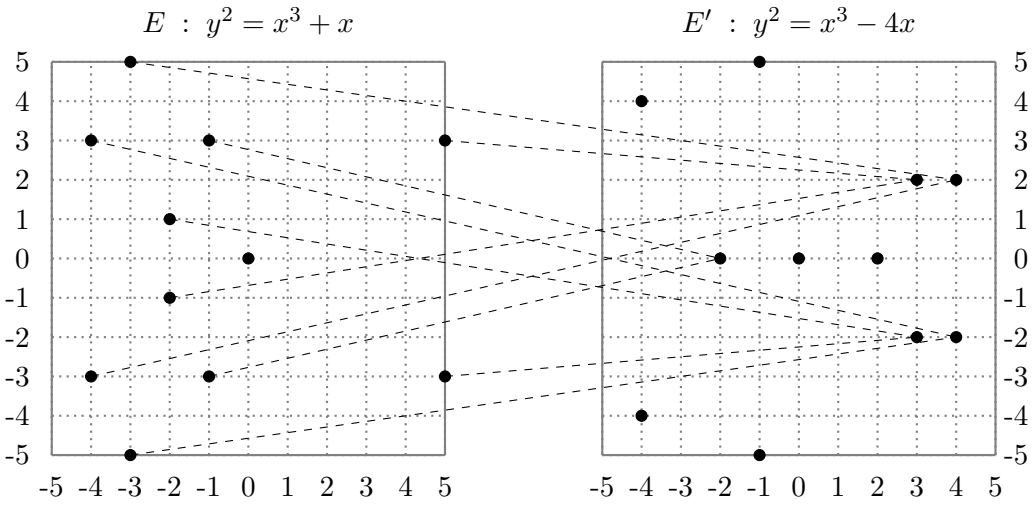


Figure 2.5: The isogeny $(x, y) \mapsto \big((x^2+1)/x,\ y(x^2-1)/x^2\big)$, as a map between curves defined over $\mathbb{F}_{11}$.

A dashed arrow indicates that a point of the left curve is sent onto a point on the right curve; the action on the point in $(0, 0)$, going to the point at infinity, is not shown. We observe a symmetry with respect to the x-axis, a consequence of the fact that φ is a group morphism. By looking closer, we may also notice that collinear points are sent to collinear points, otherwise said, opposite points are sent to opposite points, which is clearly a necessity for a group morphism.

It is evident that the isogeny is 2-to-1, however we are unable to see all fibers over $\mathbb{F}_p$, because the isogeny is only surjective over the algebraic closure. This is not dissimilar from the way power-by-n maps act on the multiplicative group $\mathbb{K}^*$ of a field $\mathbb{K}$: the map $x \mapsto x^2$, for example, is a 2-to-1 (algebraic) group morphism on $\mathbb{F}_{11}^*$, and those elements that have no pre-image, the non-squares, will have exactly two square roots in $\mathbb{F}_{11^2}$, and so on.

The most unique property of separable isogenies is that they are entirely determined by their kernel.

Proposition 2.20. *Let $\mathbb{K}$ be a field, and let E be an elliptic curve defined over $\overline{\mathbb{K}}$. Let G be a finite subgroup of E. There is a curve E', and a separable isogeny φ, such that $\ker\varphi = G$ and $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$. Furthermore, E' and φ are unique up to composition with an isomorphism.*

Proof. See [85, III, Prop. 4.12]. □

That is, for any finite subgroup $G \subseteq E$ we have an exact sequence[3] of algebraic groups

$$0 \to G \to E \xrightarrow{\varphi} E' \to 0.$$

Uniqueness up to isomorphisms justifies the notation E/G for the isomorphism class of the image curve E'. Conversely, since for Prop. 2.16 any non-constant isogeny of elliptic curves necessarily has finite kernel, we have a bijection between the finite subgroups of a curve

[3] An exact sequence is a sequence of groups and group homomorphisms where the image of each homomorphism is equal to the kernel of the next.

E and the isogenies with domain E up to isomorphisms. The computational counterpart to the kernel-isogeny correspondence is given by Vélu's much celebrated formulæ.

Proposition 2.21. *Let $\mathbb{K}$ be a field, and let $E : y^2 = x^3+ax+b$ be an elliptic curve defined over $\mathbb{K}$. Let $G \subseteq E(\overline{\mathbb{K}})$ be a finite subgroup. The separable isogeny $\varphi : E(\overline{\mathbb{K}}) \to E/G(\overline{\mathbb{K}})$, with kernel G, can be written as*

$$\varphi(P) = \left(x(P) + \sum_{Q \in G\setminus\{O\}} \Big(x(P+Q) - x(Q) \Big), y(P) + \sum_{Q \in G\setminus\{O\}} \Big(y(P+Q) - y(Q) \Big) \right);$$

and the curve E/G has equation $y^2 = x^3 + a'x + b'$, where

$$a' = a - 5 \sum_{Q \in G\setminus\{O\}} (3x(Q)^2 + a),$$
$$b' = b - 7 \sum_{Q \in G\setminus\{O\}} (5x(Q)^3 + 3ax(Q) + 2b).$$

Proof. See [93]. □

In other words, on input (the curve constants defining) E and the points in G, these formulæ output the constants defining E/G and the explicit maps for φ, i.e. the maps that move any points on E (except those in the kernel G) to their corresponding image on E/G. We refer to [20, Pag. 6-7] for some basic application cases. This correspondence between kernels and isogenies is rich in consequences: the following useful facts are fairly straightforward.

Corollary 2.22. *Any isogeny of elliptic curves can be decomposed as a product of prime degree isogenies.*

Corollary 2.23. *Let E be defined over an algebraically closed field $\mathbb{K}$, let l be a prime different from the characteristic of $\mathbb{K}$, then there are exactly $l+1$ isogenies of degree l with domain E, up to isomorphism.*

In particular, the last result follows immediately from that relation $E[l] \cong \mathbb{Z}_l \times \mathbb{Z}_l$, which has exactly $l+1$ subgroups of order l, and they are all such subgroups. Slightly more work is required to prove the following, fundamental, theorem.

Theorem 2.24 (Dual isogeny theorem)**.** *Let $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ be an isogeny of degree m. There is a unique isogeny $\hat{\varphi} : E' \to E$ such that*

$$\hat{\varphi} \circ \varphi = [m]_E, \quad \varphi \circ \hat{\varphi} = [m]_{E'}.$$

$\hat{\varphi}$ is called the dual isogeny of *φ; it has the following properties:*

1. *$\hat{\varphi}$ has degree m.*
2. *$\hat{\varphi}$ is defined over $\mathbb{K}$ if and only if φ is.*
3. *$\widehat{\psi \circ \varphi} = \hat{\varphi} \circ \hat{\psi}$ for any isogeny $\psi : E' \to E''$.*
4. *$\widehat{\psi + \varphi} = \hat{\psi} + \hat{\varphi}$ for any isogeny $\psi : E \to E'$.*
5. *$\deg \varphi = \deg \hat{\varphi}$.*
6. *$\hat{\hat{\varphi}} = \varphi$.*

Proof. See [85, III.6]. □

Note that being isogenous is an equivalence relation: reflexivity and transitivity are obvious, while symmetry is guaranteed by the dual isogeny theorem.

Isomorphisms

Two projective curves C_1 and C_2 are isomorphic if they are related by an invertible morphism φ; this means that there is a morphism φ^{-1} such that $\varphi^{-1} \circ \varphi$ and $\varphi \circ \varphi^{-1}$ are the identity maps on $C_1(\overline{\mathbb{K}})$ and $C_2(\overline{\mathbb{K}})$, respectively. If we admit the same definition also for elliptic curves, we would have, for example, that the Q-translation morphism $\tau_Q : E \to E$ described by Remark 2.14 would have as inverse τ_{-Q} and would therefore be an isomorphism. We would like these maps, which do not preserve the group structure, not to be said to be isomorphisms. For elliptic curves we have a stronger notion of isomorphism, since we also require the corresponding abelian groups to be isomorphic through φ; this means that the identity element must be preserved.

Definition 2.25 (Isomorphism)**.** Let $\mathbb{K}$ be a field, and let (E, O_E) and $(E', O_{E'})$ be elliptic curves over $\mathbb{K}$. An *isomorphism* of elliptic curves $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ is an isomorphism over $\overline{\mathbb{K}}$ of algebraic varieties such that $\varphi(O_E) = O_{E'}$. Equivalently, an isomorphism of elliptic curves is an invertible isogeny. If there is an isomorphism from E to E' then we write $E \cong E'$.

Isomorphism classes are traditionally encoded by an invariant, whose origins can be traced back to complex analysis.

Proposition 2.26 (j-invariant)**.** *Let $\mathbb{K}$ be a field, and let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{K}$. Define the* j-invariant *of E as*

$$j(E) = 1728\frac{4a^3}{4a^3 + 27b^2}.$$

Two curves are isomorphic over the algebraic closure $\overline{\mathbb{K}}$ if and only if they have the same j-invariant.

Proof. See [85, III, Prop. 1.4]. □

Definition 2.27 (Automorphism)**.** Let $\mathbb{K}$ be a field. An *automorphism* of an elliptic curve $E/\mathbb{K}$ is an isomorphism from E to itself. The group of all automorphisms of E that are defined over a field $\mathbb{K}'$ is denoted by $\mathrm{Aut}_{\mathbb{K}'}(E)$.

Endomorphisms

Finally, we note the special case of an isogeny $\alpha : E(\overline{\mathbb{K}}) \to E(\overline{\mathbb{K}})$ from an elliptic curve to itself; this is called an *endomorphism*. Since they are algebraic group morphisms, we can define addition of endomorphisms by $(\varphi + \psi)(P) := \varphi(P) + \psi(P)$, and the resulting map is still an endomorphism. Thus, by identifying the constant map that sends every point to the point at infinity with the neutral element, the set of endomorphisms forms a group. Additionally, we can equip this structure with a binary operation, the composition of endomorphisms, and verify that it distributes over addition, hence the set of all endomorphisms from curve E to itself forms a ring, denoted by $\mathrm{End}(E)$.

Example 2.28. The prototypical endomorphism is the multiplication-by-m endomorphism defined by

$$\begin{aligned} [m] : E &\longrightarrow E \\ P &\longmapsto [m]P \end{aligned}$$

This is obviously a group homomorphism, and it is also a rational map. Its kernel is exactly the m-th torsion subgroup $E[m]$. For convenience, in the following we describe

the multiplication-by-2 and multiplication-by-3 maps. We use an elliptic curve written in Montgomery form. The arithmetic of the group $E(\mathbb{K})$ obviously readjusts to this definition. This form of writing is the one that is most frequently used in applications and the CSIDH protocol is no exception. Consider the multiplication-by-2 or point doubling map on a fixed Montgomery curve $E_A : y^2 = x^3 + Ax^2 + x$, written as

$$\begin{aligned} [2] : E_A &\longrightarrow E_A \\ x &\longmapsto \tfrac{(x^2-1)^2}{4x(x^2+Ax+1)} \end{aligned}$$

In the following we take into account only the x-coordinate of the map, since it is sufficient to study its behavior and discover which points are sent to O. Observe that the doubling map has a denominator that creates exceptional points. Viewing the curve equation, we see that these are the three points with $y = 0$, namely $(0, 0), (\alpha, 0)$ and $(1/\alpha, 0)$, where $\alpha^2 + A\alpha + 1 = 0$. Indeed, these are the three points of order 2 on E_A , and together with the neutral element, O, they are the entire kernel of the doubling map. This kernel forms a subgroup of the points in E_A , with group structure

$$\ker([2]) \cong \mathbb{Z}_2 \times \mathbb{Z}_2$$

i.e. the 2-torsion is precisely three cyclic subgroups of order 2. Each subgroup has one of the three points of exact order 2, together with the identity element O (see Figure 2.6). Now consider the multiplication-by-3 or point tripling map on the elliptic curve $E_A : y^2 = x^3 + Ax^2 + x$, written as

$$\begin{aligned} [3] : E_A &\longrightarrow E_A \\ x &\longmapsto \tfrac{x(x^4-6x^2-4Ax^3-3)^2}{(3x^4+4Ax^3+6x^2-1)^2} \end{aligned}$$

Again, the denominator will give rise to exceptional points to the tripling map. Suppose its four roots are $\beta, \gamma, \zeta, \theta$; each of these correspond to x-coordinates of points of order 3 in E_A , and this time there are two (non-zero) y-coordinates for each such x. Together with O, there are then 9 points that are sent to O under [3], and this time we have

$$\ker([3]) \cong \mathbb{Z}_3 \times \mathbb{Z}_3$$

i.e. the 3-torsion is made up by four cyclic subgroups of order 3 (see Figure 2.7).

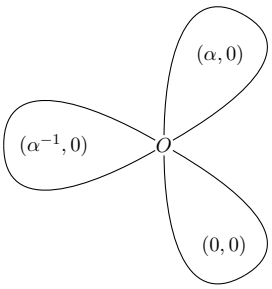


Figure 2.6: $\ker([2]) \cong \mathbb{Z}_2 \times Z_2$. Three cyclic subgroups of order 2.

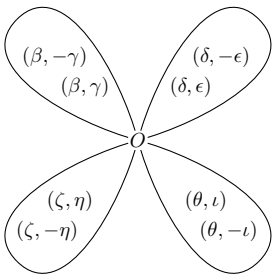


Figure 2.7: $\ker([3]) \cong \mathbb{Z}_3 \times \mathbb{Z}_3$. Four cyclic subgroups of order 3.

Example 2.29. Given an elliptic curve $E/\mathbb{F}_q$, we consider the set of points defined by $E(\overline{\mathbb{F}_q})$. The Frobenius endomorphism of the extension $E(\overline{\mathbb{F}_q})/E(\mathbb{F}_q)$ is the map

$$\varphi : E(\overline{\mathbb{F}_q}) \to E(\overline{\mathbb{F}_q})$$

defined by

$$\begin{cases} (x, y) & \longmapsto (x^q, y^q) \\ O & \longmapsto \quad O \end{cases}$$

It is immediate to verify that this map is an endomorphism, so that its name is justified. Furthermore, since this is a morphism, it can be easily shown that it preserves the order of the points, that is $|\, P \,| = |\, \varphi(P) \,|$.

The Frobenius endomorphism plays a very important role in the theory of elliptic curves, as we will see in the next section.

Proposition 2.30. *Let E and E' be elliptic curves over $\mathbb{K}$. If $n \in \mathbb{N} \setminus \{0\}$ then $[n]$ is not the zero isogeny. Furthermore, $Hom(E, E')$ is a torsion-free $\mathbb{Z}$-module (i.e. if $\varphi \in Hom(E, E')$ is non-zero then $[n] \circ \varphi$ is non-zero for all $n \in \mathbb{Z} \setminus \{0\}$), and $\mathrm{End}(E)$ has no zero divisors.*

Proof. See [35, Lem. 9.6.11]. □

We shall now give a complete characterization of the endomorphism ring for any elliptic curve. Since $\mathrm{End}(E)$ is a torsion-free $\mathbb{Z}$-module, each $m \in \mathbb{Z}$ defines a different multiplication-by-m endomorphism, indeed suppose $[n] = [m]$ for some $n, m \in \mathbb{N}$. Then $[n] - [m] = [0]$, so that $[n - m] = [0]$. Since we have no torsion point on $\mathrm{End}(E)$ we conclude that $n - m = 0$, i.e. $n = m$. It follows that the map $\mathbb{Z} \to \mathrm{End}(E)$ sending n to $[n]$ is injective and we can simply view $\mathbb{Z}$ as a sub-ring of $\mathrm{End}(E)$. But could $\mathrm{End}(E)$ be larger?

Definition 2.31 (Algebra)**.** Let $\mathbb{K}$ be a field. An *algebra* A over $\mathbb{K}$ (equivalently, a $\mathbb{K}$-algebra A) is a vector space over $\mathbb{K}$ equipped with an additional binary bi-linear operation $\cdot : A \times A \to A$.

Definition 2.32 (Order)**.** Let K be a finitely generated $\mathbb{Q}$-algebra. An *order* $\mathcal{O} \subseteq K$ is a sub-ring of K that is a finitely generated $\mathbb{Z}$-module, and that contains a $\mathbb{Q}$-basis for K.

Definition 2.33 (Quadratic number field)**.** A *quadratic number field* is a quadratic extension K of the rationals; it is called *real* if there exists an embedding $K \subseteq \mathbb{R}$, *imaginary* otherwise.

All such fields can be expressed as $\mathbb{Q}(\sqrt{d})$ for some square-free integer d. The field of *Gaussian numbers* $\mathbb{Q}(\sqrt{-1}) = \mathbb{Q}(i)$, whose ring of integers is $\mathbb{Z}[i]$ (also called the ring of Gaussian integers), is an example of an imaginary one.

Definition 2.34 (Quaternion algebra)**.** A *quaternion algebra* is an algebra of the form

$$K = \mathbb{Q} + \alpha\mathbb{Q} + \beta\mathbb{Q} + \alpha\beta\mathbb{Q},$$

where the generators satisfy the relations

$$\alpha^2, \beta^2 \in \mathbb{Q}, \quad \alpha^2 < 0, \quad \beta^2 < 0, \quad \beta\alpha = -\alpha\beta.$$

Now we have all the necessary bases to state and understand the following important result, which highlights the structure of the endomorphism ring of an elliptic curve.

Theorem 2.35 (Deuring)**.** *Let $\mathbb{K}$ be a field of characteristic p, and let E be an elliptic curve defined over $\mathbb{K}$. The ring* $\mathrm{End}(E)$ *is isomorphic to one of the following:*

- *The integer ring $\mathbb{Z}$.*
- *An order $\mathcal{O}$ in a quadratic imaginary field. In this case we say that E has complex multiplication by $\mathcal{O}$;*
- *A maximal order in a quaternion algebra ramified at p and ∞.*

Proof. See [85, III, Cor. 9.4] and [57]. □

We will observe how regular elliptic curves always have complex complication, while supersingular ones can have both complex or quaternion multiplication, depending on the field on which they are defined, and so they are generally not covered by the theory of complex multiplication. In general we know that the endomorphism ring of an elliptic curve over $\mathbb{F}_q$ is an order in a division algebra A. Depending on the number of $\mathbb{F}_q$-rational points there are some more precise results about this, as can be seen in the next theorem.

Theorem 2.36. *Let $p > 3, q = p^n$ and E be a supersingular elliptic curve over $\mathbb{F}_q$ with $E(\mathbb{F}_q) = q + 1 - t$ where $\mid t \mid \leq 2\sqrt{q}$. Then one of the following cases must be true:*

1. *n is even and $t = \pm 2\sqrt{q}$.*
2. *n is even, $p \not\equiv 1 \pmod 3$ and $t = \pm\sqrt{q}$.*
3. *n is even, $p \not\equiv 1 \pmod 4$ and $t = 0$.*
4. *n is odd and $t = 0$.*

In this situation the corresponding division algebra A is also determined by the cases. Let π_q be the q-th power Frobenius endomorphism. In the first case A is a quaternion algebra over $\mathbb{Q}$, π_q is a rational integer and $\mathrm{End}_{\mathbb{F}_q}(E)$ is a maximal order in A. In the other three cases $A = \mathbb{Q}(\pi_q)$ is an imaginary quadratic field over $\mathbb{Q}$ and $\mathrm{End}_{\mathbb{F}_q}(E)$ is an order in A with index $[\mathrm{End}_{\mathbb{F}_q}(E) : A]$ coprime with p.

Proof. See [96]. □

In the previous theorem we introduced a new notation, namely $\mathrm{End}_{\mathbb{F}_q}(E)$. With this symbol we denote the set of endomorphisms (of a given elliptic curve E) defined over $\mathbb{F}_q$. In this way we are restricting the more general set $\mathrm{End}(E)$, indeed it is possible that a curve, defined on a given field, has an endomorphism defined on an extension of this one. The following paragraph shows a particular case of these kind of maps.

Twists

We state the main results regarding twists, an important tool we use throughout this discussion. Here we assume that $\mathbb{F}_q$ is a field of characteristic different from 2.

Definition 2.37 (Twist)**.** Let $\mathbb{F}_q$ be a field and E_1, E_2 be elliptic curves over $\mathbb{F}_q$. Suppose that they are isomorphic over $\mathbb{F}_{q^d}$ for some $d > 1$, but are not isomorphic over any smaller extension of $\mathbb{F}_q$. E_1 and E_2 are said to be *degree-d twists* of each other (or, simply, a *twist*). In particular, a degree-2 twist is called a *quadratic* twist. Degree-3 and 4 twists will be called, respectively, *cubic* and *quartic* twists.

Proposition 2.38. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over $\mathbb{F}_q$. E has a quadratic twist given by*

$$E^d : dy^2 = x^3 + Ax + B$$

where $d \in \mathbb{F}_q^*$ is a non-square.

Proof. See [95, II]. □

Proposition 2.39. *Let E and E^d be a quadratic twist one of the other. Then*

$$|E(\mathbb{F}_q)| + |E^d(\mathbb{F}_q)| = 2q + 2$$

Equivalently, $t_E = -t_{E^d}$, where t_E is the trace[4] of the Frobenius endomorphism of the curve.

Proof. Consider the curve E given by $y^2 = f(x)$ and its quadratic d-twist E^d given by $dy^2 = f(x)$. Denote with $\chi : \mathbb{F}_q \to \{-1, 0, 1\}$ the map defined by

$$\chi(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x \text{ is a non-zero quadratic residue in } \mathbb{F}_q \\ -1 & \text{otherwise} \end{cases}$$

We immediately obtain that

$$\begin{cases} |E(\mathbb{F}_q)| = 1 + q - t_E \\ |E(\mathbb{F}_q)| = 1 + \sum_{x \in \mathbb{F}_q} (1 + \chi(f(x)) \end{cases}$$

The first equation will be discussed in Theorem 2.42. We conclude that

$$t_E = -\sum_{x \in \mathbb{F}_q} \chi(f(x))$$

$$t_{E^d} = -\sum_{x \in \mathbb{F}_q} \chi(f(x)/d) = -t_E$$

where the last equality is justified by the fact that d is not a quadratic residue over $\mathbb{F}_q$. □

Note that a generic curve and its twists always have the same j-invariant. Furthermore:

- If $j(E) \neq 0, 1728$, then $\mathrm{Aut}_{\mathbb{F}_q}(E)$ has order 2 with generator $(x, y) \mapsto (x, -y)$.
- If $j(E) = 1728$, then $\mathrm{Aut}_{\mathbb{F}_q}(E)$ is cyclic of order 4 with generator $\psi : (x, y) \mapsto (-x, iy)$ where $i \in \mathbb{F}_q$ is a primitive fourth root of unity.
- If $j(E) = 0$, then $\mathrm{Aut}_{\mathbb{F}_q}(E)$ is cyclic of order 6 with generator $\rho : (x, y) \to (\nu x, -y)$ where $\nu \in \mathbb{F}_q$ is a primitive third root of unity.

These results play a key role in the construction of isogeny graphs, as we will see later.

Remark 2.40. Given a finite field and two elliptic curves E, E^d defined on it, for all x there exist a y such that the point (x, y) belongs to either E or E^d, for some $d \in \mathbb{F}_q^*$ non-square, indeed consider a finite field $\mathbb{F}_q$, a map $f : \mathbb{F}_q \to \mathbb{F}_q$ such that $x \mapsto x^3 + Ax + B$ and an elliptic curve $E/\mathbb{F}_q := y^2 = f(x)$. Let x_P be an element of $\mathbb{F}_q$ and consider $f(x_P)$. If $f(x_P)$ is a quadratic residue in $\mathbb{F}_q$ then $(x_P, \pm\sqrt{f(x_P)})$ fits the relation and the result follows immediately, instead if $f(x_P) \in F_q^*$ is a non-square, we define $d := f(x_P)^{-1}$ and consider the relation $f(x_P)^{-1} \cdot y^2 = f(x)$. Observe that $(x_P, f(x_P))$ fits the equation, therefore it as a point of E^d.

[4] We refer to Theorem 2.41 for a proper definition.

2.3 Nontrivial Results

Let us see now how to use the morphisms described above to enrich our knowledge about elliptic curves.

Cardinality of $E/\mathbb{F}_q$

Every endomorphism on an elliptic curve satisfies a quadratic characteristic polynomial with integer coefficients.

Theorem 2.41. *Let $\mathbb{K}$ be a field, and let E be an elliptic curve over $\mathbb{K}$. Let $\varphi \in End(E)$ be a non-zero endomorphism, and denote with d the degree of φ. Then there is an integer t such that $\varphi^2 - t\varphi + d = 0$ in $End(E)$. In other words, for all $P \in E(\mathbb{K})$,*

$$\varphi(\varphi(P)) - [t]\varphi(P) + [d]P = O$$

The integer t is called the trace of the endomorphism.

Proof. See [35, Th. 9.9.3]. □

In particular, if we are given a finite field $\mathbb{F}_q$, the Frobenius satisfies the aforementioned relation for all $P \in E(\mathbb{F}_q)$. In other words, for every point $P = (x, y) \in E(\overline{\mathbb{F}_q})$ the following equation holds:

$$(x^{q^2}, y^{q^2}) + q(x, y) = t(x^q, y^q).$$

The integer t in equation above is naturally called the trace of Frobenius. This object is strongly related to the cardinality of $E/\mathbb{F}_q$, as the following result points out.

Theorem 2.42. *Let E be an elliptic curve over $\mathbb{F}_q$ and let $P(t)$ be the characteristic polynomial of the Frobenius. Then*

$$|E(\mathbb{F}_q)| = P(1) = 1 + q - t.$$

Proof. See [35, Th. 9.10.3]. □

From the previous result it is clear that, if we were able to limit the value of t, we would also have a bound for the value $|E/\mathbb{F}_q|$. Hasse's theorem on elliptic curves, also referred to as the Hasse bound, exploit exactly this fact and provides an estimate of the number of points of an elliptic curve over a finite field, bounding the value both from above and below.

Theorem 2.43 (Hasse)**.** *Let E be an elliptic curve over $\mathbb{F}_q$ and denote by t the trace of the q-power Frobenius map. Then*

$$|t| \leq 2\sqrt{q}.$$

Proof. See [35, Th. 9.10.7]. □

Since we know that $|E(\mathbb{F}_q)| = 1 + q - t$, Hasse's theorem states that, given an elliptic curve $E/\mathbb{F}_q$, the number of elements of $E(\mathbb{F}_q)$ satisfies the following relation

$$q + 1 - 2\sqrt{q} \leq |E(F_q)| \leq q + 1 + 2\sqrt{q}$$

Division Polynomials

A useful tool to investigate the group structure of $E(\mathbb{F}_q)$ is a family of multivariate polynomials of $\mathbb{F}_q[x, y]$ called the division polynomials of E. They play a central role in the study of counting points on elliptic curves and are defined recursively as following.

$$\begin{aligned}
\psi_0(x,y) &= 0 \\
\psi_1(x,y) &= 1 \\
\psi_2(x,y) &= 2y \\
\psi_3(x,y) &= 3x^4 + 6Ax^2 + 12Bx - A^2 \\
\psi_4(x,y) &= 4y(x^6 + 5Ax^4 + 20Bx^3 - 5A^2x^2 - 4ABx - 8B^2 - A^3) \\
&\vdots \\
\psi_{2m+1}(x,y) &= \psi_{m+2}(x,y)\psi_m^3(x,y) - \psi_{m-1}(x,y)\psi_{m+1}^3(x,y) \\
\psi_{2m}(x,y) &= (2y)^{-1}\psi_m(x,y)(\psi_{m+2}(x,y)\psi_{m-1}^2(x,y) - \psi_{m-2}(x,y)\psi_{m+1}^2(x,y))
\end{aligned}$$

Observe that a generic division polynomial depends on the parameters A, B that define the elliptic curve.

The roots of a generic l-th division polynomial are closely related to the l-torsion subgroup of an elliptic curve, in particular it is possible to show that the roots of ψ_{2n+1} are the x coordinates of the points of $E[2n+1] \setminus \{\mathcal{O}\}$, where $E[2n+1]$ is the $(2n+1)$-th torsion subgroup of E. Similarly, the roots of ψ_{2n}/y are the x-coordinates of the points of $E[2n] \setminus E[2]$.

An application that is very useful for our purposes is the use of these polynomials to compute the multiplication of a given point P by a scalar n. To compute nP for a large integer n, it is inefficient to add P to itself repeatedly. Instead of doing this, nP could be evaluated in another way using the n-th division polynomial: given $P \in E(\mathbb{F}_q)$ the following important relation holds.

$$n(x,y) =: (x_n, y_n) = \left(x - \frac{\psi_{n-1}\psi_{n+1}}{2\psi_n^2}, \frac{\psi_{2n}}{2\psi_n^4}\right). \tag{2.3}$$

Isogeny Classes

We have previously showed that being isogenous is an equivalence relation, it thus makes sense to speak of the *isogeny class* of an elliptic curve. We start with a theorem that links isogeny classes with the theory we previously described.

Definition 2.44 (Endomorphism algebra)**.** Let E be an elliptic curve over $\mathbb{F}_q$. Denote with π the Frobenius endomorphism of E and with t its trace. Define then $D_\pi := t^2 - 4q$ and observe that $D_\pi < 0$, thanks to Theorem 2.43. $\mathbb{Q}(\sqrt{D_\pi})$ is called the *endomorphism algebra* of E.

It is possible to verify that $\pi \in \mathbb{Q}(\sqrt{D_\pi})$; so, at least in the ordinary case, remembering Theorem 2.35, we can affirm that

$$\mathbb{Z}[\pi] \subseteq \mathrm{End}(E) \subseteq \mathbb{Q}(\sqrt{D_\pi}).$$

Theorem 2.45 (Serre-Tate)**.** *Two elliptic curves E, E' with complex or quaternion multiplication are isogenous if and only if their* endomorphism algebras *are isomorphic.*

In layman's terms, this theorem is telling us that:

- Two curves with complex multiplications by $\mathcal{O}$ and $\mathcal{O}'$ respectively are isogenous if and only if $\mathcal{O} \subseteq \mathcal{O}'$ or $\mathcal{O}' \subseteq \mathcal{O}$; or equivalently if and only if $\mathcal{O}$ and $\mathcal{O}'$ have the same field of fractions.
- Any two supersingular curves defined over a field of characteristic p are isogenous, indeed, thanks to Teorem 2.36, both curves have zero Frobenius trace.

An easy consequence for the finite field case is the following.

Corollary 2.46. *Two elliptic curves E, E' defined over a finite field $\mathbb{K}$ are isogenous over $\mathbb{K}$ if and only if $|E(\mathbb{K})| = |E'(\mathbb{K})|$.*

Modular Polynomials

Given an elliptic curve E defined on a finite field $\mathbb{F}_q$, and a subgroup $G \subseteq E(\mathbb{F}_q)$, then there exists an isogeny $\varphi : E \to E/G$. We have shown how this map can be computed with the formulæ of Vélu, however this is not the only possible way: there is another tool to compute isogenies, which does not explicitly involve the kernel subgroup or points on the curve. This more elegant approach involves modular polynomials. It is beyond the scope of this discussion to present the theory of modular functions and modular curves therefore we just state the results that are relevant to us (some basic references are [60, Sections 5.2 and 5.3] and [22, Section 11]).

Let l be an integer with $l \geq 2$. The modular polynomial $\varphi_l(x, y) \in \mathbb{Z}[x, y]$ has the following remarkable property: a pair $j, j' \in \mathbb{F}_q$ satisfies $\varphi(j, j') = 0$ if and only if there are elliptic curves E, E' over $\mathbb{F}_q$ with $j(E) = j$ and $j(E') = j'$ and an isogeny $\varphi : E \to E'$ of degree l. It follows from the dual isogeny theorem that $\varphi_l(y, x) = \varphi_l(x, y)$. Hence, given an elliptic curve E over $\mathbb{F}_q$, to find the j-invariants of the l-isogenous curves one simply computes the uni-variate polynomial $\varphi_l(j(E), y) \in \mathbb{F}_q[y]$ and then computes its roots in $\mathbb{F}_q$. An algorithm due to Elkies allows to compute the kernel of the corresponding isogeny when given E and the j-invariant j' of the isogenous curve E'.

Chapter 3

Isogeny Graphs

In this section we explain some aspects of elliptic curves with complex and quaternionic multiplication. In this way we will provide the theoretical basis to understand isogeny graphs. For further details we refer to [85], [25] and [24].

3.1 Complex Multiplication

We present one of the most powerful tools for the study of isogeny graphs: the theory of *complex multiplication*. This theory concerns all elliptic curves whose endomorphism ring is an order in an imaginary quadratic field. In particular, we study the set of curves sharing the same endomorphism ring $\mathcal{O}$, and denote this set with $\mathrm{Ell}(\mathcal{O})$. To do this we introduce a new object, the ideal class group $\mathrm{Cl}(\mathcal{O})$. We show that these two sets are in one-to-one relation, so that

$$\mathrm{Ell}(\mathcal{O}) \overset{1:1}{\longleftrightarrow} \mathrm{Cl}(\mathcal{O})$$

and we find out that $\mathrm{Cl}(\mathcal{O})$ acts on $\mathrm{Ell}(\mathcal{O})$ freely and transitively. We will explore these concepts later in the chapter. This last result is of fundamental importance for the CSIDH protocol. Before defining the ideal class group, we still need to recall some basic definitions from algebraic number theory; for a more detailed treatment, see [61].

Definition 3.1 (Discriminant)**.** Let d be a square free integer, the *discriminant* of the quadratic number field $\mathbb{Q}(\sqrt{d})$ is d if $d \equiv 1 \bmod 4$, and $4d$ otherwise.

The discriminant is an object that can be defined more generically, and what we have provided above is a specific definition where the field extension has degree 2. In this case the discriminant is also called the *fundamental discriminant*.

Definition 3.2 (Ring of integers)**.** Let K be a number field, an *algebraic integer* of K is an element $\alpha \in K$ that is root of an irreducible monic polynomial with integer coefficients. The algebraic integers of K form a ring, called the *ring of integers* of K. We will denote this set with $\mathcal{O}_K$.

For example, $\mathbb{Z}[i]$ is the ring of integers of $\mathbb{Q}(i)$. More generally, if we consider a quadratic field $K = \mathbb{Q}(\sqrt{-d})$, the corresponding ring of integers $\mathcal{O}_K$ is given by

$$\begin{aligned} \mathbb{Z}[\sqrt{d}] \quad &\text{if} \quad d \equiv 2, 3 \pmod 4 \\ \mathbb{Z}[(1+\sqrt{d})/2] \quad &\text{if} \quad d \equiv 1 \pmod 4 \end{aligned}$$

By Definition 2.32, an order of a quadratic field K is a sub-ring of K that is a $\mathbb{Z}$-module of rank 2. The ring of integers $\mathcal{O}_K$ of K fits the bill: it always has $(1, \sqrt{\Delta})$ or $(1, (1+\sqrt{\Delta})/2)$ as *integral basis*, i.e., as a set of $\mathbb{Z}$-module generators. Furthermore, it is easy to prove that any other order is contained in $\mathcal{O}_K$; for this reason we will sometimes call it the *maximal order* of K. More precisely, we can prove the following.

Proposition 3.3. *Let K be a quadratic number field, and let $\mathcal{O}_K$ be its ring of integers. Any order $\mathcal{O} \subseteq K$ can be written as $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_K$ for an integer f, called the* conductor *of $\mathcal{O}$. If Δ_K is the discriminant of K, the* discriminant *of $\mathcal{O}$ is $f^2\Delta_K$. If $\mathcal{O}, \mathcal{O}'$ are two orders of discriminants Δ, Δ', then $\mathcal{O} \subseteq \mathcal{O}'$ if and only if $\Delta' | \Delta$.*

Proof. See [57, Prop. 21]. □

It can be shown that the integer f in the previous theorem, that is the conductor of the order $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_K$, is equal to the index $[\mathcal{O}_K : \mathcal{O}]$, which is necessarily finite.

When K is imaginary quadratic, any order $\mathcal{O} \subseteq K \subseteq \mathbb{C}$ is a complex lattice[1] by definition. We now define a broader class of lattices.

Definition 3.4 (Fractional ideal)**.** Let $\mathcal{O}$ be an order in a number field K. A *fractional ideal* of $\mathcal{O}$ (equivalently, an $\mathcal{O}$-fractional ideal) is a non-zero subgroup $\mathfrak{a} \subseteq K$ such that

- $x\mathfrak{a} \subseteq \mathfrak{a}$ for all $x \in \mathcal{O}$.
- there exists a non-zero $x \in \mathcal{O}$ such that $x\mathfrak{a} \subseteq \mathcal{O}$.

If $\mathfrak{a}$ is generated by a single element, then it is called *principal*. If $\mathfrak{a} \subseteq \mathcal{O}$, then it is called an *integral* ideal. The norm of an $\mathcal{O}$-integral ideal $\mathfrak{a} \subseteq \mathcal{O}$ is defined as $N(\mathfrak{a}) = | \mathcal{O}/\mathfrak{a} |$.

Note that the ideals of $\mathcal{O}$ are exactly the fractional ideals contained in $\mathcal{O}$; since we usually use fractional ideals in this discussion, we will often refer to these objects simply as ideals, and we will use the name integral ideal for ordinary ones.

Example 3.5. As a basic example of fractional ideal we can take $K := \mathbb{Q}(\sqrt{-1})$ and $\mathcal{O} := \langle 1, 2i \rangle$ over $\mathbb{Q}$. We then define $\mathfrak{a} := \{\frac{x}{2i} \mid x \in \mathcal{O}\}$. It is immediate to verify that $x\mathfrak{a} \subseteq \mathfrak{a}$ for all $x \in \mathcal{O}$ and $2i \cdot \mathfrak{a} \subseteq \mathcal{O}$, so that the request of Definition 3.4 are satisfied and $\mathfrak{a}$ is a fractional ideal.

We define the product of two fractional ideals $\mathfrak{a}, \mathfrak{b}$ as

$$\mathfrak{ab} := \{xy \mid x \in \mathfrak{a}, y \in \mathfrak{b}\}$$

Definition 3.6 (Invertible fractional ideal)**.** An $\mathcal{O}$-ideal $\mathfrak{a}$ is *invertible* if there exists another ideal $\mathfrak{a}^{-1}$ such that $\mathfrak{a}\mathfrak{a}^{-1} = \mathfrak{a}^{-1}\mathfrak{a} = \mathcal{O}$.

Invertible ideals form an abelian group, written multiplicatively, under the operation defined above, where $\mathcal{O}$ is the neutral element. It is immediate to verify that principal ideals form a subgroup of it. Furthermore, if $\mathcal{O}$ is the maximal order of K, it is possible to prove that any $\mathcal{O}$-ideal is invertible.

[1] A discrete subgroup of $\mathbb{C}$ that contains an $\mathbb{R}$-basis of $\mathbb{C}$.

Definition 3.7 (Ideal class group)**.** Let $\mathcal{O}$ be an order in a number field K. Let $\mathcal{I}(\mathcal{O})$ be the group of invertible fractional $\mathcal{O}$-ideals, and let $\mathcal{P}(\mathcal{O})$ be the group of principal ideals. The *ideal class group* of $\mathcal{O}$ is the quotient group

$$\mathrm{Cl}(\mathcal{O}) = \mathcal{I}(\mathcal{O})/\mathcal{P}(\mathcal{O}).$$

Its order is called the *class number* of $\mathcal{O}$, and denoted by $h(\mathcal{O})$. When $\mathcal{O}$ is the maximal order, $\mathrm{Cl}(\mathcal{O})$ is also called the class group of K.

It is well known that the class group is a finite abelian group, so that its order is finite. What is relevant to us is that each element of $\mathrm{Cl}(\mathcal{O})$ acts on $\mathrm{Ell}(\mathcal{O})$ in a very particular way, as shown in the following.

Definition 3.8 ($\mathfrak{a}$-torsion)**.** Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Let $\mathcal{O}$ be the endomorphism ring of E, and let $\mathfrak{a} \subseteq \mathcal{O}$ be an integral invertible ideal of norm coprime to q. We define the $\mathfrak{a}$-*torsion subgroup* of E as

$$E[\mathfrak{a}] = \{P \in E \mid \alpha(P) = 0 \text{ for all } \alpha \in \mathfrak{a}\}.$$

Given an ideal $\mathfrak{a} \subseteq \mathcal{O}$ as above, it is natural to define the (separable) isogeny $\varphi_\mathfrak{a} : E \to E_\mathfrak{a}$, where $E_\mathfrak{a} = E/E[\mathfrak{a}]$. The following theorem points out some important results about the relation between the ideal class group and the set of elliptic curves with complex multiplication by a fixed order.

Theorem 3.9. *Let $\mathbb{F}_q$ be a finite field, and let $\mathcal{O} \subseteq \mathbb{Q}(\sqrt{-D})$ be an order in a quadratic imaginary field. Denote by $\mathrm{Ell}_q(\mathcal{O})$ the set of elliptic curves defined over $\mathbb{F}_q$ with complex multiplication by $\mathcal{O}$. Assume that $\mathrm{Ell}_q(\mathcal{O})$ is non-empty, then the class group $\mathrm{Cl}(\mathcal{O})$ acts* freely[2] *and* transitively[3] *on it; i.e., there is a map*

$$\begin{aligned} \mathrm{Cl}(\mathcal{O}) \times \mathrm{Ell}_q(\mathcal{O}) &\to \mathrm{Ell}_q(\mathcal{O}) \\ (\mathfrak{a}, E) &\mapsto \mathfrak{a} \cdot E \end{aligned}$$

such that $\mathfrak{a} \cdot (\mathfrak{b} \cdot E) = (\mathfrak{a}\mathfrak{b}) \cdot E$ for all $\mathfrak{a}, \mathfrak{b} \in \mathrm{Cl}(\mathcal{O})$ and $E \in \mathrm{Ell}_q(\mathcal{O})$, and such that for any $E, E' \in \mathrm{Ell}_q(\mathcal{O})$ there is a unique $\mathfrak{a} \in \mathrm{Cl}(\mathcal{O})$ such that $E' = \mathfrak{a} \cdot E$.

Said otherwise, $\mathrm{End}(E) \cong \mathrm{End}(E_\mathfrak{a}) \cong \mathcal{O}$, $E_\mathfrak{a}$ only depends on the class of $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$, and the map $(\mathfrak{a}, E) \mapsto E_\mathfrak{a}$ defines a group action of $\mathrm{Cl}(\mathcal{O})$ on the set of elliptic curves with complex multiplication by $\mathcal{O}$.

Remark 3.10. We observe that $E_\mathfrak{a}$ is only defined when $\mathfrak{a} \subseteq \mathcal{O}$, however in Theorem **??** we are using a generic element of $\mathrm{Cl}(\mathcal{O})$, which should not being contained in $\mathcal{O}$. This is not a problem, in fact every class in this set contains an integral invertible ideal: suppose $\mathfrak{b}$ to be a generic element of $\mathrm{Cl}(\mathcal{O})$, then $\mathfrak{b}$ is an invertible ideal, and therefore by Definition 3.4 there exists non-zero $x \in \mathcal{O}$ such that $\mathfrak{a} := x\mathfrak{b} \subseteq \mathcal{O}$. Regarding the definition of $\mathrm{Cl}(\mathcal{O})$, it follows that these two objects belong to the same class.

A set that is acted upon freely and transitively by a group G, is also called a *principal homogeneous space* or a *torsor* for G. An immediate consequence of the theorem above is that that, for any fixed base point $E \in \mathrm{Ell}_q(\mathcal{O})$, there is a bijection

$$\begin{aligned} \mathrm{Cl}(\mathcal{O}) &\longrightarrow \mathrm{Ell}_q(\mathcal{O}) \\ \text{Ideal class of } \mathfrak{a} &\longmapsto \text{Isomorphism class of } \mathfrak{a} \cdot E. \end{aligned}$$

and so the torsor $\mathrm{Ell}_q(\mathcal{O})$ has cardinality equal to the class number $h(\mathcal{O})$.

[2]Namely: if $\mathrm{Ell}(\mathcal{O}) \neq \varnothing$ and if for each pair x, y in $\mathrm{Ell}(\mathcal{O})$ there exists a g in $\mathrm{Cl}(\mathcal{O})$ s.t. $g \cdot x = y$.

[3]Namely: if $g \in \mathrm{Cl}(\mathcal{O})$ and there exists an x in $\mathrm{Ell}(\mathcal{O})$ with $g \cdot x = x$, then g is the identity.

3.2 Quaternionic Multiplication

In this discussion we always deal with elliptic curves defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$: it is a well known result in literature that every supersingular elliptic curve defined over $\overline{\mathbb{F}}_p$ is isomorphic to one defined over $\mathbb{F}_{p^2}$. When E is defined over $\mathbb{F}_p$ we fall in case (4) of Prop. 2.36, and $\mathrm{End}_{\mathbb{F}_p}(E)$, the ring of $\mathbb{F}_p$-rational endomorphisms, is isomorphic to an order in $\mathbb{Q}(\sqrt{-p})$. When E is defined over $\mathbb{F}_{p^2}$, however, $t \in \{0, \pm p, \pm 2p\}$. The cases $t \in \{0, \pm p\}$ only happen for a very limited number of curves with j-invariant 0 or 1728; we are thus mostly interested in case (1) of Prop. 2.36, where $t = \pm 2p$, i.e., $\pi = \pm p$. In this case the full endomorphism ring $\mathrm{End}(E)$ (i.e., not restricted to $\mathbb{F}_p$-rational endomorphisms) is isomorphic to a maximal order the quaternion algebra $B_{p,\infty}$ ramified at p and at infinity.

Example 3.11. The elliptic curve $y^2 = x^3 + x$ has supersingular reduction at all primes $p = 3 \bmod 4$. Its ring of $\mathbb{F}_p$-rational endomorphisms is generated by $\pi = \sqrt{-p}$, and it is not maximal in $\mathbb{Q}(\sqrt{-p})$. The automorphism $\iota : (x, y) \mapsto (-x, iy)$ is only defined over $\mathbb{F}_{p^2}$, and does not commute with π. The full endomorphism ring is isomorphic to the order generated by π and ι inside the quaternion algebra $B_{p,\infty}$.

Let us first consider curves over $\mathbb{F}_{p^2}$. The following discussion is not necessary for CSIDH to work, in fact the protocol just make use of supersingular curves for which the endomorphism ring is isomorphic to an order in an imaginary quadratic field. We will however briefly explain the theory of quaternionic multiplication for completeness.

Given a maximal order $\mathcal{O}$ in a quaternion algebra, we would like to study the set of supersingular elliptic curves $\mathrm{Ell}(\mathcal{O})$. As before, we introduce the (left) class set $\mathrm{Cl}(\mathcal{O})$, and we show that the two sets are in one-to-one relation:

$$\mathrm{Ell}(\mathcal{O}) \overset{1:1}{\longleftrightarrow} \mathrm{Cl}(\mathcal{O})$$

Like the CM case, isogenies are in correspondence with (left) ideals of $\mathcal{O}$. More precisely, let $\mathfrak{a} \subseteq B_{p,\infty}$ a lattice, the *left order* of $\mathfrak{a}$ is the ring $\mathcal{O}(\mathfrak{a}) = \{x \in B_{p,\infty} \mid x\mathfrak{a} \subseteq \mathfrak{a}\}$. Two lattices $\mathfrak{a}, \mathfrak{b}$ are said to be *right isomorphic* if $\mathfrak{a} = \mathfrak{b}x$ for some $x \in B_{p,\infty}$. If $\mathcal{O} \subseteq B_{p,\infty}$ is an order, $\mathfrak{a}$ is called a *left ideal* of $\mathcal{O}$ if $\mathcal{O} \subseteq \mathcal{O}(\mathfrak{a})$; the *left class set* $\mathrm{Cl}(\mathcal{O})$ is the set of right ideal classes of left ideals of $\mathcal{O}$.

The cardinality $|\mathrm{Cl}(\mathcal{O})|$ only depends on the quaternion algebra, and is called the *class number* of $B_{p,\infty}$. Analogous definitions can be given by swapping left and right; we refer to [94, Chapter 42] for more properties and definitions.

Theorem 3.12. *Let $B_{p,\infty}$ be the quaternion algebra ramified at p and infinity, and consider a maximal order $\mathcal{O} \subseteq B_{p,\infty}$. Let $E_0/\mathbb{F}_{p^2}$ be a supersingular elliptic curve with $\mathrm{End}(E_0) \cong \mathcal{O}$.*

1. *The number of isomorphism classes of supersingular elliptic curves is equal to the class number of $B_{p,\infty}$.*
2. *There is a one-to-one correspondence $\mathfrak{a} \mapsto \mathfrak{a} \cdot E_0$ between $\mathrm{Cl}(\mathcal{O})$ and the set of isomorphism classes of supersingular elliptic curves, such that $\mathrm{End}(\mathfrak{a} \cdot E_0)$ is isomorphic to the right order of $\mathfrak{a}$.*

3.3 Graphs

In this section we set out some elementary results regarding graph theory, with particular emphasis to those properties that are useful in cryptography. In particular we focus on those graphs that have good pseudo-randomness properties, i.e. those graphs for which a sufficiently large random walk ends on each vertex with probability close to the uniform. We describe these properties formally, and see how to deterministically construct such graphs. The aim of this exposition is to provide the basis to fully understand the deeper theory behind isogeny graphs. We first recall some basic concepts. In the following, given the considerable amount of notions and definitions necessary for the discussion, we prefer a less formal approach. We refer to [24] for further details.

Definition 3.13 (Graph)**.** A *directed (un-directed) graph* is a pair $G = (V, E)$, where V is a set, whose elements are called *vertices*, and $E \subseteq V \times V$ is a set of ordered (un-ordered) pairs, whose elements are called *edges*, so that two vertices v and w are said to be *connected by an edge* if $\{v, w\} \in E$.

The vertices v and w of an edge $\{v, w\}$ are called the endpoints of the edge, and the edge is said to join v and w. Clearly a vertex may not belong to any edge. These graphs are sometimes called *simple graphs* for distinguishing them from *multi graphs*.

Definition 3.14 (Multi-graph)**.** A *directed (undirected) multi-graph* is a pair $G = (V, E)$, where V is a set, whose elements are called *vertices*, and $E \subseteq V \times V$ is a multi-set of ordered (un-ordered) pairs, whose elements are called *edges*.

A multi-graph is a generalization that allows multiple edges to have the same pair of endpoints. In this discussion we always deal with multi-graphs, therefore, to avoid weighing down the notation, we simply call them graphs. Note that with these definitions graphs are allowed to contain loops, which are edges that join a vertex to itself. Figure 3.1 provides an example of such a graph.

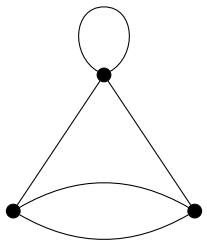


Figure 3.1: An undirected multi-graph.

As we always deal with undirected graph, unless otherwise stated, a graph will always be considered undirected. A *path* between two vertices v, v' is a sequence of vertices

$$v \to v_1 \to \cdots \to v'$$

such that each vertex is connected to the next one by an edge. The *distance* between two vertices is the length of the shortest path between them; if there is no such path, the vertices are said to be at infinite distance. Given a connected graph $G = (V, E)$, the *diameter* of this G is the largest of all distances between its vertices.

Definition 3.15 (Connected graph)**.** A graph is called *connected* if any two vertices do have a path connecting them; it is called *disconnected* otherwise.

Given a graph, the *neighbours* of a vertex v are the vertices of V connected to it by an edge. The number of neighbours defines the *degree* of v.

Definition 3.16 (Regular graph)**.** A *regular graph* is a graph in which each vertex has the same number of neighbours, i.e. every vertex has the same degree. A regular graph with vertices of degree k is called a *k-regular graph* or *regular graph of degree k*.

Definition 3.17 (Adjacency matrix)**.** Given a graph $G = (V, E)$ we define the *adjacency matrix* A associated to G as the matrix $A_{i,j} := |\{(i, j) \in E\}|$, that is, as the number of edges that connect i to j.

For a simple graph we always have $A_{i,j} \in \{0, 1\}$. Differently for a multi-graph it could happen that $A_{i,j} > 1$. Since our graphs G are undirected, the adjacency matrix will always be symmetric, thus it will always have n real eigenvalues (called the eigenvalues of G)

$$\lambda_1 \geq \cdots \geq \lambda_n.$$

We can immediately bound the eigenvalues of G.

Proposition 3.18. *Let G be a k-regular graph. Then its largest and smallest eigenvalues λ_1, λ_n satisfy*

$$k = \lambda_1 \geq \lambda_n \geq -k.$$

Proof. See [91, Lem. 2] □

Because of this equality, λ_1 is called the *trivial eigenvalue*. An *expander graph* is a k-regular graph such that its non-trivial eigenvalues are bounded away, in absolute value, in a way we'll see soon. The random-like properties of graphs are typically expressed in terms of *expansion*. We recall here some basic facts about expanders; for an in depth review, see [42], [31].

Definition 3.19 (Expander graph)**.** Let $\varepsilon > 0$ and $k \geq 1$. A k-regular graph is called a (one-sided) *ε-expander* if

$$\lambda_2 \leq (1 - \varepsilon)k$$

and a *two-sided ε-expander* if it also satisfies

$$\lambda_n \geq -(1 - \varepsilon)k \ = (1 - \varepsilon)(-k).$$

A sequence $G_i = (V_i, E_i)$ of k-regular graphs with $|V_i| \to \infty$ is said to be a one-sided (resp. two-sided) *expander family* if there is an $\varepsilon > 0$ such that G_i is a one-sided (resp. two-sided) ε-expander for all sufficiently large i.

Theorem 3.20. *For any infinite family of k-regular graphs on n nodes, denoted $\{G_n\}_n$, we have*

$$\lambda(G_n) := \max(|\lambda_2|, |\lambda_n|) \geq 2\sqrt{k - 1} - o(1)$$

where the last term is some number that goes to zero as n gets large.

Definition 3.21 (Ramanujan graph)**.** A k-regular graph such that $|\lambda_j| \leq 2\sqrt{k - 1}$ for any λ_j except λ_1 is called a *Ramanujan graph*.

Ramanujan graphs are not only expander graphs, they are essentially the best possible expander graphs: the constant $2\sqrt{k-1}$ in the definition of Ramanujan graphs is the best possible constant for each k and for large graphs: thanks to Theorem 3.20, for every k and $\varepsilon > 0$, there exists a natural number n such that all k-regular graphs G with at least n vertices satisfy $\lambda(G) > 2\sqrt{k-1} - \varepsilon$. This type of graphs naturally arise in isogeny graphs of supersingular elliptic curves[4].

Edge expansion quantifies how well subsets of vertices are connected to the whole graph, or, said otherwise, how far the graph is from being disconnected.

Definition 3.22 (Edge expansion)**.** Let $F \subseteq V$ be a subset of the vertices of G. The *boundary of* F, denoted by $\partial F \subseteq E$, is the subset of the edges of G that go from F to $V \setminus F$. The *edge expansion ratio* of G, denoted by $h(G)$ is the quantity

$$h(G) = \min_{\substack{F \subseteq V, \\ |F| \leq |V/2|}} \frac{|\partial F|}{|F|}.$$

Note that $h(G) = 0$ if and only if G is disconnected. Edge expansion is strongly tied to expander graphs, as the following theorem shows.

Theorem 3.23 (Discrete Cheeger inequality)**.** *Let G be a k-regular one-sided ε-expander, then*

$$\frac{\varepsilon}{2}k \leq h(G) \leq \sqrt{2\varepsilon}k.$$

Qualitatively, we can describe these graphs as having *short diameter* and *rapidly mixing walks*. Another reason for which these graphs are important in cryptography is that finding paths in these graphs, i.e. routing, is hard: there are no known subexponential algorithms to solve this problem, either classically or on a quantum computer.

Proposition 3.24. *Let G be a k-regular one sided ε-expander graph. For any vertex v and any radius $r > 0$, let $B(v, r)$ be the* ball *of vertices at distance at most r from v. Then, there is a constant $c > 0$, depending only on k and ε, such that*

$$|B(v, r)| \geq \ (1 + c)^r$$

In particular, this shows that the diameter of an expander is bounded by $O(\log n)$, where $n := |G|$, indeed if we consider $|B(v, \log n)|$, the number of nodes that can be reached from v in $\log n$ steps, we have that this number is greater than $(1 + c)^{\log n} \in O(n)$, so we are able to reach each node of G. A *random walk* of length i is a path $v_1 \to \cdots \to v_i$, defined by the random process that selects v_i uniformly at random among the neighbors of v_{i-1}. Loosely speaking, the next proposition says that, in an expander graph, random walks of length close to its diameter terminate on any vertex with probability close to uniform.

Proposition 3.25 (Mixing theorem)**.** *Let $G = (V, E)$ be a k-regular two-sided ε-expander graph. Let $F \subseteq V$ be any subset of the vertices of G, and let v be any vertex in V. Then a random walk of length at least*

$$\frac{\log(|F|^{1/2}/(2|V|))}{\log(1 - \varepsilon)}$$

starting from v will land in F with probability at least $|F|/(2|V|)$.

[4]They play a fundamental role in the SIDH protocol.

Proof. See [50]. □

The walk length in the mixing theorem is also called the *mixing length* of the expander graph. Random regular graphs typically make good expanders, but only a handful of deterministic constructions are known, most of them based on Cayley and Schreirer graphs [64, 31, 42], which we are going to define in the following.

Definition 3.26 (Cayley graph)**.** Let G be a group and $S \subseteq G$ be a symmetric subset, i.e. not containing 1 and closed under inversion. The *Cayley graph* of (G, S) is the undirected graph whose vertices are the elements of G, and such that $g, g' \in G$ are connected by an edge if and only if there exists $s \in S$ such that $g' = sg$.

An immediate generalization of Cayley graphs are Schreier graphs, where the standard multiplication between elements of the same group is replaced by the action of a group G on a given set X, as the following definition shows.

Definition 3.27 (Schreirer graph)**.** Let G be a group *acting freely* on a set X, in the sense that there is a map

$$\begin{aligned} G \times X &\to X \\ (\sigma, x) &\mapsto \sigma \cdot x \end{aligned}$$

such that $\sigma \cdot x = x$ if and only if $\sigma = 1$, and $\sigma \cdot (\tau \cdot x) = (\sigma\tau) \cdot x$, for all $\sigma, \tau \in G$ and $x \in X$. Let $S \subseteq G$ be a *symmetric* subset. The *Schreier graph* of (S, X) is the graph whose vertices are the elements of X, and such that $x, x' \in X$ are connected by an edge if and only if $x' = \sigma \cdot x$ for some $\sigma \in S$.

Because of the constraints on the group action and the set S, Schreier graphs are undirected and regular, and they usually make good expanders. Note that Cayley graphs are the Schreier graphs of the (left) action of a group on itself.

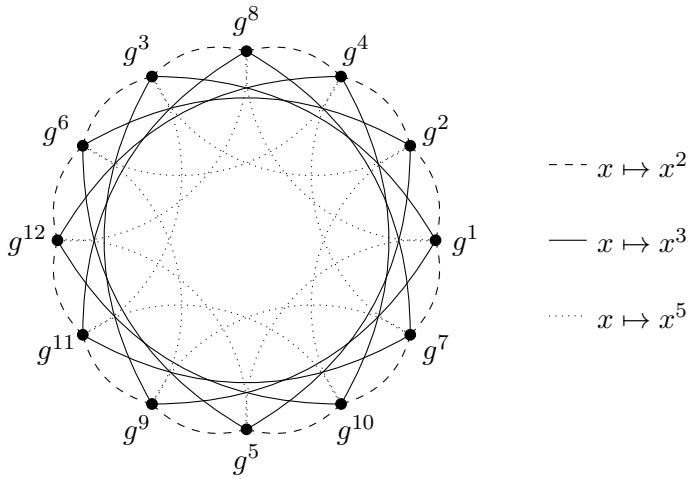


Figure 3.2: Schreier graph of $(\{2, 3, 5, 2^{-1}, 3^{-1}, 5^{-1}\}, \mathbb{Z}_{13}^*)$.

Example 3.28. As an example, take as a set a cyclic group G of order n, then the group $(\mathbb{Z}/n\mathbb{Z})^*$ acts naturally on G by the law $\sigma \cdot g = g^\sigma$ for any $g \in G$ and $\sigma \in (\mathbb{Z}/n\mathbb{Z})^*$. This action is not free on G, but it is so on the subset P of all generators of G; we can thus build the Schreier graph (S, P), where S is a symmetric subset that generates $(\mathbb{Z}/n\mathbb{Z})^*$. An example of such graph for the case $n = 13$ is shown in Figure 3.2, where the set $S \subseteq (\mathbb{Z}/13\mathbb{Z})^*$ has been chosen to contain $2, 3, 5$ and their inverses.

3.4 l-Isogeny Graphs

This is a key section: here we combine the theory of elliptic curves and graph theory. We show how it is possible to construct a graph whose vertices are isomorphism classes of elliptic curves and whose edges represent set of isogenies defined between these curves. Later we will restrict these graphs by basically imposing two limitations:

- We limit ourselves to consider as vertices of the graph only the elliptic curves having as endomorphism ring (an algebraic structure isomorphic to) a fixed order $\mathcal{O}$.
- We consider as edges only the isogenies defined over $\mathbb{F}_p$, the base field of the curve, where p is a prime. In other words, we will ignore isogenies strictly defined on the algebraic closure $\overline{\mathbb{F}}_p$.

In these graphs we observe that some connected components admit a sub-graph isomorphic to a Cayley graph. We study these components in order to define a cryptographic protocol above. We proceed in order with the discussion, starting from the basic definitions.

Definition 3.29 (Isogeny graph)**.** Let L be a non-empty set of small primes. Let $\mathbb{K}$ be a field. An *isogeny graph* $G(\mathbb{K}, L)$ is a directed graph where its vertices are $\mathbb{K}$-isomorphism classes of elliptic curves over $\mathbb{K}$ and its edges are equivalence classes of l-isogenies defined over $\mathbb{K}$ between such curves for $l \in L$ (two isogenies are equivalent if they have the same kernel). If we only consider $L = \{l\}$, we write $G(\mathbb{K}, l)$ and we say that this is an l-isogeny graph and that two curves are l-isogenous.

Remark 3.30. When working with a finite field $\mathbb{F}_p$, usually vertices are represented by j-invariants, but this choice is appropriate only if we are considering regular curves: an ordinary elliptic curve over $\mathbb{F}_p$ is never isogenous to its non-trivial quadratic twist since they have a different number of $\mathbb{F}_p$-rational points, so we never have to care about twists when considering isogenies between ordinary elliptic curves over $\mathbb{F}_p$. If the curves are supersingular though, this is not the case. Let $p > 3$ be a prime; a supersingular elliptic curve over $\mathbb{F}_p$ has $p+1$ points and so, thanks to Theorem 2.39, all quadratic twists have the same number of points. Thus the twists are isogenous but lie in different $\mathbb{F}_p$-isomorphism classes. Therefore it is not very precise to represent the vertices in the supersingular isogeny graph over $\mathbb{F}_p$ with j-invariants, since in this way different isomorphism classes over $\mathbb{F}_p$ would collapse to only one vertex and the picture of in- and outgoing isogenies would be distorted. If we want to differentiate between twists, we have to store more information than just the j-invariants of the elliptic curves.

Remark 3.31. A priori an isogeny graph is a directed graph, but given two elliptic curves E_1 and E_2 whose j-invariants are not in $\{0, 1728\}$, there are exactly as many edges (E_2, E_1) as (E_1, E_2), obtained by taking dual isogenies. Annoyingly, the nodes with j-invariants 0 and 1728 are more complicated, since these are exactly the curves with extra automorphisms: an elliptic curve E has fewer incoming than outgoing edges if and only if either $j(E) = 0$ and $\sqrt{-3} \in \mathbb{K}$, or if $j(E) = 1728$ and $\sqrt{-1} \in \mathbb{K}$. Throughout this discussion, we will assume for simplicity that $\sqrt{-1}, \sqrt{-3} \notin \mathbb{K}$, so that neither of these automorphisms are defined over $\mathbb{K}$ and we may view the graph as an undirected graph. In the case of a finite prime field $\mathbb{K} = \mathbb{F}_p$ this is equivalent to require $j = 0$ and $p \equiv 1 \pmod 3$ or $j = 1728$ and $p \equiv 1 \pmod 4$. However we will show that this is not a problem for the case of supersingular curves define over $\mathbb{F}_p$.

With these premises, isogeny graphs are usually drawn undirected.

Traditionally, for regular curves we study the graph whose isogenies are defined on the same field $\mathbb{F}_q$ of the curves, while for the supersingular ones we study the graph whose isogenies are defined on the algebraic closure $\overline{\mathbb{F}}_q$. We will follow this approach, but since we are interested in graphs in which curves and isogenies are defined over $\mathbb{F}_p$, we will have to do some additional work on the supersingular isogeny graph.

Depending on the constraints we put, we get graphs with different structures. The most important ones will be *isogeny volcanoes*, *Cayley graphs*, and *supersingular graphs*. The constrains that are usually imposed are the definition field of the curve ($\mathbb{C}, \mathbb{F}_p, \mathbb{F}_{p^2}$ or $\overline{\mathbb{F}}_p$), the definition field of isogenies ($\mathbb{F}_p$ or $\mathbb{F}_{p^2}$), in the case we are working in a finite field, or the degree l of the isogenies to which we want to restrict. We focus our attention on isogenies of fixed degree l, distinguishing the case in which the endomorphism ring is isomorphic to $\mathbb{Z}$, an order in an imaginary quadratic field, or an order in a quaternion algebra. For each case we will answer the most natural questions about it: given a curve E we will show how many isogenies of degree l admit E as a domain, what is the global structure of the graph, how many are the connected components, and if they share the same structure.

3.4.1 $\mathrm{End}(E) \cong \mathbb{Z}$

The first case we deal with is those of curves with endomorphism ring isomorphic to $\mathbb{Z}$. Let us start from the local structure: given an elliptic curve E and a prime l, how many isogenies of degree l do have E as domain? Thanks to Proposition 2.20, we know this is equivalent to ask how many subgroups of order l the curve has; but then we immediately know there are exactly $l+1$ isogenies whenever $l \neq p$. For example, let us consider a curve $E/\mathbb{C}$ such that $\mathrm{End}(E) = \mathbb{Z}$. Its l-isogeny graph, that is the connected component of the graph of l-isogenies containing E, is $(l+1)$-regular, and cannot have loops, otherwise that would provide a non-trivial endomorphism of E of degree a power of l. Hence, the l-isogeny graph of E is an infinite $(l+1)$-tree, as pictured in Figure 3.3.

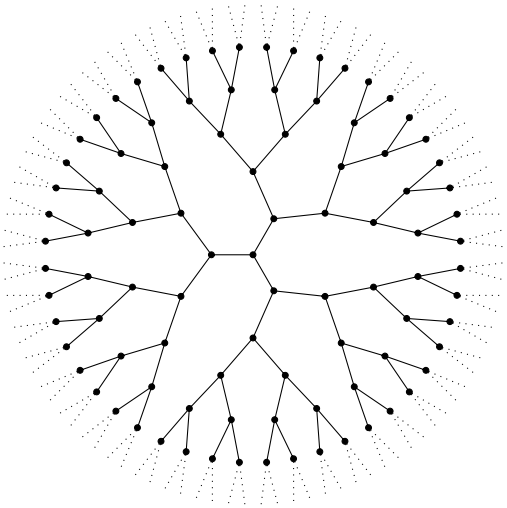


Figure 3.3: Infinite 2-isogeny graph of elliptic curves without complex multiplication.

3.4.2 $\mathrm{End}(E) \cong \mathcal{O} \subseteq \mathbb{Q}(\sqrt{-d})$

Here we focus on an isogeny graph of a given elliptic curve $E/\mathbb{K}$ with complex multiplication, that is a curve whose endomorphism ring is isomorphic to an order $\mathcal{O}$ in an imaginary quadratic field $\mathbb{Q}(\sqrt{-d})$. Exactly as in the complex case, and for the same reason, there are $l+1$ isogenies defined over $\overline{\mathbb{K}}$ with domain E, whenever the characteristic of the field does not divide l. Clearly we are interested in elliptic curves with complex multiplication which are defined on a finite field $\mathbb{F}_q$. When we think about curves over finite fields, some isogenies may only be defined on the algebraic closure $\overline{\mathbb{F}}_q$, and we would like to restrict our graphs to those isogenies that are defined over $\mathbb{F}_q$. The Frobenius morphism is a very useful tool in this search. Formally, the following result holds:

Proposition 3.32. *Let $\varphi : E(\overline{\mathbb{F}}_q) \to E/G(\overline{\mathbb{F}}_q)$ be an isogeny. Then φ is $\mathbb{F}_q$-rational if and only if $\pi(G) = G$.*

Since we are considering separable isogenies of degree l, also the kernel of the map, that is the subgroup G will have order l, from which it immediately follows $G \subseteq E[l]$. In order to computer for how many subgroups $G \subseteq E[l]$ of order l we have $\pi(G) = G$, we restrict the domain of the Frobenius to $E[l]$. Since $E[l] \cong \mathbb{Z}_l \times \mathbb{Z}_l$ we can imagine π as an element of $GL_2(\mathbb{F}_l)$, up to conjugation. Since all subgroups of $E[l]$ are cyclic, in particular our subgroups G will be of the form $G = \langle v \rangle$, for some $v \in E[l]$ of order l. Note that every non zero element of $E[l]$ has order l. So let us look for how many elements $v \in E[l]$ we have $\pi(v) = \lambda \cdot v$. In this way $G := \langle v \rangle$ defines an isogeny over $\mathbb{F}_q$ with domain E. To answer this question it is sufficient to diagonalize the Frobenius matrix. We thus are in one of the following four cases:

- π is not diagonalizable in $\mathbb{F}_l$, then E has no l-isogenies.
- π has one eigenvalue of (geometric) multiplicity one, i.e., it is conjugate to a non-diagonal matrix $\left(\begin{smallmatrix} \lambda & * \\ 0 & \lambda \end{smallmatrix}\right)$; then E has one l-isogeny.
- π has one eigenvalue of multiplicity two, i.e., it acts like a scalar matrix $\left(\begin{smallmatrix} \lambda & 0 \\ 0 & \lambda \end{smallmatrix}\right)$; then E has $l+1$ isogenies of degree l.
- π has two distinct eigenvalues, i.e., it is conjugate to a diagonal matrix $\left(\begin{smallmatrix} \lambda & 0 \\ 0 & \mu \end{smallmatrix}\right)$ with $\lambda \neq \mu$; then E has two l-isogenies.

Naturally, the number of eigenvalues of π depends on the factorization of its characteristic polynomial $x^2 - tx + q$ over $\mathbb{F}_l$, or equivalently on whether $\Delta_\pi = t^2 - 4q$ is a square modulo l. We can sum up the results found in the following proposition.

Proposition 3.33. *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over a finite field $\mathbb{F}_q$ of characteristic p, and let $l \neq p$ be a prime.*

1. *There are $l+1$ distinct isogenies of degree l with domain E defined over the algebraic closure $\overline{\mathbb{F}}_q$.*
2. *There are $0, 1, 2$ or $l+1$ isogenies of degree l with domain E defined over $\mathbb{F}_q$.*

In this way we manage to answer the first question we asked ourselves, namely how many isogenies defined over $\mathbb{F}_q$ do have E as a domain. But what about the global structure? Any curve $E/\mathbb{F}_q$ can be seen as the reduction modulo p of some curve $E/\mathbb{C}$; thus it must inherit the connectivity structure of the isogeny graph of $E/\mathbb{C}$. Let $E/\mathbb{F}_q$ have complex multiplication by an order $\mathcal{O}$ in a number field $K = \mathbb{Q}(\pi)$. Write $\mathcal{O}_K$ for the maximal

order of K, then we know that $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_K$. We have already seen that two elliptic curves are isogenous if and only if they have the same endomorphism algebra K; Kohel refined this as follows.

Proposition 3.34 (Horizontal and vertical isogenies)**.** *Let $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ be an isogeny of prime degree l, and let $\mathcal{O}, \mathcal{O}'$ be the orders corresponding to E, E'. Then, either $\mathcal{O} \subseteq \mathcal{O}'$ or $\mathcal{O}' \subseteq \mathcal{O}$, and one of the following is true:*

- *$\mathcal{O} = \mathcal{O}'$, in this case φ is said to be* horizontal.
- *$[\mathcal{O}' : \mathcal{O}] = l$, in this case φ is said to be* ascending.
- *$[\mathcal{O} : \mathcal{O}'] = l$, in this case φ is said to be* descending.

Proof. See [57, Prop. 21]. □

This result introduces the concept of depth in isogeny graphs of regular curves. To formalize it we introduce the following definition.

Definition 3.35 (p-adic valuation)**.** Let p be a prime number, the *p-adic order* or *p-adic valuation* of a non-zero integer n is the highest exponent v such that p^v divides n. The p-adic valuation of 0 is defined to be infinity. The p-adic valuation is commonly denoted with $v_p(n)$.

For a fixed prime l, Kohel defines a curve E to be *at the surface* if $v_l([\mathcal{O}_K : \mathrm{End}(E)]) = 0$. E is said to be *at depth d* if $v_l([\mathcal{O}_K : \mathrm{End}(E)]) = d$. The maximal depth is defined as $d_{\max} = v_l([\mathcal{O}_K : \mathbb{Z}[\pi]])$. Curves at depth $d_{\max}$ are said to be *at the floor (of rationality)*, and $d_{\max}$ is called the *height* of the graph of E. In this view an l-isogeny is said to be *horizontal* if it goes to a curve at the same depth, *descending* if it goes to a curve at greater depth, *ascending* if it goes to a curve at lesser depth.

Observe that vertical isogenies can only exist for primes that divide the conductor of $\mathbb{Z}[\pi]$, so the horizontal case is the generic one. But how many horizontal and vertical l-isogenies does a given curve have? To answer this question, let us first recall the Legendre symbol, with which we will be able to state the proposition that answers our question.

Definition 3.36 (Legendre symbol)**.** Let p be an odd prime number and a be an integer. We say that a is a quadratic residue modulo p if it is congruent to a perfect square modulo p, otherwise we say that it is a non-quadratic residue modulo p. The *Legendre symbol* is a function of a and p defined as

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p \text{ and } a \not\equiv 0 \pmod{p}, \\ -1 & \text{if } a \text{ is a non-quadratic residue modulo } p, \\ 0 & \text{if } a \equiv 0 \pmod{p}. \end{cases}$$

Proposition 3.37. *Let E be an elliptic curve over a finite field $\mathbb{K}$. Let $\mathcal{O} \subseteq \mathcal{O}_K$ be its endomorphism ring, f its conductor, Δ_K the discriminant of $\mathcal{O}_K$, π the Frobenius endomorphism, f_π the conductor of $\mathbb{Z}[\pi]$. Let l be a prime different from the characteristic of $\mathbb{K}$, then the types of degree l isogenies with domain E are as follows:*

- *If $l | f$ and $l \nmid (f_\pi / f)$, there is one ascending isogeny.*
- *If $l | f$ and $l | (f_\pi / f)$, there is one ascending isogeny and l descending ones.*

- *If $l \nmid f$ and $l \nmid (f_\pi/f)$, there are $1 + \left(\frac{\Delta_K}{l}\right)$ horizontal isogenies, where $\left(\frac{\Delta_K}{l}\right)$ represents the Legendre symbol.*
- *If $l \nmid f$ and $l|(f_\pi/f)$ there are $1+\left(\frac{\Delta_K}{l}\right)$ horizontal isogenies and $l-\left(\frac{\Delta_K}{l}\right)$ descending isogenies.*

Proof. See [57, Prop. 21]. □

We refer to the following table for a summary of the result obtained. The third column describes the types of isogenies which can occur: horizontal, ascending and descending, respectively.

		$\rightarrow$	$\uparrow$	$\downarrow$
$v_l(\Delta_\pi/\Delta_K) = 0$	$l \nmid [\mathcal{O}_K : \mathcal{O}]] \wedge l \nmid [\mathcal{O} : \mathbb{Z}[\pi]]$	$1 + \left(\frac{\Delta_K}{l}\right)$		
$v_l(\Delta_\pi/\Delta_K) > 1$	$l \nmid [\mathcal{O}_K : \mathcal{O}]] \wedge l \mid [\mathcal{O} : \mathbb{Z}[\pi]]$	$1 + \left(\frac{\Delta_K}{l}\right)$		$l - \left(\frac{\Delta_K}{l}\right)$
	$l \mid [\mathcal{O}_K : \mathcal{O}]] \wedge l \mid [\mathcal{O} : \mathbb{Z}[\pi]]$		1	l
	$l \mid [\mathcal{O}_K : \mathcal{O}]] \wedge l \nmid [\mathcal{O} : \mathbb{Z}[\pi]]$		1	

Table 3.1: Number and types of l-isogenies

The table above summarizes Proposition 3.37 in much easier terms and clarifies once and for all the structure of the isogeny graph associated with an elliptic curve with complex multiplication. To explain that, let us distinguish the cases:

- If $v_l(\Delta_\pi/\Delta_K) = 0$ then, since $\Delta_\pi = f_\pi^2 \Delta_K$, we have

$$\begin{aligned} v_l(\Delta_\pi/\Delta_K) = 0 &\iff v_l(f_\pi^2\Delta_K/\Delta_K) = 0 \\ &\iff v_l(f_\pi^2) = 0 \\ &\iff f_\pi^2 = 1 \\ &\iff \mathbb{Z}[\pi] = \mathcal{O}_K \end{aligned}$$

  Given a quadratic field K, we have already shown that $\mathbb{Z}[\pi] \subseteq \mathrm{End}(E) \subseteq \mathcal{O}_K$, so in this case every elliptic curve with $\mathrm{End}(E) \subseteq \mathcal{O}_K$ is forced to have $\mathrm{End}(E) \cong \mathcal{O}_K$, and therefore there will be only horizontal isogenies.

- If $v_l(\Delta_\pi/\Delta_K) > 1$, or, equivalently, if $\mathbb{Z}[\pi] \subsetneq \mathcal{O}_K$, there will also be vertical isogenies, in particular if E is not at the floor, there are $l + 1$ isogenies of degree l from E, in total. If E is at the floor, there are no descending l-isogenies from E. If E is at the surface, then there are $\left(\frac{\Delta_K}{l}\right) + 1$ horizontal l-isogenies from E (and no ascending l-isogenies). If E is not at the surface, there are no horizontal l-isogenies from E, and one ascending l-isogeny.

This theorem shows that, away from the surface, isogeny graphs just look like l-regular complete trees of bounded height, with l descending isogenies at every level except the floor.

Putting the pieces together, we see that graphs of ordinary curves have a very rigid structure: a cycle of horizontal isogenies and a tree of descending isogenies of height $v_l(f_\pi)$ (have a look at Figure 3.4).

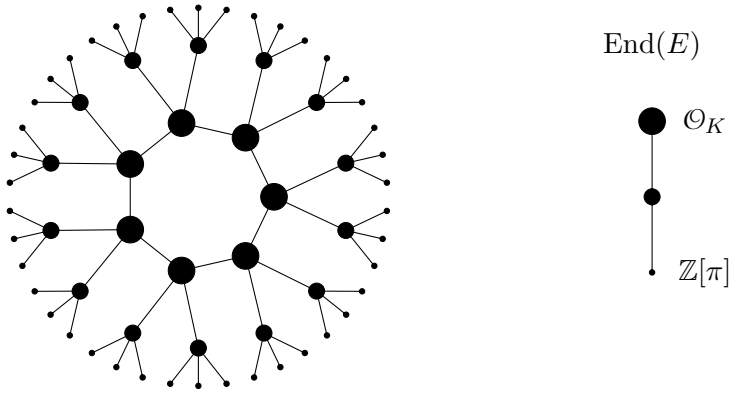


Figure 3.4: A volcano of 3-isogenies and the corresponding tower of orders.

The surface is the component of the graph that has a more varied structure:

(0) If $\left(\frac{\Delta_K}{l}\right) = -1$, there are no horizontal isogenies: the isogeny graph is just a complete tree of degree $l+1$ (in the graph theoretic sense) at each level but the last. We call this the *Atkin case*.

(1) If $\left(\frac{\Delta_K}{l}\right) = 0$, there is exactly one horizontal isogeny $\varphi : E \to E'$ at the surface. Since E' also has one horizontal isogeny, it necessarily is $\hat{\varphi}$, so the surface only contains two elliptic curves, each the root of a complete tree. We call this the *ramified case*.

(2) The case $\left(\frac{\Delta_K}{l}\right) = 1$ is arguably the most interesting one. Each curve at the surface has exactly two horizontal isogenies, thus the sub-graph made by curves on the surface is two-regular and finite, i.e., a cycle. Below each curve of the surface there are $l-1$ curves, each the root of a complete tree. We call this the *Elkies case*. Similarly we say that l is an *Elkies prime*.

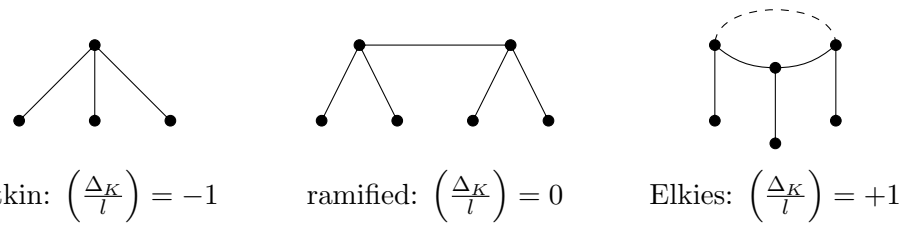


Figure 3.5: The three shapes of volcanoes of 2-isogenies of height 1.

The three cases are summarized in Figure 3.5. Their looks have justified the name *isogeny volcanoes* for them [32]; in the Elkies case, we call *crater* the cycle at the surface.
We are left with one last question: how large are these graphs? We know from the theory of complex multiplication that there is a bijection

$$\begin{aligned} \mathrm{Cl}(\mathcal{O}) &\longrightarrow \mathrm{Ell}_q(\mathcal{O}) \\ \text{Ideal class of } \mathfrak{a} &\longmapsto \text{Isomorphism class of } \mathfrak{a} \cdot E. \end{aligned}$$

This confirms what we already knew, that $|\mathrm{Ell}_q(\mathcal{O})| = h(\mathcal{O})$, but also lays the bases for showing the following proposition, which answers our question on the size of l-isogeny volcanoes.

Proposition 3.38. *Let $\mathcal{O}$ be a quadratic imaginary order, and assume that $\mathrm{Ell}_q(\mathcal{O})$ is non-empty. Let l be a prime such that $\mathcal{O}$ is l-maximal, i.e., such that l does not divide the conductor of $\mathcal{O}$. All l-isogeny volcanoes of curves in $\mathrm{Ell}_q(\mathcal{O})$ are isomorphic. Furthermore, one of the following is true.*

(0) If the ideal (l) is prime in $\mathcal{O}$, then there are $h(\mathcal{O})$ distinct l-isogeny volcanoes of Atkin type, with surface in $\mathrm{Ell}_q(\mathcal{O})$.

(1) If (l) is ramified in $\mathcal{O}$, i.e., if it decomposes as a square $\mathfrak{l}^2$, then there are $h(\mathcal{O})/2$ distinct l-isogeny volcanoes of ramified type, with surface in $\mathrm{Ell}_q(\mathcal{O})$.

(2) If (l) splits as a product $\mathfrak{l} \cdot \hat{\mathfrak{l}}$ of two distinct prime ideals, then there are $h(\mathcal{O})/n$ distinct l-isogeny volcanoes of Elkies type, with craters in $\mathrm{Ell}_q(\mathcal{O})$ of size n, where n is the order of $\mathfrak{l}$ in $\mathrm{Cl}(\mathcal{O})$.

But we can extract even more information from the group action. Note that (l) splits in $\mathcal{O}$ if and only if Δ is a non-zero square modulo l. This is equivalent to require that the Frobenius endomorphism splits modulo l, i.e., that

$$\pi^2 - t\pi + q = (\pi - \lambda)(\pi - \mu) \mod \ell$$

for two distinct eigenvalues λ, μ. It is immediate to verify that the eigenspaces associated to these eigenvalues are $E[(\pi - \lambda, l)]$ and $E[(\pi - \mu, l)]$, therefore ideals $\mathfrak{l} := (\pi - \lambda, l)$ and $\hat{\mathfrak{l}} := (\pi - \mu, l)$ define two subgroups of E to which we can associated two $\mathbb{F}_q$-rational isogenies.

$$\varphi_{\mathfrak{l}} : E \to \mathfrak{l} \cdot E$$
$$\varphi_{\hat{\mathfrak{l}}} : E \to \hat{\mathfrak{l}} \cdot E$$

Since $\mathfrak{l}\hat{\mathfrak{l}} = \hat{\mathfrak{l}}\mathfrak{l} = (l)$ we managed to characterize the prime ideals of Prop. 3.38. These classes are the inverse one of the other in $\mathrm{Cl}(\mathcal{O})$, and therefore $\varphi_{\mathfrak{l}}$ and $\varphi_{\hat{\mathfrak{l}}}$ are dual isogenies.

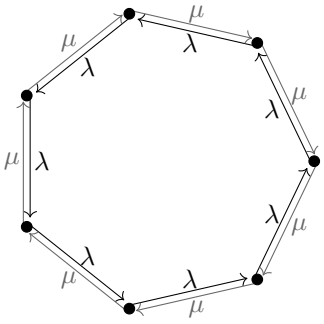


Figure 3.6: An isogeny cycle for an Elkies prime l, with edge directions associated with the Frobenius eigenvalues λ and μ.

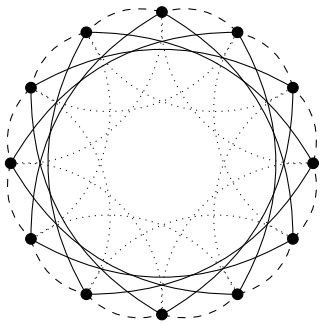


Figure 3.7: Graph of horizontal isogenies on 12 curves, with isogenies of three different degrees

We have already seen with Thm. **??** how invertible elements in $\mathrm{Cl}(\mathcal{O})$ preserve the endomorphisms structure of the curve, and therefore define horizontal isogenies in $\mathrm{Ell}(\mathcal{O})$. Hence, we see that the eigenvalues λ and μ define two opposite directions on the ℓ-isogeny crater, independent of the starting curve, as shown in Figure 3.6. The size of the crater is the order of $(\pi - \lambda, l)$ in $\mathrm{Cl}(\mathcal{O})$, and the set $\mathrm{Ell}_q(\mathcal{O})$ is partitioned into craters of equal size. In this way we have just built a basic Schreier graph with parameters $(\mathrm{Cl}(\mathcal{O}), S, \mathrm{Ell}_q(\mathcal{O}))$, where $S := \{\mathfrak{l}, \hat{\mathfrak{l}}\}$. In the sequel, we shall work with a larger edge set S, which will amount to *glue many isogeny craters together*, as shown in Figure 3.7.

We just introduced Cayley graphs constructed from isogeny craters and, unsurprisingly, they turn out to be expanders, provided we add enough edges to them.

Theorem 3.39. *Let $\mathcal{O}$ be a quadratic imaginary order, and assume that $\mathrm{Ell}_q(\mathcal{O})$ is non-empty. Let $\delta > 0$, and define the graph G on $\mathrm{Ell}_q(\mathcal{O})$ where two vertices are connected whenever there is a horizontal isogeny between them of prime degree bounded by $O((\log q)^{2+\delta})$. Then G is a regular graph and, under the generalized Riemann hypothesis for the characters of $\mathrm{Cl}(\mathcal{O})$, there exists an ε independent of $\mathcal{O}$ and q such that G is a two-sided ε-expander.*

Proof. See [50]. □

3.4.3 $\mathrm{End}(E) \cong \mathcal{O} \subseteq B_{p,\infty}$

The last case to be treated is that of elliptic curves whose endomorphism ring is isomorphic to a maximal order $\mathcal{O}$ in a quaternionic field $B_{p,\infty}$. As we have already noticed, this case only occurs with supersingular curves. In the following discussion we study the isogeny graph of a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, indeed every supersingular curve is isomorphic to one defined over this field. We stress that isogenies between these kind of curves are not necessarily defined over $\mathbb{F}_{p^2}$. For this graph we give below a bound to the number of nodes and show that, once fixed the definition field, there is only one connected component. This will turn out to be a Ramanujan graph, therefore with excellent expansion properties. If we restrict the study domain only to supersingular curves defined over $\mathbb{F}_p$, and we only consider isogenies over $\mathbb{F}_p$, then the resulting graph can be proved to be an isogeny volcano. Now let us explain in detail all these facts.

From the theory of quaternionic multiplication we immediately obtain a bound for the size of an isogeny graph of supersingular curves defined over $\mathbb{F}_{p^2}$. Thanks to Theorem 3.12 and to the Eichler mass formula, which quantify the class number of $B_{p,\infty}$, we obtain the exact size of the isogeny class.

Corollary 3.40. *Let S_{p^2} the set of all supersingular j-invariants in $\mathbb{F}_{p^2}$. For a prime $p > 3$ we have*

$$|S_{p^2}| = \left\lfloor \frac{p}{12} \right\rfloor + \begin{cases} 0 & \text{if } p \equiv 1 \mod 12, \\ 1 & \text{if } p \equiv 5, 7 \mod 12, \\ 2 & \text{if } p \equiv 11 \mod 12. \end{cases}$$

Proof. See [26]. □

Since isogeny graphs have isomorphism classes of elliptic curves as vertex, with this result just stated we have a bound on the size of a supersingular isogeny graph over $\mathbb{F}_{p^2}$. We can even say more about the global structure of the graph, as the following result shows:

Theorem 3.41. *Let $l \neq p$ be two primes. The l-isogeny graph of supersingular curves in $\bar{\mathbb{F}}_p$, is connected, $(l+1)$-regular, and has the Ramanujan property.*

Proof. See [67, 75, 76]. □

So, graphs of supersingular curves *defined over* $\mathbb{F}_{p^2}$ with l-isogenies, for a *single* prime $l \neq p$, define expander graphs; two examples of such graphs are shown in Figure 3.8. These graphs are called *supersingular isogeny graphs* and in general they are not isomorphic to a Cayley graph.

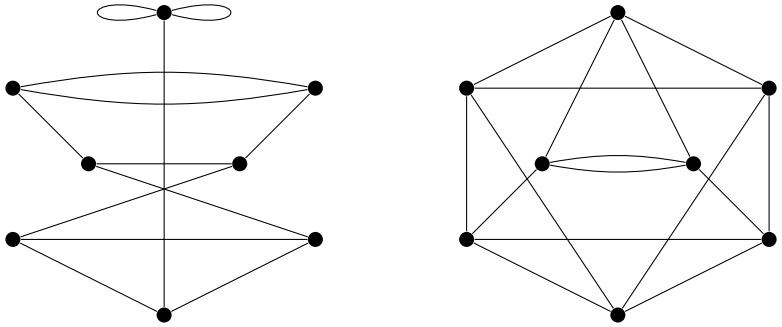


Figure 3.8: Supersingular isogeny graphs of degree 2 (left) and 3 (right) on $\mathbb{F}_{97^2}$.

Delfs and Galbraith [26] showed that if we limit to consider isogenies over $\mathbb{F}_p$, then all connected components are volcanoes, even in the supersingular case (where the depth is at most 1 at $l = 2$ and 0 otherwise): if we take a supersingular elliptic curve E over $\mathbb{F}_p$ with $p > 3$, we end up in Case 4 of Theorem 2.36. Thus we know according to the theorem that $\mathrm{End}_{\mathbb{F}_p}(E)$ is an order in $K := \mathbb{Q}(\sqrt{\pi_p})$ and its conductor is coprime with p. Since $\pi_p^2 + p = 0$ holds, we get $K = \mathbb{Q}(\sqrt{-p})$. Furthermore,

$$\mathbb{Z}[\pi_p] = \mathbb{Z}[\sqrt{-p}] = \mathbb{Z}\left[\frac{d+\sqrt{d}}{2}\right] \subseteq \mathrm{End}_{\mathbb{F}_p}(E) \subseteq \mathbb{Z}\left[\frac{d_K+\sqrt{d_K}}{2}\right] = \mathcal{O}_K$$

has to hold where the discriminant of $\mathrm{End}_{\mathbb{F}_p}(E)$ is $d = -4p$, $\mathcal{O}_K$ is the maximal order and d_K the fundamental discriminant of K. Due to the properties of the fundamental discriminant, we have $d = c^2 \cdot d_K$ where c is the conductor of $\mathbb{Z}[\pi_p]$ in $\mathcal{O}_K$. From these observations we can conclude that

- If $p \equiv 1 \pmod 4$, remembering Def. 3.1 we always get $d_K = d = -4p$, $\mathbb{Z}[\pi_p] = \mathcal{O}_K$ and hence $\mathrm{End}_{\mathbb{F}_p}(E) = \mathbb{Z}[\sqrt{-p}]$ for a supersingular elliptic curve E over $\mathbb{F}_p$.
- If $p \equiv 3 \pmod 4$, we get $d_K = -p$. Thus $\mathbb{Z}[\pi_p] = \mathbb{Z}[\sqrt{-p}]$ has conductor $c = 2$ in $\mathcal{O}_K = \mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$ and $\mathrm{End}_{\mathbb{F}_p}(E)$ must be one of those two orders.

In terms of isogeny-volcanoes we can say that we have at most two levels and we say E is on the surface (resp. E is on the floor) if $\mathrm{End}_{\mathbb{F}_p}(E) = \mathcal{O}_K$ (resp. $\mathrm{End}_{\mathbb{F}_p}(E) = \mathbb{Z}[-p]$). Note that for $p \equiv 1 \pmod 4$ surface and floor coincide.
In the supersingular case there are fewer possibilities for l-isogenies up and down than for ordinary volcanoes (though, even in the ordinary case tall volcanoes are quite rare). The following statement makes it clear:

Proposition 3.42. *Let φ be a non-horizontal isogeny between supersingular elliptic curves over $\mathbb{F}_p$. Then the degree of φ is divisible by 2.*

Proof. See [26, Lem. 2.2]. □

Therefore we have no isogenies of odd prime degree going up or down in this graph. To determine how many isogenies and how many nodes there are we need some theory about the ideal class group. We recall the relevant results in the following statements. For further detail we refer to [26]. First we can make an observation about the number of $\mathbb{F}_p$-isomorphism classes of supersingular elliptic curves over $\mathbb{F}_p$ with a given j-invariant, based on the following proposition.

Proposition 3.43. *Let $p > 3$ be a prime and $j \in \mathbb{F}_p$. Define $C_{p,j}$ as the set of $\mathbb{F}_p$-isomorphism classes of elliptic curves defined over $\mathbb{F}_p$ with j-invariant j. Then we get*

$$|C_{p,j}| = \begin{cases} 6 & \text{if } j = 0 \text{ and } p \equiv 1 \pmod 3 \\ 4 & \text{if } j = 1728 \text{ and } p \equiv 1 \pmod 4 \\ 2 & \text{otherwise} \end{cases}$$

Proof. It follows directly from [10, Theorem 2.2]. □

Since we know that for an elliptic curve E over $\mathbb{F}_p$, with

$$\begin{aligned} j(E) = 0: &\quad E \text{ is supersingular} \iff p \equiv 2 \pmod 3 \\ j(E) = 1728: &\quad E \text{ is supersingular} \iff p \equiv 3 \pmod 4 \end{aligned}$$

holds, we can deduce from Proposition 3.43 that given a supersingular j-invariant j there are always exactly two $\mathbb{F}_p$-isomorphism classes of elliptic curves over $\mathbb{F}_p$ with this j-invariant. The following result states more precisely the whole structure of a supersingular isogeny graph.

Theorem 3.44. *Let $p > 3$ be a prime.*

1. *$p \equiv 1 \pmod 4$: There are $h(-4p)$ (with this notation we mean $h(\mathbb{Q}(\sqrt{-4p}))$ supersingular elliptic curves over $\mathbb{F}_p$, all having the same endomorphism ring $\mathbb{Z}[\sqrt{-p}]$. From each of these there is one outgoing $\mathbb{F}_p$-rational horizontal 2-isogeny as well as two horizontal l-isogenies for every prime $l > 2$ with $(\frac{-p}{l}) = 1$.*
2. *$p \equiv 3 \pmod 4$: There are two levels in the supersingular isogeny graph. From each vertex there are two horizontal l-isogenies for every prime $l > 2$ with $(\frac{-p}{l}) = 1$.*
   - *If $p \equiv 7 \pmod 8$, on each level $h(-p)$ vertices are situated. Surface and floor are connected $1:1$ with 2-isogenies and on the surface we also have two horizontal 2-isogenies from each vertex.*
   - *If $p \equiv 3 \pmod 8$, we have $h(-p)$ vertices on the surface and $3h(-p)$ on the floor. Surface and floor are connected $1:3$ with 2-isogenies, and there are no horizontal 2-isogenies.*

Proof. See [26, Thm. 2.7]. □

This provides a structure similar to that of the ordinary isogeny volcano, except that in our case we have no more than two levels and for $l > 2$ only exactly two outgoing isogenies from each elliptic curve (if any). This result can be used to adapt the algorithms from the ordinary case that rely on the volcano structure.

We present a few small examples of the irregular structure of the full supersingular isogeny

graph $X(\mathbb{F}_{p^2}, l)$. After that we display, for the same examples, the graphs $X(\mathbb{F}_p, l)$ which have a much more regular structure. For the examples we use the primes $p = 101, 103$ and 83, one for each of the different cases that occur, respectively. To demonstrate the two occurring structures we build the graphs for isogeny degrees $l = 2$ and the smallest prime $l > 2$ in each case for that isogenies exist. We refer to [26] for further details.

Remark 3.45 (Edge representation)**.** Until now we have never encountered isogeny graphs in which a curve E has j-invariant $j \in \{0, 1728\}$. This allowed us to simplify the treatment by drawing the graphs as undirected, and in fact in the applications we will find a way to exclude this from happening. However, limited to the examples below, we will encounter isogeny graphs in which inevitably $j \in \{0, 1728\}$. Note that for $j(E) \equiv 0$ and $j(E) \equiv 1728 \pmod p$ there are respectively three and two non-equivalent isogenies mapping from E to another curve E, but their dual isogenies are all equivalent. This is due to the fact that $|\operatorname{Aut}(E)| = 6$ or $|\operatorname{Aut}(E)| = 4$ in these cases. If $\varphi : E(\overline{\mathbb{K}}) \to E'(\overline{\mathbb{K}})$ is an isogeny and $\rho \in \operatorname{Aut}(E)$, then $\varphi \circ \rho$ may not be equivalent (i.e., have the same kernel) as ψ, whereas the dual of $\varphi \circ \rho$ is $\hat{\rho} \circ \hat{\varphi}$, so this is equivalent to the dual of φ (they have the same kernel). For these reasons in the following examples we will denote all edges by specifying their direction and emphasizing these multiple isogenies using a single arrow together with an integer to indicate the multiplicity.

Remark 3.46 (Node representation)**.** In the previous examples we have always depicted an isogeny graph not giving importance to its vertices, that is ignoring the class of elliptic curves that a given vertex represented. This reason was due to the fact that we were interested to catch the overall structure of the graph, rather than the nature of individual nodes. In the examples that follow we instead focus on nodes, and on how these ones change as we narrow the base field from $\mathbb{F}_{p^2}$ to $\mathbb{F}_p$. For this reason we denote each node with the associated j-invariant. We have already seen that this way of representing the graph is wrong. We fix the representation in the following way: in the case where two curves are isomorphic over an extension of greater degree with respect to the one under analysis, we will in any case label their vertices with their j-invariant, and we will represent this vertex several times.

Example 3.47. For the first example we take $p = 101$, so that $p \equiv 1 \pmod 4$. Thanks to Cor. 3.40, we expect $\lfloor \frac{101}{12} \rfloor + 1 = 9$ supersingular j-invariants in $\mathbb{F}_{p^2}$. In the next figure we show how they are connected using 2-isogenies. The nodes labeled α and $\overline{\alpha}$ represent j-invariants in $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$ where $\overline{\alpha}$ is the conjugate of α. The graph can be easily computed with help of modular polynomials.

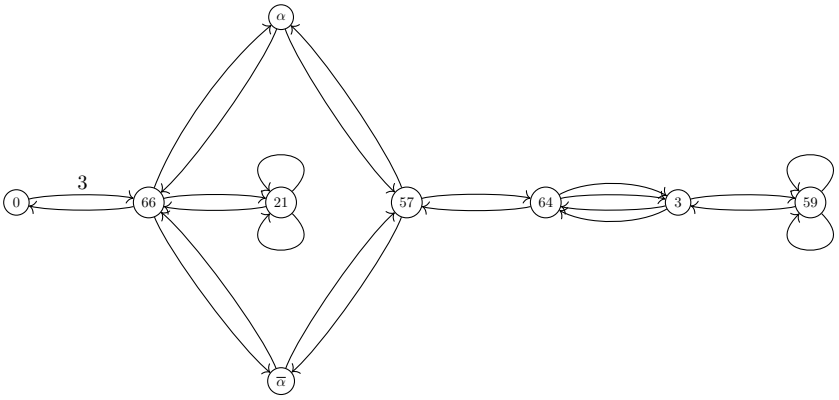


Figure 3.9: Supersingular 2-Isogeny Graph over $\overline{\mathbb{F}}_{101}$

In $X(\mathbb{F}_p, l)$ we will have $h(-4p) = 14$ nodes which are supersingular elliptic curves over $\mathbb{F}_p$ with endomorphism ring $\mathbb{Z}[\sqrt{-101}]$. There will be only one outgoing 2-isogeny from each curve, so naturally the graph can not be connected. It can be seen in the following figure.

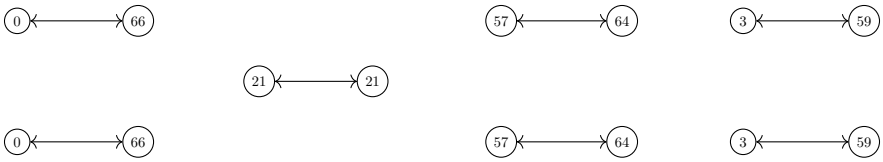


Figure 3.10: $\mathbb{F}_p$-Rational Supersingular Isogeny Graph $X(\mathbb{F}_{101}, 2)$

It is notable that in this graph there are fewer connecting isogenies than in the full graph before. For example, in the first graph we have two isogenies going from the node 64 to the node 3 and two ones back, which are all missing in the new graph. This is due to the fact that those isogenies are not defined over $\mathbb{F}_p$, so they are not computed as edges in $X(\mathbb{F}_p, 2)$. Likewise the two loops from 59 to itself are isogenies over $\mathbb{F}_{p^2}$ that are dual to each other, whereas the loop at 21 is a $\mathbb{F}_p$-rational isogeny that is its own dual. We also observe, thanks to Prop. 3.43, that all nodes are duplicated. This is due to the fact that in the graph described by Figure 3.9, each node associated with a j-invariant encloses elliptic curves which are isomorphic over an extension of $\mathbb{F}_{101}$, therefore it is correct to represent them as a single vertex in this figure, but it is not correct to do the same in Figure 3.10.

Example 3.48. For higher isogeny degree, the number of outgoing isogenies from each vertex grows, so the graph becomes complicated to draw. Here we take $l = 3$, as $\left(\frac{-p}{3}\right) = 1$.

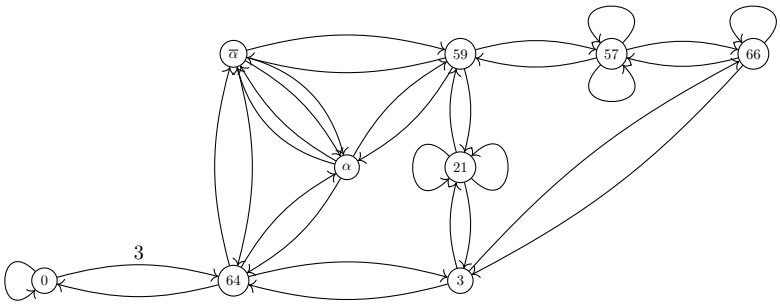


Figure 3.11: Supersingular Isogeny Graph $X(\overline{\mathbb{F}}_{101}, 3)$

Despite the complicated picture of the full graph, the graph over $\mathbb{F}_p$ becomes just a big circle. In particular, it is already fully connected. This is because the ideal class group of $\mathbb{Q}(\sqrt{-101})$ is generated by a prime ideal of norm 3.

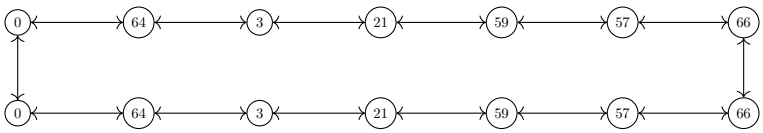


Figure 3.12: Rational Supersingular Isogeny Graph $X(\mathbb{F}_{101}, 3)$

Again we can see how the isogenies from the full graph that are defined over $\mathbb{F}_{p^2}$ vanish in the rational graph, and the single loops become isogenies from an elliptic curve to its quadratic twist.

Example 3.49. Let $p = 83$, so that $p \equiv 3 \pmod 4$ and $p \equiv 3 \pmod 8$. The full graph will have $\lfloor \frac{83}{12} \rfloor + 2 = 8$ vertices. Again we have two j-invariants $\alpha, \overline{\alpha} \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$. The full 2-isogeny graph has the following structure.

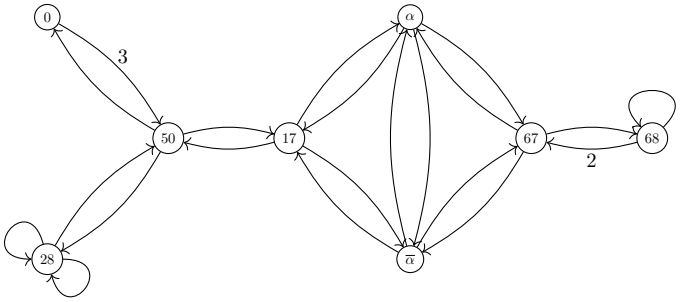


Figure 3.13: Supersingular Isogeny Graph $X(\overline{\mathbb{F}}_{83}, 2)$

In the graph over $\mathbb{F}_p$ we get $h(-p) = 3$ supersingular elliptic curves on the surface and $h(-4p) = 9$ ones on the floor. In the next figure we can see how 2-isogenies connect floor and surface as explained in the last case of Theorem 3.44.

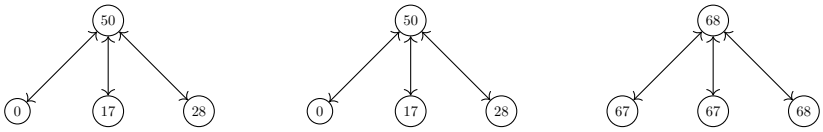


Figure 3.14: Rational Supersingular Isogeny Graph $X(\mathbb{F}_{83}, 2)$

Example 3.50. If we repeat the procedure for $l = 3$, the full graph looks like this.

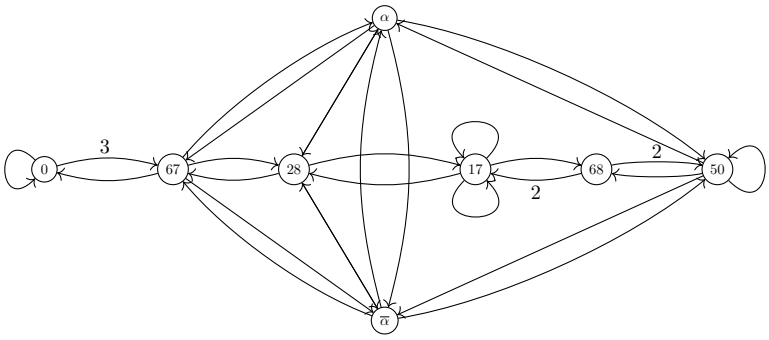


Figure 3.15: Supersingular Isogeny Graph $X(\overline{\mathbb{F}}_{83}, 3)$

And in the graph over $\mathbb{F}_p$ we get two isogeny circles, one on the floor and one on the surface.

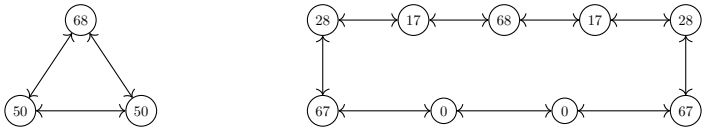


Figure 3.16: Rational Supersingular Isogeny Graph $X(\mathbb{F}_{83}, 3)$

Example 3.51. Our example here is $p = 103$ where $p \equiv 3 \pmod 4$ and $p \equiv 7 \pmod 8$. we expect $\lfloor \frac{103}{12} \rfloor + 1 = 9$ supersingular j-invariants in $\mathbb{F}_{p^2}$. In this case we have four nodes in $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$.

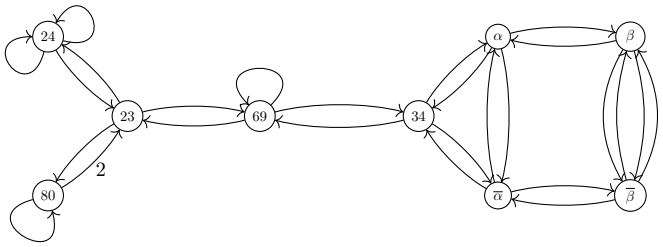


Figure 3.17: Supersingular Isogeny Graph $X(\overline{\mathbb{F}}_{103}, 2)$

The 2-isogeny graph over $\mathbb{F}_p$ in this case is connected: we have $h(-p) = 5$ supersingular elliptic curves on the surface and also 5 on the floor, since they are in $1:1$ correspondence.

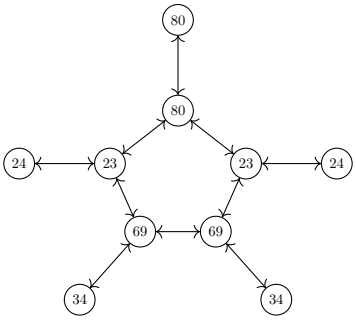


Figure 3.18: Rational Supersingular Isogeny Graph $X(\mathbb{F}_{103}, 2)$

Example 3.52. The smallest prime $l > 2$ with $(\frac{-103}{l}) = 1$ is $l = 7$. In the full graph every vertex has eight outgoing isogenies so it is not nice to draw. The sub-graph of $X(\overline{\mathbb{F}}_{103}, 7)$ only consisting of j-invariants in $\mathbb{F}_{103}$ is presented in the next figure, so it can be compared to $X(\mathbb{F}_{103}, 7)$ below.

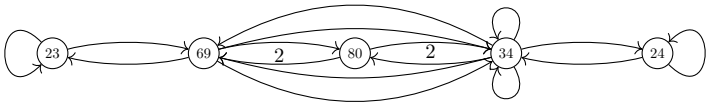


Figure 3.19: Subgraph of Supersingular Isogeny Graph $X(\overline{\mathbb{F}}_{103}, 7)$

Again we get two isogeny cycles such that both floor and surface are fully connected when we draw the graph $X(\mathbb{F}_{103}, 7)$. This is because the ideal class group is cyclic and generated by a prime ideal of norm 7.

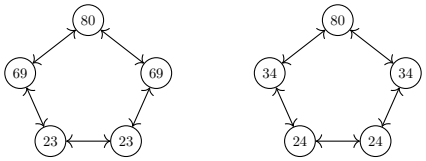


Figure 3.20: Rational Supersingular Isogeny Graph $X(\mathbb{F}_{103}, 7)$

Chapter 4

Isogeny-Based Cryptography

Now we present the main algorithms based on isogenies of elliptic curves. We begin by describing the work of Couveignes [23], which dates back to 1997, where he proposed the first isogeny-based algorithm using ordinary elliptic curves with the same endomorphism ring $\mathcal{O}$. His work was not published until ten years later. In 2006, Rostovtsev and Stolbunov independently propose a key-exchange protocol which is essentially the same as the one described by Couveignes; we therefore refer to this protocol as Couveignes-Rostovtsev-Stolbunov, or CRS for short. We describe this protocol in the first section. Despite the innovative ideas, there are still some problems: it is difficult to find an efficient implementation, furthermore Childs, Jao and Soukharev manage to construct a subexponential attack to the protocol. This attack strongly relies on the fact that $\mathrm{Cl}(\mathcal{O})$ is commutative, hence indirectly on the fact that $\mathcal{O}$ is commutative. While this may be tolerable (e.g., classical subexponential factorization methods have not ended the widespread use of RSA), a much bigger concern is that the scheme is unacceptably slow.

In the second section we show how these weaknesses have led Jao and De Feo to consider the use of supersingular elliptic curves, whose full ring of endomorphisms is an order in a quaternion algebra; in particular it is non-commutative. Their resulting key-agreement scheme goes under the name SIDH [49], and was first published in 2011. As we will see, SIDH is not just the Couveignes-Rostovtsev-Stolbunov scheme in which one substitutes supersingular elliptic curves for ordinary elliptic curves.

In the third section we show how Castryck, Lange, Martindale, Panny and Renesn in 2018 managed to make the CRS computationally feasible [15]. The key to achieve this result is to restrict to supersingular elliptic curves over a prime field $\mathbb{F}_p$. Instead of the full ring of endomorphisms, which is non-commutative, one should consider the sub-ring of $\mathbb{F}_p$-rational endomorphisms, which is again an order $\mathcal{O}$ in an imaginary quadratic field. In this way CSIDH does not grant protection from the Childs-Jao-Soukharev attack, but solves the main problem of CRS, that is its inefficiency. The section continues with a detailed analysis of this protocol. Since the ultimate purpose of this discussion is to present CSIDH and discuss its security, the discussion of Couveignes-Rostovtsev-Stolbunov and SIDH is reduced to what is necessary for the understanding and contextualization of CSIDH.

4.1 CRS

The Couveignes-Rostovtsev-Stolbunov protocol arises from the work of Couveignes regarding the so-called homogeneous spaces. We summarize the related theory below.

4.1.1 Homogeneous Spaces

Definition 4.1 (Principal homogeneous space)**.** A *principal homogeneous space* (PHS) for an abelian group G is a set X equipped with a freely transitive (i.e. regular) action of G. That is, for any P and Q in X, there exists a unique g in G such that $Q = g \cdot P$. Equivalently, for every P in X, the map $\varphi_P : G \to X$ which sends g to $g \cdot P$, is a bijection.

Remark 4.2. Schreier graphs are a natural consequence of principal homogeneous spaces. The idea of Rostovtsev and Stolbunov will be precisely to improve the Couveignes protocol, introducing these graphs.

Example 4.3. A trivial example of a principal homogeneous space is a group acting on itself: fixed a group G, let $X := G$ and define the action

$$\begin{aligned} G \times G &\to G \\ (g, a) &\mapsto g \cdot a \end{aligned}$$

Example 4.4. The classic example of a non-trivial homogeneous space is a vector space on $\mathbb{R}^2$ which acts by translation on its underlying affine space.

Example 4.5. In the notations of Theorem **??**, the set $\mathrm{Ell}_q(\mathcal{O})$ is a principal homogeneous space for the class group $\mathrm{Cl}(\mathcal{O})$, via the map

$$\begin{aligned} \mathrm{Cl}(\mathcal{O}) \times \mathrm{Ell}_q(\mathcal{O}) &\to \mathrm{Ell}_q(\mathcal{O}) \\ (\mathfrak{a}, E) &\mapsto \mathfrak{a} \cdot E \end{aligned}$$

as the theorem itself shows.

Given a homogeneous space, there are several algorithmic problems one would like to consider. We focus on the following ones.

Definition 4.6 (Vectorization problem)**.** Let X be a PHS over a group G and let P, Q be elements of X. The associated *vectorization problem* consists in finding the unique $g \in G$ such that $Q = g \cdot P$.

Definition 4.7 (Parallelization problem)**.** Let X be a PHS over a group G and let $P, A, B \in X$ in a way that there exist $g_1, g_2 \in G$ such that $A = g_1 \cdot P$ and $B = g_2 \cdot P$. The associated *parallelization problem* consists in finding the unique $S \in G$ such that $S = (g_1 g_2) \cdot P$. Note that in this case $S = g_1 \cdot B = g_2 \cdot A$.

The notation of the previous two definitions makes sense in the context of Example 4.4: vectorization consists in computing the displacement vector between two points P and Q in $\mathbb{R}^2$, while parallelization consists in completing the parallelogram with vertices P, A and B. We refer to Figure 4.1 for a geometric perspective.

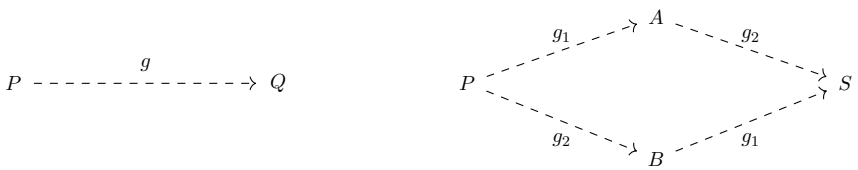


Figure 4.1: Vectorization on the left and parallelization on the right The dashed arrows denote the actions of the unknown group elements g, g_1 and g_2.

We are interested in principal homogeneous spaces for which the previous problems are difficult, but at the same time we would like that other tasks are easy to solve. This directly leads us to the following definition.

Definition 4.8 (Hard homogeneous space)**.** Let G be an abelian group, and X a set which is a principal homogeneous space over G. We say that X is a *hard homogeneous space* (HHS) over G if:

- The following tasks are easy to solve (e.g. polynomial time):
  - Compute the group operations in G.
  - Sample randomly from G with (close to) uniform distribution.
  - Decide validity and equality of a representation of elements of X.
  - Compute the action of a group element $g \in G$ on some $x \in X$.
- The following tasks are hard to solve (e.g. not polynomial time):
  - Solve the vectorization problem.
  - Solve the parallelization problem.

This is a very natural object to study from the point of view of cryptography. Indeed, any such hard homogeneous space leads in a quite natural and elegant manner to cryptographic schemes for authentication and key-exchange: given two participants Alice and Bob, their private keys are random elements $a, b \in G$. Their public keys are $a \cdot x_0$ and $b \cdot x_0$, where $x_0 \in X$ is a public parameter set in advance. The common secret will therefore be

$$a \cdot (b \cdot x_0) = (ab) \cdot x_0 = (ba) \cdot x_0 = b \cdot (a \cdot x_0)$$

The private keys are protected by the difficulty of the vectorization problem, while the public key is protected by the parallelization one. A diagram which sums up the operations of this protocol is proposed in Table 4.1.

Public parameters	A Hard Homogeneous Space (G, X). An element $x_0 \in X$.	
	Alice	**Bob**
Pick random secret	$a \in G$	$b \in G$
Compute public data	$g_a = a \cdot x_0$	$g_b = b \cdot x_0$
Exchange data	$g_a \longrightarrow$	$\longleftarrow g_b$
Compute shared secret	$g_{ab} = a \cdot (g_b)$	$g_{ab} = b \cdot (g_a)$

Table 4.1: Key exchange protocol based on a hard homogeneous space.

Couveignes had the idea to use isogeny graphs of ordinary elliptic curves to define a (conjectured) hard homogeneous space that does not base its security on the discrete logarithm problem: the idea is to use as hard homogeneous space the craters of the set $\mathrm{Ell}(\mathcal{O})$ of elliptic curves with endomorphism ring an order $\mathcal{O}$ in an imaginary quadratic field, for a given prime field $\mathbb{F}_q$. We then take $\mathrm{Cl}(\mathcal{O})$ as a group. The pair $(\mathrm{Cl}(\mathcal{O}), \mathrm{Ell}(\mathcal{O}))$ forms an hard homogeneous space, so we can apply the construction defined above. However, given a generic element of $\mathrm{Cl}(\mathcal{O})$, the best algorithm to evaluate its action on $\mathrm{Ell}_q(\mathcal{O})$ has subexponential complexity in q, making the protocol infeasible.

Remark 4.9. Instead of choosing a unique generic element of G, we could choose many *small* elements and consider their product to obtain our private key: consider the Schreier graph described in Example 3.28, where $X := \mathbb{Z}_{13}^*$ and the group acting on that set is a subgroup of $\mathbb{Z}_{13}$, more precisely it is the symmetric subgroup $S := D \cup D^{-1}$, with $D := \{2, 3, 5\}$. Let us assume for the moment that the group action meets the requirements to be an HHS. We can use this structure to define a protocol like the one just presented. Compared to the protocol defined above, we make a variation on the construction of the private key: instead of taking a random element in S, we choose a sequence $\rho = (\sigma_1, \ldots, \sigma_m) \in D^*$. We therefore observe that such a sequence, together with a *starting vertex* $g \in G$, defines a walk in the Schreier graph (S, G) by starting in g, and successively taking the edges corresponding to the labels in ρ. We write $\rho(g)$ for the vertex where the walk defined by ρ and g ends. This is simply an alternative way to randomly sample an element of G. Observe that, for any $g \in G$

$$\begin{aligned}\rho(g) &= (\sigma_1 \cdots \sigma_m) \cdot g \\ &= (\sigma_1 \cdots \sigma_{m-1})\sigma_m \cdot g \\ &= (\sigma_1 \cdots \sigma_{m-1}) \cdot g^{\sigma_m} \\ &= \exp_g \left(\prod \sigma_i \right)\end{aligned}$$

Hence, the order of the steps in a route does not matter: what counts is only how many times each element of D appears in ρ. Now suppose that the two participants of the protocol, Alice and Bob, choose as private keys $\rho_A := \{2, 3, 2, 5\}$ $\rho_B := \{3, 3, 5, 2\}$. The key exchange works as described in Figure 4.2.

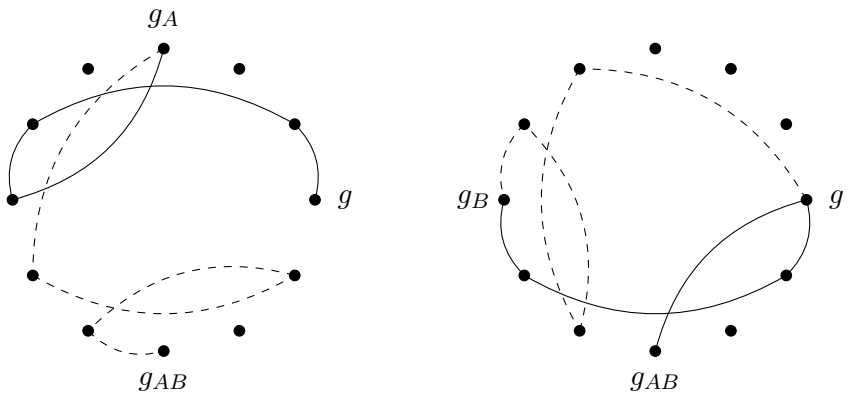


Figure 4.2: Example of key exchange on the Schreier graph of Figure 3.2. Alice's route is represented by continuous lines, Bob's route by dashed lines. On the left, Bob computes the shared secret starting from Alice's public data. On the right, Alice does the analogous computation.

We immediately realize that this protocol is nothing else than the classical Diffie-Hellman protocol on the group G, presented in a twisted way.

While this example instance is of no practical interest, its instantiation using a Schreier graph of the HHS $\mathrm{Ell}_q(\mathcal{O})$ yields a usable variant of Couveignes' key exchange. We fix a set S of small norm representatives of ideal classes of $\mathrm{Cl}(\mathcal{O})$, corresponding to small degree isogenies between curves in $\mathrm{Ell}_q(\mathcal{O})$. Instead of uniformly sampling secrets from $\mathrm{Cl}(\mathcal{O})$, we sample non-backtracking random walks in the Schreier graph of $(\mathrm{Cl}(\mathcal{O}), S, \mathrm{Ell}_q(\mathcal{O}))$, and exchange j-invariants as public data. The walks can be computed efficiently as a

composition of small degree isogenies, and, assuming the graph is an expander and the walks are long enough, they approach the uniform distribution on $\mathrm{Ell}_q(\mathcal{O})$. This leads us to the Couveignes-Rostovtsev-Stolbunov cryptosystem.

4.1.2 The Protocol

With this observation, we can give a key exchange protocol based on random walks in graphs of horizontal isogenies.

The protocol never explicitly computes $\mathcal{O}$, instead, it determines parameters in the following order: first it computes a *large enough* finite field $\mathbb{F}_q$. Then a curve E defined over $\mathbb{F}_q$ and, given π the Frobenius endomorphism, its discriminant $D_\pi = t_\pi^2 - 4q$ of E (through point counting, and it is verified that it contains a *large enough* prime factor). Note that t_π denotes the Frobenius trace. We compute a set $L = \{l_1, \ldots, l_m\}$ of primes that split in $\mathbb{Z}[\pi]$, i.e. such that $\left(\frac{D_\pi}{l_i}\right) = 1$, and for each prime l_i we take note of the factorization

$$\pi^2 - t_\pi \pi + q = (\pi - \lambda_i)(\pi - \mu_i) \pmod{l_i}$$

Finally one of the roots, say λ_i, is chosen arbitrarily as *positive direction*.
We stress that the condition on the l_i's guarantees that each graph of l_i-isogenies on $\mathrm{Ell}_q(\mathcal{O})$ is 2-regular. The choice of a *positive direction* allows us to *orient* the graph, by associating to λ_i the isogeny with kernel $E[l_i] \cap \ker(\pi - \lambda_i)$. The key exchange now proceeds like the ordinary Diffie-Hellman protocol:

1. (Setup) To set up the cryptosystem we fix a prime q, an elliptic curve $E/\mathbb{F}_q$ with order $\mathcal{O}$ and such that D_π contains a large prime factor, and a set L mades up of primes that split in $\mathcal{O}$.

2. (Key Generation) Alice chooses a random walk $\rho_A \in L^*$ made of steps in L along the positive direction she has chosen for each l, ending in $E_A = \rho_A(E)$. Note that E_A only depends on how many times each l_i appears in ρ_A, and not on their order. Bob does the same, choosing a random walk ρ_B and computing $E_B = \rho_B(E)$.

3. (Key Exchange) Alice and Bob exchange E_A and E_B. Alice computes the shared secret $\rho_A(E_B)$ and Bob computes the shared secret $\rho_B(E_A)$.

The protocol is summarized in Table 4.2.

Public parameters	An elliptic curve E over a finite field $\mathbb{F}_q$, D_π, the discriminant of the Frobenius endomorphism of E, A set of primes $L = \{l_1, \ldots, l_m\}$ such that $\left(\frac{D_\pi}{l_i}\right) = 1$, A *Frobenius eigenvalue* λ_i for each l_i,	
	Alice	**Bob**
Pick random secret	$\rho_A \in L^*$	$\rho_B \in L^*$
Compute public data	$E_A = \rho_A(E)$	$E_B = \rho_B(E)$
Exchange data	$E_A \longrightarrow$	$\longleftarrow E_B$
Compute shared secret	$E_{AB} = \rho_A(E_B)$	$E_{AB} = \rho_B(E_A)$

Table 4.2: Couveignes-Rostovtsev-Stolbunov key exchange protocol.

4.1.3 Security

We conclude this section with a discussion on the security of the Couveignes-Rostovtsev-Stolbunov protocol. For now we do not go into the detail of the attacks we present, and reserve its discussion for later sections. For the moment we consider appropriate only to keep in mind the existence of such an attack. All the protocol's security rests on the *isogeny path problem*: given E and E_A, find an isogeny $\varphi : E \to E_A$ of smooth order. To be safe against exhaustive search and meet in the middle attacks, the set $\mathrm{Ell}_q(\mathcal{O})$ must be large. However, some isogeny classes are much smaller than average, this is why we also need check that D_π has a large prime factor.

In 2010, Childs, Jao and Soukharev [17] show that breaking the Couveignes-Rostovtsev-Stolbunov scheme amounts to solve an instance of the abelian hidden-shift problem, for which quantum algorithms with subexponential time complexity are known to exist [58, 79]. While this attack may be tolerable, the main problem of the protocol is that it is unacceptably slow: despite recent clever speed-ups due to De Feo, Kieffer, and Smith [27, 54], several minutes are needed for a single key exchange at a presumed classical security level of 128 bits.

4.2 SIDH

The attack due to Childs-Jao-Soukharev strongly relies on the fact that $\mathrm{Cl}(\mathcal{O})$ is commutative, hence indirectly on the fact that $\mathcal{O}$ is commutative. This led Jao and De Feo [52] to consider the use of supersingular elliptic curves. Their resulting (interactive) key-agreement scheme goes under the name "Supersingular Isogeny Diffie-Hellman" (SIDH). The current state-of-the-art implementation is SIKE [51], which was recently submitted to the NIST competition on post-quantum cryptography [71]. We briefly present the protocol.

The use of supersingular elliptic curves has two advantages:

- the endomorphism ring is isomorphic to an order in a quaternion algebra, in particular, it is not commutative, and consequently the class group so. In any case, unlike CRS, we no longer have the action of a group on a curve in a given isogeny class, but we will exploit another solution to allow the key exchange.
- The underlying l-isogeny graph is no longer cyclic, but is a $(l+1)$-regular graph, which is both connected and expander. We observe that, since the graph has $O(p)$ vertices, a sequence of $\log p$ isogenies of degree l starting from any vertex leads to a uniform distribution within the isogeny class, as Prop. 3.25 shows.

The main idea is the following: Alice and Bob choose two random walks in two distinct l-isogeny graph, on the same vertex set of all supersingular j-invariants defined over $\mathbb{F}_{p^2}$ (consider in this regard Figure 3.8), they publish their curves and then complete the protocol analogously to Diffie-Hellman. The problem is that now there is not a commutative group action, therefore it is not immediate to allow these two paths to commute.

Remark 4.10. Consider a finite field $\mathbb{F}_q$, an elliptic curve E over $\mathbb{F}_q$, and two isogenies $\alpha : E \to E/\langle A \rangle$ and $\beta : E \to E/\langle B \rangle$, for respectively two subgroup $\langle A \rangle, \langle B \rangle$ of $E(\mathbb{F}_q)$. We want to define two isogenies $\alpha' : E/\langle B \rangle \to E/\langle A, B \rangle$ and $\beta' : E/\langle A \rangle \to E/\langle A, B \rangle$, in such

a way that the image curves are the same (up to isomorphisms). Observe that

$$\begin{aligned}\mathrm{Im}(\beta') = E/\langle A, B\rangle &\iff \mathrm{Im}(\beta' \circ \alpha) = E/\langle A, B\rangle \\ &\iff \ker(\beta' \circ \alpha) = \langle A, B\rangle \\ &\iff \ker(\beta') = \alpha(\langle B\rangle)\end{aligned}$$

Where the last step is justified by the fact that by hypothesis $\ker(\alpha) = \langle A\rangle$. The same reasoning applies to α'. A corresponding diagram is shown in Figure 4.3.

$$\begin{array}{rcc} \ker\alpha = \langle A\rangle & E & \xrightarrow{\quad\alpha\quad} E/\langle A\rangle \\ \ker\beta = \langle B\rangle & \Big\downarrow{\scriptstyle\beta} & \Big\downarrow{\scriptstyle\beta'} \\ \ker\alpha' = \langle\beta(A)\rangle & & \\ \ker\beta' = \langle\alpha(B)\rangle & E/\langle B\rangle & \xrightarrow[\alpha']{} E/\langle A, B\rangle \end{array}$$

Figure 4.3: Commutative isogeny diagram.

We will use exactly this idea to allow us to reach a common secret in the supersingular graph of SIDH, let us see how.

4.2.1 The Protocol

- (Setup) To set up the cryptosystem, we fix two distinct primes l_A and l_B[1], and two exponents e_A and e_B. Let p be a prime such that $p = f \cdot l_A^{e_A} \cdot l_B^{e_B} \pm 1$ for some very small f. We want to choose l_A and l_B such that $l_A^{e_A}$ and $l_B^{e_B}$ are about the same size, ideally $l_A^{e_A} \approx l_B^{e_B} \approx \sqrt{p}$. Now Alice and Bob agree on a supersingular curve E defined over $\mathbb{F}_{p^2}$. Note that

$$E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/l_A^{e_A}\mathbb{Z})^2 \times (\mathbb{Z}/l_B^{e_B}\mathbb{Z})^2 \times (\mathbb{Z}/f\mathbb{Z})^2.$$

  Now they fix public bases of their respective torsion groups:

$$\begin{aligned} E[l_A^{e_A}] &= \langle P_A, Q_A\rangle, \\ E[l_B^{e_B}] &= \langle P_B, Q_B\rangle. \end{aligned}$$

- (Key generation) To start the protocol, Alice chooses random secret integers m_A and n_A in $\mathbb{Z}/l_A^{e_A}\mathbb{Z}$ and construct the point $A := [m_A]P_A + [n_A]Q_A$, of order $l_A^{e_A}$. Now she consider the subgroup

$$\langle A\rangle = \langle [m_A]P_A + [n_A]Q_A\rangle \subseteq E[l_A^{e_A}]$$

  which has clearly order $l_A^{e_A}$. It generates the kernel of $\varphi_A : E \to E_A \cong E/\langle A\rangle$, with degree $l_A^{e_A}$, which she computes as a series of l_A-isogenies. Her public key is $(E_A, \varphi_A(P_B), \varphi_A(Q_B))$. Analogously Bob construct a private subgroup

$$\langle B\rangle = \langle [m_B]P_B + [n_B]Q_B\rangle \subseteq E[l_B^{e_B}]$$

  of order $l_B^{e_B}$ and computes the $l_B^{n_B}$-isogeny $\varphi_B : E \to E_B \cong E/\langle B\rangle$ as a series of l_B-isogenies; his public key is $(E_B, \varphi_B(P_A), \varphi_B(Q_A))$.

[1]These values will be very small, typically 2 or 3.

- (Key exchange) To compute the shared secret $E/\langle A, B\rangle$, Alice needs to compute the isogeny $\alpha' : E/\langle B\rangle \to E/\langle A, B\rangle$, whose kernel is generated by $\varphi_B(A)$. We see that the kernel of α' depends on both secrets, thus Alice cannot compute it without Bob's assistance. From Bob's public key Alice takes $\varphi_B(P_A)$ and $\varphi_B(Q_A)$. In this way she compute

$$\begin{aligned}[m_A]\varphi_B(P_A) + [n_A]\varphi_B(Q_A) &= \varphi_B([m_A](P_A)) + \varphi_B([n_A](Q_A)) \\ &= \varphi_B([m_A](P_A) + [n_A](Q_A)) \\ &= \varphi_B(A)\end{aligned}$$

  and computes the $l_A^{e_A}$-isogeny

$$\varphi'_A : E_B \to E_{BA} = E_B/\langle\varphi_B(A)\rangle \cong E/\langle A, B\rangle$$

  Bob performs the analogous computation, with the help of Alice, and computes the $l_B^{e_B}$-isogeny

$$\varphi'_B : E_A \to E_{AB} = E_A/\langle\varphi_A(B)\rangle \cong E/\langle A, B\rangle$$

  The shared secret is the j-invariant $j(E_{AB}) = j(E_{BA})$ in $\mathbb{F}_{p^2}$.

The protocol is sum up in Table 4.3.

Public parameters	Primes l_A, l_B, and a prime $p = l_A^{e_A} l_B^{e_B} f \pm 1$. A supersingular elliptic curve E over $\mathbb{F}_{p^2}$ of order $(p \pm 1)^2$. A basis $\langle P_A, Q_A\rangle$ of $E[l_A^{e_A}]$. A basis $\langle P_B, Q_B\rangle$ of $E[l_B^{e_B}]$.	
	Alice	**Bob**
Pick random secret	$A = [m_A]P_A + [n_A]Q_A$	$B = [m_B]P_B + [n_B]Q_B$
Compute secret isogeny	$\alpha : E \to E_A = E/\langle A\rangle$	$\beta : E \to E_B = E/\langle B\rangle$
Exchange data	$E_A, \alpha(P_B), \alpha(Q_B) \longrightarrow$	$\longleftarrow E_B, \beta(P_A), \beta(Q_A)$
Compute shared secret	$E/\langle A, B\rangle = E_B/\langle\beta(A)\rangle$	$E/\langle A, B\rangle = E_A/\langle\alpha(B)\rangle$

Table 4.3: Supersingular Isogeny Diffie-Hellman key exchange protocol.

4.2.2 Security

Extra Points. The protocol shows that in the key exchange we share much more information than we are used to with a normal Diffie-Hellman style key exchange, indeed Alice transmits not only the image curve $E/\langle A\rangle$, but also the image of the points P_B, Q_B, therefore the security of this protocol cannot be based on the isogeny path problem, and must be formalized ad hoc.

Definition 4.11 (SIDH isogeny problem)**.** Let (E, R_1, S_1, R_2, S_2) be a SIDH public key. Let E_A be such that there is an isogeny $\varphi_A : E \to E_A$ of degree $l_1^{e_1}$. Let $R'_2 = \varphi_A(R_2), S'_2 = \varphi_A(S_2)$. The problem is: Given $(E, R_1, S_1, R_2, S_2, E_A, R'_2, S'_2)$, determine an isogeny $\varphi_A : E \to E_A$ of degree $l_1^{e_1}$ such that $R'_2 = \varphi_A(R_2)$ and $S'_2 = \varphi_A(S_2)$.

Remark 4.12. Let $0 \le x, y < l_2^{e_2}$ and set $T = [x]R_2 + [y]S_2$. Then $\varphi_A(T) = [x]R'_2 + [y]S'_2$ can be computed. Hence an attacker can compute as many pairs $(T, \varphi_A(T))$ on the graph

of φ_A as he likes. A natural approach is to compute φ_A by solving an interpolation problem. However the difficulty is that φ_A has degree $l_1^{e_1}$ and so is described by rational functions of exponential degree. As of now, there are no algorithms that take advantage of this fact to force the protocol, however sharing this extra information could be a weakness in the future.

Public Key Validation. One of SIDH's biggest problems concerns its public key validation, i.e. the ability to verify that a public key was honestly generated. Suppose we have an algorithm that, given a prime l, a positive integer n, two elliptic curves E_0, E, efficiently decides whether the curve E is l^n-isogenous to E_0. Such an algorithm would allow us to verify whether Alice or Bob's public key was honestly generated (by calling the algorithm on (l_A, n_A, E_0, E_A) or (l_B, n_B, E_0, E_B), respectively). However, as we see in [92], this algorithm can also be used to efficiently recover secret keys from public keys. Indeed, take Alice's public curve E_A; there are $l_A + 1$ curves l_A-isogenous to it. Computing each of these isogenies $\varphi : E_A \to E'_A$, we call the algorithm on $(l^A, n_A - 1, E_0, E'_A)$; if it returns true, then φ is the last l_A-isogeny in Alice's secret key. Iterating this procedure reveals the entire key. This shortcoming leads to polynomial-time active attacks [36] for which countermeasures are expensive. For example, SIKE [51], which is the only isogeny-based candidate KEM in the NIST process, handles this by applying a transformation proposed by Hofheinz, Hövelmanns, and Kiltz [48] which is similar to the Fujisaki–Okamoto transform [34], essentially doubling the running time on the recipient's side compared to an ephemeral key exchange.

Lack of Symmetry. On a formal level, there are some profound differences between SIDH and classical Diffie-Hellman. The most obvious is the lack of symmetry in SIDH between Alice and Bob, whose roles are no longer interchangeable. This is reflected by their distinct and incompatible key spaces, which are in turn distinct from the shared secret space and the space the base curve lives in. Alice's private key encodes a sequence of l_A-isogenies of length n_A, while Bob's encodes a sequence of l_B-isogenies of length n_B. Alice's public key belongs to the space of (isomorphism classes of) elliptic curves equipped with a distinguished $l_B^{n_B}$-torsion basis, while Bob's is equipped with an $l_A^{n_A}$-torsion basis instead. The base curve E_0 is drawn from yet another space: it is equipped with an $l_A^{n_A} l_B^{n_B}$-torsion basis.

4.3 CSIDH

In this section we show that adapting the Couveignes-Rostovtsev-Stolbunov scheme to supersingular elliptic curves is possible, provided that one restricts to supersingular elliptic curves defined over a prime field $\mathbb{F}_p$. Instead of the full ring of endomorphisms, which is non-commutative, we consider the sub-ring of $\mathbb{F}_p$-rational endomorphisms, which is again an order $\mathcal{O}$ in an imaginary quadratic field. As before $\mathrm{Cl}(\mathcal{O})$ acts via isogenies on the set of $\mathbb{F}_p$-isomorphism classes of elliptic curves whose $\mathbb{F}_p$-rational endomorphism ring is isomorphic to $\mathcal{O}$ and whose trace of Frobenius has a prescribed value; in fact if $p \geq 5$ then there is only one option for this value, namely 0, in contrast with the ordinary case. Starting from these observations, the desired adaptation of the Couveignes-Rostovtsev-Stolbunov scheme almost unrolls itself. We call the resulting scheme CSIDH, where the C stands for *commutative*. While this fails to address Jao and De Feo's initial motivation for using supersingular elliptic curves, which was to avoid the $L_q[1/2]$ quantum attack due to Childs-Jao-Soukharev, we show that CSIDH eliminates the main problem of the

Couveignes-Rostovtsev-Stolbunov scheme, namely its inefficiency.

We begin by describing the protocol, and then go on to discuss the design choices. Then we describe how does the public-key validation work, and discuss the security of the protocol, both from a classical and quantum perspective. We discuss the attacks either from the theoretical point of view, i.e. as asymptotic estimates, and as real instances. Finally, we propose a proof-of-concept implementation, with related speed estimates.

4.3.1 The Protocol

We immediately present the protocol, after which we justify the constructive choice and explain how some operations are possible.

- (Setup) Global parameters of the scheme are a large prime $p = 4 \cdot l_1 \cdots l_n - 1$, where the l_i are small distinct odd primes, and the supersingular elliptic curve E_0 over $\mathbb{F}_p$ defined by $y^2 = x^3 + x$, with endomorphism ring $\mathcal{O} = \mathbb{Z}[\pi]$, where π denotes the Frobenius endomorphism.
- (Key generation) The private key is an n-tuple $(e_1, ..., e_n)$ of integers, each sampled randomly from a range $\{-m, ..., m\}$. These integers represent the private key, that is the ideal class $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}] \in \mathrm{Cl}(\mathcal{O})$, where $\mathfrak{l}_i = (l_i, \pi - 1)$. The public key is the coefficient $A \in \mathbb{F}_p$ of the elliptic curve $[\mathfrak{a}]E_0 : y^2 = x^3 + Ax^2 + x$ obtained by applying the action of $[\mathfrak{a}]$ to the curve E_0.
- (Key Exchange) Suppose Alice and Bob have key pairs $([\mathfrak{a}], A)$ and $([\mathfrak{b}], B)$. Upon receiving Bob's public key $B \in \mathbb{F}_p \setminus \{\pm 2\}$, Alice verifies that the elliptic curve $E_B : y^2 = x^3 + Bx^2 + x$ is indeed in $\mathrm{Ell}_p(\mathcal{O}, \pi)$. She then applies the action of her secret key $[\mathfrak{a}]$ to E_B to compute the curve $[\mathfrak{a}]E_B = [\mathfrak{a}][\mathfrak{b}]E_0$. Bob proceeds analogously with his own secret $[\mathfrak{b}]$ and Alice's public key A to compute the curve $[\mathfrak{b}]E_A = [\mathfrak{b}][\mathfrak{a}]E_0$. The shared secret is the Montgomery coefficient S of the common secret curve $[\mathfrak{a}][\mathfrak{b}]E_0 = [\mathfrak{b}][\mathfrak{a}]E_0$ written in the form $y^2 = x^3 + Sx^2 + x$, which is the same for Alice and Bob.

The protocol is summarized in Table 4.4.

Public parameters	A prime p of the form $4 \cdot l_1 \cdots l_n - 1$ $E_0 := y^2 = x^3 + x$ over $\mathbb{F}_p$	
	Alice	**Bob**
Pick random secret	$(e_1, ..., e_n) \in \{-m, ..., m\}^*$	$(e'_1, ..., e'_n) \in \{-m, ..., m\}^*$
Compute public data	$E_A = [\mathfrak{a}]E_0 = [\mathfrak{l}_1^{e_1} ... \mathfrak{l}_n^{e_n}]E_0$	$E_B = [\mathfrak{b}]E_0 = [\mathfrak{l}_1^{e'_1} ... \mathfrak{l}_n^{e'_n}]E_0$
Exchange data	$E_A \longrightarrow$	$\longleftarrow E_B$
Compute shared secret	$E_{AB} = [\mathfrak{a}]E_B$	$E_{AB} = [\mathfrak{b}]E_A$

Table 4.4: CSIDH key exchange protocol.

While describing the protocol we have overlooked some important aspects, for example, why do we take a prime p of the form $4 \cdot l_1 \cdots l_n - 1$? How do we prove that the endomorphism ring of E_0 is $\mathbb{Z}[\pi]$? Why do we have that, for any l_i as above, in the class group $l_i\mathcal{O} = \mathfrak{l}_i \cdot \overline{\mathfrak{l}_i} = (l_i, \pi - 1)(l_i, \pi + 1)$, or equivalently, why are the Frobenius eigenvalues ± 1? Why can the curve $[\mathfrak{a}][\mathfrak{b}]E_0$ still be expressed in Montgomery form? Let's give an answer to all these questions.

4.3.2 Design Choices

In this subsection, we discuss the parameters choice for the protocol defined above. We show how these choices make CSIDH a feasible protocol, differently from its predecessor.

Supersingular curves

One of the biggest limitations of the Couveignes-Rostovtsev-Stolbunov protocol is the inefficiency in the search for Elkies primes. We stress that these primes are of paramount importance for the protocol, as they guarantee us to decompose the ideal $l\mathcal{O}$ as the product of the two prime ideals $\mathfrak{l} = (l, \pi - \lambda)$ e $\bar{\mathfrak{l}} = (l, \pi - \mu)$, where π denotes the Frobenius endomorphism. De Feo-Kieffer-Smith tried to speed up this search by choosing a field of characteristic p, with p congruent to -1 modulo many small odd primes l. At this point they looked for an elliptic curve $E/\mathbb{F}_p$ such that $|E(\mathbb{F}_p)|$ is congruent to 0 modulo as many l as possible. This would guarantee the existence of points of order l over $E(\mathbb{F}_p)$. These properties ensure that $l\mathcal{O}$ decomposes as a product of $\mathfrak{l} = (l, \pi - 1)$ and $\bar{\mathfrak{l}} = (l, \pi + 1)$. For such primes l it is possible to compute the action of the corresponding classes $[\mathfrak{l}]$ and $[\bar{\mathfrak{l}}] = [\mathfrak{l}]^{-1}$ by applying Vélu's formulæ to the curve E. Hoping that this procedure is feasible for a fairly large number of such primes, we expect a generic element of the class group to be written as a product of powers of such small ideals $\mathfrak{l}$. In this way it would be possible to efficiently compute the action of a generic element of $\mathrm{Cl}(\mathcal{O})$. Despite the considerable effort leading to various improvements, the results obtained by De Feo, Kieffer and Smith are discouraging. With the best parameters found within 17000 hours of CPU time (for which we have only 7 small primes), evaluating one class-group action still requires several minutes of computation to complete. This suggests that without new ideas, the original Couveignes-Rostovtsev-Stolbunov scheme will not become anything close to practical in the foreseeable future.

This obstacle becomes trivial when using supersingular curves instead of ordinary curves, since for $p \geq 5$ any supersingular elliptic curve over $\mathbb{F}_p$ has exactly $p + 1$ rational points, i.e. $|E(\mathbb{F}_p)| = p + 1$, so that

$$\begin{aligned} |E(\mathbb{F}_p)| &\equiv p + 1 \pmod{l_i} \\ &\equiv 4 \cdot l_1 \cdots l_n - 1 + 1 \pmod{l_i} \\ &\equiv 0 \pmod{l_i} \end{aligned}$$

In other words $|\ E(\mathbb{F}_p)\ |$ is congruent to 0 modulo all primes l_i that we used in building p.

The use of supersingular elliptic curves over $\mathbb{F}_p$ has several other advantages: we have already observed that $\mathrm{Cl}(\mathcal{O})$ is a finite abelian group, whose cardinality is asymptotically $|\,\mathrm{Cl}(\mathcal{O})| \approx \sqrt{|D_\pi|} = \sqrt{|t_\pi^2 - 4p|}$. More precise heuristics actually predict that $|\,\mathrm{Cl}(\mathcal{O})|$ grows a little bit faster than $\sqrt{|D_\pi|}$, but the ratio is logarithmically bounded so we content ourselves with the above estimate. If the absolute value $|t|$ of the trace of Frobenius is "not too big", the discriminant D_π is about the size of p, hence by the above approximation we may assume $|\,\mathrm{Cl}(\mathcal{O})| \approx \sqrt{p}$. In the case of supersingular curves the trace of the Frobenius t is 0, therefore the absolute value of the discriminant $D_\pi = |t^2 - 4p| = 4p$ is as big as possible. As a direct consequence, the size of $\mathrm{Cl}(\mathcal{O})$ is close to its maximum possible value for a fixed choice of p. Conversely, this implies that for a fixed security level we can do an almost minimal choice for p, which directly affects the key size.

Remark 4.13. The choice of using supersingular curves therefore allows us to have advantages both for the research of the Elkies primes and the size of the class group. In

this regard we recall the CM construction from [11], which in principle could be used to construct ordinary elliptic curves with many points of small order, but the related endomorphism ring have very small class groups, excluding them for the Couveignes-Rostovtsev-Stolbunov key-exchange.

Parameters

As shown in the protocol in Table 4.4, we use a prime p of the form $4 \cdot l_1 \cdots l_n - 1$, where the l_i's are small odd distinct primes. It turns out that this choice is also beneficial for other reasons: in this way we have $p \equiv 3 \pmod 4$, and if we fix an elliptic curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$ (which has j-invariant $j = 1728$), thanks to Prop. 3.43, it turns out it is supersingular. The Frobenius endomorphism π satisfies the characteristic equation $\pi^2 - t\pi + p = 0$ for $t = 0$, that is, $\pi^2 = -p$, so its $\mathbb{F}_p$-rational endomorphism ring is an order[2] in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$.

Rational Elkies Primes

From the theory of complex multiplication, in particular from Theorem 3.44, the choices made above imply that the l_i-isogeny graph is a disjoint union of cycles. Moreover, since $\pi^2 - 1 \equiv 0 \pmod{l_i}$ the ideals $l_i\mathcal{O}$ split as $l_i\mathcal{O} = \mathfrak{l}_i \cdot \overline{\mathfrak{l}_i}$, where $\mathfrak{l}_i = (l_i, \pi - 1)$ and $\hat{\mathfrak{l}_i} = (l_i, \pi + 1)$. In other words, all the l_i are Elkies primes.

Sampling From the Class Group

Ideally, we would like to know the exact structure of the ideal-class group $\mathrm{Cl}(\mathcal{O})$ to be able to sample elements uniformly at random. This can be done in subexponential time using an algorithm of Hafner and McCurley [46], but unfortunately, this requires too much computation for the sizes of D_π we are working with, hence we resort to heuristic arguments. Assuming that the $\mathfrak{l}_i$ do not have very small order and are "evenly distributed" in the class group, we can expect ideals of the form $\mathfrak{l}_1^{e_1}\mathfrak{l}_2^{e_2} \cdot \mathfrak{l}_n^{e_n}$ for small e_i to lie in the same class only very occasionally. For efficiency reasons, it is desirable to sample the exponents e_i from a short range centered around zero, say $\{-m, ..., m\}$ for some integer m. We will argue later that choosing m such that $2m + 1 \geq \sqrt[n]{|\ \mathrm{Cl}(\mathcal{O})\ |}$ is sufficient. Since the prime ideals $\mathfrak{l}_i$ are fixed global parameters, the ideal $\prod_i \mathfrak{l}_i^{e_i}$ may simply be represented as a vector $(e_1, ..., e_n)$.

Remark 4.14. In practice, the proof-of-concept implementation proposed at the end of this discussion uses 74 small odd prime, to which we associate ideals $l_1, l_2, ..., l_{74}$. Heuristically, under the assumption that the product $\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$ evenly generate elements in $\mathrm{Cl}(\mathcal{O})$ (as the exponents change), we can compute the minimum interval in which we can choose the exponents themselves, in fact to be able to represent a class group of 256-bit size we just have to impose $\log(2 \cdot x + 1)^{74} = 256$ and observe that for $x = 5$ we obtain $\log(2 \cdot 5 + 1)^{74} \approx 255.9979$. We therefore expect that all elements of the class group can be expressed as $[l_1]^{e_1}[l_2]^{e_2} \cdot [l_{74}]^{e_{74}}$, where the exponents e_i are sampled from $\{-5, ..., 5\}$. The action of such an element can be computed as the composition of at most $5 \cdot 74 = 370$ easy isogenies evaluations. This should be compared to the use of 7 small primes (as was the case with De Feo-Kieffer-Smith), where the same approach would require exponents in an huge interval: the equation $\log(2 \cdot x + 1)^7 = 256$ can be solved for $x \approx 2^{256/7-1} \approx 2^{35}$ and the computation of the action of a generic element in $\mathrm{Cl}(\mathcal{O})$ could require $2^{35} \cdot 7$

[2] More precisely, we will show that $\mathrm{End}_{\mathbb{F}_p}(E_0) = \mathbb{Z}[\pi]$, which has conductor 2.

isogeny evaluations. Considering this, De Feo-Kieffer-Smith also had to resort to other prime numbers, with less beneficial properties, requiring them to work in extensions of $\mathbb{F}_p$, and therefore failing to make the protocol feasible.

Evaluating the Class Group Action

We now assume that any element of the class group can be represented as a product of small prime ideals, hence we describe how to compute $[\mathfrak{l}]E$ for a prime ideal $\mathfrak{l} = (l, \pi - \lambda)$. There are (at least) the following ways to proceed, which vary in efficiency depending on the circumstances:

- Find $\mathbb{F}_p$-rational roots of the modular polynomial $\varphi_l(j(E), Y)$ to determine the two j-invariants of possible co-domains (i.e., up to four non-isomorphic curves, though in the ordinary case wrong twists can easily be ruled out); compute the kernel polynomials [57] $\chi \in \mathbb{F}_p[x]$ for the corresponding isogenies (if they exist); if $(x_p, y_p) = [\lambda](x, y)$ modulo χ and the curve equation, then the co-domain was correct, else another choice is correct.
- Factor the l-th division polynomial $\psi_l(E)$ over $\mathbb{F}_p$; collect irreducible factors with the right Frobenius eigenvalues (as above); use Kohel's formulæ [57, Section 2.4] to compute the co-domain.
- Find a basis of the l-torsion - possibly over an extension field - and compute the eigenspaces of Frobenius; apply Vélu's formulæ to a basis point of the correct eigenspace to compute the co-domain.

As observed in [27, 54], the last method is the fastest if the necessary extension fields are small. The optimal case is $\lambda = 1$, which is exactly the case in which we fall. Note that if $p \equiv -1 \pmod{l}$, then $\lambda = 1$ automatically implies $\mu = -1$. In this case the kernel of $\varphi_{\mathfrak{l}_i}$ is the intersection of the kernels of the scalar multiplication $[l_i]$ and the endomorphism $\pi - 1$. That is, it is the subgroup generated by a point P of order l_i which lies in the kernel of $\pi - 1$ or, in other words, is defined over $\mathbb{F}_p$. Similarly, since

$$(\pi - 1)(\pi + 1) = (\pi^2 - 1)$$

and $\ker(\pi - 1) = \mathbb{F}_p$, $\ker(\pi^2 - 1) = \mathbb{F}_{p^2}$ directly implies $\ker(\pi + 1) = \mathbb{F}_{p^2} \setminus \mathbb{F}_p \cup \{O\}$, the kernel of $\varphi_{\overline{\mathfrak{l}_i}}$ is generated by a point Q of order l_i that is defined over $\mathbb{F}_{p^2}$ but not $\mathbb{F}_p$ and such that $\pi(Q) = -Q$. This greatly simplifies and accelerates the implementation, since both co-domains can easily be computed using Vélu's formulæ over an at most quadratic extension.

Computing the action of an ideal class represented by $\prod_i \mathfrak{l}_i^{e_i}$ on an elliptic curve E proceeds as outlined above. Since $\pi^2 = -p \equiv 1 \pmod{l_i}$, we are in the favourable situation that the eigenvalues of Frobenius on all l_i-torsion subgroups are $+1$ and -1 and we can efficiently compute the action of $\mathfrak{l}_i$. This step could simply be repeated for each ideal $\mathfrak{l}_i^{\pm 1}$ whose action is to be evaluated.

Remark 4.15. There are actually at least two other improvements that can be made, as we will show later. With the former, we will observe that a good choice of curve model allows for pure prime field computations. Alternatively, with the second, we will observe that for the crater on which we work, every curve present therein has its quadratic twist that is mirrored with respect to the axis passing through $y^2 = x^3 + x$. This allows us to work in $\mathbb{F}_p$ and then switch to the quadratic twist of the resulting curve.

4.3.3 Public-Key Validation

One of the biggest unsolved problems of SIDH is the lack of public-key validation, that is, the inability to understand if a public key was generated honestly. The risk we want to avoid is that an attacker exploits a weak public key to infer information about our private key. For CSIDH this problem does not arise, thanks to the following proposition.

Proposition 4.16. *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 $\mathrm{End}_{\mathbb{F}_p}(E) = \mathbb{Z}[\pi]$ if and only if there exists $A \in \mathbb{F}_p$ such that E is $\mathbb{F}_p$-isomorphic to the curve $E_A : y^2 = x^3 + Ax^2 + x$. Moreover, if such an A exists then it is unique.*

This result shows that all Montgomery curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ that are supersingular appear in the $\mathrm{Cl}(\mathcal{O})$-orbit of E_0. Moreover their $\mathbb{F}_p$-isomorphism class is uniquely determined by A., hence it may serve as a shared secret[3] without taking j-invariants. Therefore, by choosing public keys to consist of a Montgomery coefficient $A \in \mathbb{F}_p$, Proposition 4.16 guarantees that A represents a curve in the correct isogeny class $\mathrm{Ell}(\mathcal{O}, \pi)$, where $\pi^2 = -p$ and $\mathcal{O} = \mathbb{Z}[\pi]$, under the assumption that it is smooth (i.e. $A \in \mathbb{F}_p/\{\pm 2\}$) and supersingular. Since we work over $\mathbb{F}_p$ with $p \equiv 3 \pmod 8$ and start from the curve $E_0 : y^2 = x^3 + x$ with $\mathbb{F}_p$-rational endomorphism ring $\mathcal{O} = \mathbb{Z}[\pi]$. All we need to do upon receiving a candidate public key $y^2 = x^3 + Ax^2 + x$ is check for supersingularity, which is an easy task.

Verifying Supersingularity

As $p \geq 5$, it is a known fact that an elliptic curve E defined over $\mathbb{F}_p$ is supersingular if and only if $|E(\mathbb{F}_p)| = p + 1$. In general, proving that an elliptic curve has a given order N is easy if the factorization of N is known; exhibiting a subgroup (or in particular, a single point) whose order d is a divisor of N greater than $4\sqrt{p}$ implies the order must be correct. Indeed, the condition $d > 4\sqrt{p}$ implies that there exists only one multiple of d in the Hasse interval $[p + 1 - 2\sqrt{p}\ ;\ p + 1 + 2\sqrt{p}]$. This multiple must be the group order by Lagrange's theorem. Note that in our case a random point generally has very large order d: from the well known theory regarding the classification of finite abelian groups, $E(\mathbb{F}_p) \cong \mathbb{Z}_4 \times \prod_i \mathbb{Z}_{l_i}$, so that $l_i | d$ with probability $(l_i - 1)/l_i$, indeed an element of $E(\mathbb{F}_p)$ has order a multiple of l_i if and only if it is of the form $(x_4, x_{l_1}, ..., x_{l_n}) \in \mathbb{Z}_4 \times \prod_i \mathbb{Z}_{l_i}$ with $l_i \neq 0$. The probability that this happens is therefore

$$\mathbb{P}(l_i \mid d) = \frac{(l_i - 1) \cdot 4 \cdot \prod_{j \neq i} l_j}{4 \cdot \prod_j l_j} = \frac{(l_i - 1)}{l_i}.$$

Ignoring the even part, this shows that the expected order is lower bounded by

$$\prod_{i=1}^{n} \Big[l_i \cdot \mathbb{P}(l_i \mid d) + 1 \cdot \mathbb{P}(l_i \nmid d) \Big] = \prod_{i=1}^{n} (l_i - 1 + \frac{1}{l_i}) \approx p$$

This product is about the same size as p, and it is easily seen that a random point will with overwhelming probability have order (much) greater than $4\sqrt{p}$. This observation leads to a straightforward verification method, see Algorithm 1.

[3] The combination of large size of $\mathrm{Cl}(\mathcal{O})$ and representation by a single $\mathbb{F}_p$-element A explains the small key size of the scheme.

Algorithm 1 Verifying supersingularity

Input An elliptic curve $E/\mathbb{F}_p$, where $p = 4 \cdot l_1 \cdots l_n - 1$.
Output Supersingular or ordinary.

1: **function** VERIFYSUPERSINGULAR($E/\mathbb{F}_p$)
2: Randomly pick a point $P \in E(\mathbb{F}_p)$ and set $d \leftarrow 1$.
3: **for each** l_i **do**
4: Set $Q_i \leftarrow [(p+1)/l_i]P$.
5: **if** $[l_i]Q_i \neq \infty$ **then return** ordinary.
6: **if** $Q_i \neq \infty$ **then** Set $d \leftarrow l_i \cdot d$.
7: **if** $d > 4\sqrt{p}$ **then return** supersingular.

If the condition $d > 4\sqrt{p}$ does not hold at the end of Algorithm 1, the point P had too small order to prove $| E(\mathbb{F}_p) | = p+1$. In this case one may retry with a new random point P (although this outcome has negligible probability and could just be ignored). There is no possibility of wrongly classifying an ordinary curve as supersingular. Note moreover that if x-only Montgomery arithmetic is used and the point P is obtained by choosing a random x-coordinate in $\mathbb{F}_p$, there is no need to differentiate between points defined over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$; any x-coordinate in $\mathbb{F}_p$ works. Indeed, any point that has an x-coordinate in $\mathbb{F}_p$ but is only defined over $\mathbb{F}_{p^2}$ corresponds to an $\mathbb{F}_p$-rational point on the quadratic twist, which is supersingular if and only if the original curve is supersingular. There are more optimized variants of this algorithm; the bulk of the work are the scalar multiplications required to compute the points $Q_i = [(p+1)/l_i]P$. Since they are all multiples of P with shared factors, one may more efficiently compute all Q_i at the same time using a divide-and-conquer strategy, at the expense of higher memory usage.

4.3.4 Security

As in Couveignes-Rostovtsev-Stolbunov, the central problem of our new primitive is the following analogue to the classical discrete-logarithm problem.

Definition 4.17 (Key recovery problem)**.** Given two supersingular elliptic curves E_0, E defined over $\mathbb{F}_p$ with the same $\mathbb{F}_p$-rational endomorphism ring $\mathcal{O}$, find an ideal $\mathfrak{a}$ of $\mathcal{O}$ such that $[\mathfrak{a}]E_0 = E$. This ideal must be represented in such a way that the action of $[\mathfrak{a}]$ on a curve can be evaluated efficiently, for instance $\mathfrak{a}$ could be given as a product of ideals of small norm.

To be precise, this scheme relies on slightly different hardness-assumptions, as shown in definition 4.8. However, since there seems to be no way to attack the key exchange without recovering one of the keys, we will assume in the following analysis that the best approach to break the key-exchange protocol is to solve the key recovery problem.

Remark 4.18. We point out that the "inverse Diffie-Hellman problem" is easy in the context of CSIDH: given $[\mathfrak{a}]E_0$ we can compute $[\mathfrak{a}]^{-1}E_0$ by mere quadratic twisting: indeed if $p \equiv 3 \pmod 4$ it is possible to show that the connected component of E_0 is symmetric, meaning that if E is n step along $G_{\mathbb{F}_p,l}$ in one direction from E_0, then the curve that is n steps in the other direction is the quadratic twist of E[4].

[4]It is also interesting to observe that the symmetry around E_0 confirms the known fact that the class number of $\mathbb{Z}[\sqrt{-p}]$ is odd.

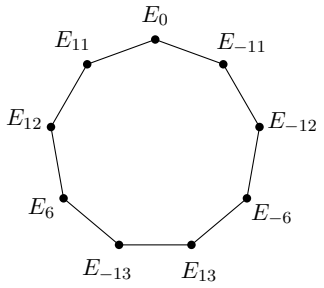


Figure 4.4: A supersingular component of $G_{\mathbb{F}_{83},3}$. For a fixed parameter a, we denoted with $E_a := y^2 = x^3 + ax^2 + x$. All curves have $\mathbb{F}_p$-rational endomorphism ring $\mathbb{Z}[\sqrt{-83}]$. Running clockwise corresponds to the repeated action of $[(3, \pi - 1)]$.

This clearly contrasts with the classical group-based setting. Note that just like identifying a point (x, y) with its inverse $(x, -y)$ in an ECDLP setting, this implies a security loss of one bit under some attacks: An attacker may consider the curves $[\mathfrak{a}]E$ and $[\mathfrak{a}]^{-1}E$ identical, which reduces the search space by half.

No Extra Information. One of the most worrying properties of SIDH seems to be that Alice and Bob publish the images of known points under their secret isogenies along with the co-domain curve, i.e., a public key is of the form $(E, \varphi(P), \varphi(Q))$ where $\varphi : E_0 \to E$ is a secret isogeny and $P, Q \in E_0$ are publicly known points. Although thus far nobody has succeeded in making use of this extra information to break the original scheme, Petit presented an attack using these points when overstretched, highly asymmetric parameters are used [73]. The Couveignes–Rostovtsev–Stolbunov scheme, and consequently our new scheme CSIDH, do not transmit such additional points: a public key consists of only an elliptic curve. Thus it is reasonable that a potential future attack against SIDH based on these torsion points would not apply to CSIDH.

Chosen-Ciphertext Attacks. This type of attack could be carried out if Eve performs a key-exchange with Alice using a weak key, and thanks to this he manages to infer some information about her private key. As explained above, the CSIDH group action features efficient public-key validation. This implies it can be used without applying a CCA transform such as the Fujisaki–Okamoto transform [34] used in SIDH, thus enabling efficient non-interactive key exchange and other applications in a post-quantum world.

Classical Security

We begin by considering classical attacks.

Exhaustive Key Search. The most immediate way to attack any cryptosystem is to use brute-force. We show how CSIDH performs under this attack. As we have already observed, a private key of our scheme consists of an exponent vector $(e_1, ..., e_n)$ where each e_i is in the range $\{-m, ..., m\}$, which represents the class $\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n} \in \mathrm{Cl}(\mathcal{O})$. This choice could lead to have the same class represented in more than one way, and this could weaken the security of the protocol. We show that this is indeed the case, yet the number of representations for the same class is small. The maximum number of these representations immediately yields the minimum entropy[5], which measures the amount of work an

[5]That is the logarithm in base 2 of the quantity under analysis

attacker has to do to successfully lead a key-retrieval.

In the following discussion we assume that $\mathrm{Cl}(\mathcal{O})$ is *almost cyclic*, in the sense that it has a very large cyclic component, say of order N not too small than $|\mathrm{Cl}(\mathcal{O})|$. According to Cohen and Lenstra's heuristics, this is true with high probability for a random imaginary quadratic field [18]. We then define the map

$$\rho : \mathrm{Cl}(\mathcal{O}) \to (\mathbb{Z}_N, +)$$

which projects an element of the class group onto the cyclic subgroup (isomorphic to) $\mathbb{Z}_N$. For each small prime ideal $\mathfrak{l}_i$ we define $\alpha_i := \rho(\mathfrak{l}_i)$. We assume $\alpha_1 = 1$; this can be done without loss of generality when at least one element l_i has order N in the class group. Observe that for each $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ any representation $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]$ of $[\mathfrak{a}]$ yields a solution of the linear congruence

$$e_1 \cdot 1 + e_2 \cdot \alpha_2 + \cdots + e_n \cdot \alpha_n \equiv \rho([\mathfrak{a}]) \pmod{N} \tag{4.1}$$

Therefore the number of solutions of this equation gives an upper bound to the number of representations of $[\mathfrak{a}]$. Observe that an element $(e_1, ..., e_n)$ is a solution of Equation 4.1 if and only if it is an element of a shifted version[6] of an integer lattice $\mathcal{L}$ generated by the rows of the matrix

$$L = \begin{pmatrix} N & 0 & 0 & \cdots & 0 \\ -\alpha_2 & 1 & 0 & \cdots & 0 \\ -\alpha_3 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\alpha_n & 0 & 0 & \cdots & 1 \end{pmatrix}$$

i.e. it is an element of the form $(Nz_1 - \alpha_2 z_2 - \cdots - \alpha_n z_n, z_2, \cdots, z_n) + v$, for some $z_i \in \mathbb{Z}$ and $v \in \mathbb{Z}^n$. This is easy to show, indeed requiring the existence of a vector $(e_1, ..., e_n)$ such that 4.1 holds is equivalent to ask that there exists an integer k for which $e_1 \cdot 1 + e_2 \cdot \alpha_2 + \cdots + e_n \cdot \alpha_n - \rho([\mathfrak{a}]) = Nk$. This is in turn equivalent to

$$\begin{cases} e_1 = Nk - e_2\alpha_2 - \cdots - e_n\alpha_n + \rho([\mathfrak{a}]) \\ e_2 = e_2 \\ \quad \vdots \\ e_n = e_n \end{cases}$$

Defining $z_1 = k, z_2 = e_2, ..., z_n = e_n, v = (\rho([\mathfrak{a}]), 0, \cdots, 0)$ we immediately get the result.

We now recall that, taking L a full-rank lattice in $\mathbb{R}^n$, and C a measurable subset of $\mathbb{R}^n$, the Gaussian Heuristic [72, Chapter 2, Definition 8] "predicts" that the number of points of $L \cap C$ is roughly $\mathrm{vol}(C)/\mathrm{vol}(L)$. So we expect $\mathrm{vol}([-m, m]^n)/\det(L) = (2m+1)^n/N$ solutions. Since we assumed $\mathrm{Cl}(\mathcal{O})$ to be cyclic of order (almost) N, we can approximate the above result with $(2m+1)^n/|\mathrm{Cl}(\mathcal{O})|$, which is not such a big value if we choose m as small as possible to get $(2m+1)^n \geq |\mathrm{Cl}(\mathcal{O})|$. Since $|\mathrm{Cl}(\mathcal{O})| \approx \sqrt{p}$, we expect there exist an ε such that the complexity of a brute force attack is around

$$2^{\log(|\mathrm{Cl}(\mathcal{O})|) - \varepsilon} = 2^{\log\sqrt{p} - \varepsilon}$$

[6] That is, a subset of $\mathbb{Z}^n$ given by $\mathcal{L} + v$, for some integer lattice $\mathcal{L}$ and some $v \in \mathbb{Z}^n$.

To verify this result, which in any case is based on non-trivial assumptions, Castryck, Lange, Martindale, Panny and Renes [15] took primes p, up to 40 bit, and tried to force the key both with brute force, or by randomly fishing represented by the class group, and with the sampling proposed in the protocol. The result is encouraging: The second method only loses a handful security-bits compared to uniform sampling. With the size of p tested, the min-entropy is at most 4 bits less than a perfectly uniform distribution on the class group, that is $\varepsilon \leq 4$. This could change for a larger choice of p, however there is no reason to think this, as long as we hold m and n so that $(2m+1)^n$ is not too much larger than $|\operatorname{Cl}(\mathcal{O})|$.

Meet-in-the-Middle Key Search. A meet-in-the-middle attack is a generic space–time trade-off cryptographic attack against encryption schemes that rely on performing multiple encryption operations in sequence. It is a known-plaintext attack. We focus on the well known baby-step giant-step and fit it to our protocol, showing that the resulting time complexity to lead an attack is $O(\sqrt[4]{p})$.

Remark 4.19 (Baby-step giant-step attack)**.** The baby-step giant-step is a meet-in-the-middle algorithm for computing the discrete logarithm or the order of an element in a finite abelian group due to Daniel Shanks [82]. The algorithm is based on a space–time trade-off. It is a fairly simple modification of trial multiplication, the naive method of finding discrete logarithms. Given a cyclic group G of order n, a generator α of the group and a group element β, the problem is to find an integer x such that $\alpha^x = \beta$. The baby-step giant-step algorithm is based on rewriting x as $x = im + j$, with $m = \lceil \sqrt{n} \rceil$, $0 \leq i < m$ and $0 \leq j < m$. Therefore, we have:

$$\begin{aligned} \alpha^x = \beta &\iff \alpha^{im+j} = \beta \\ &\iff \alpha^j = \beta \left(\alpha^{-m}\right)^i \end{aligned}$$

The algorithm pre-computes α^j for several values of j. Then it fixes α^{-m} and tries all different values of i in the right-hand side of the congruence above, in the manner of trial multiplication. It tests to see if the congruence is satisfied for any value of j, using the pre-computed values of α^j. The running time of the algorithm and the space complexity is $O(\sqrt{n})$, much better than the $O(n)$ running time of the naive brute force computation.

Since a private key $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]$ trivially decomposes as the product of two smooth ideals generated by two subsets of the starting ideal (for example, for some $k \in \{1, ..., n\}$, we can take $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_k^{e_k}] \cdot [\mathfrak{l}_{k+1}^{e_{k+1}} \cdots \mathfrak{l}_n^{e_n}]$) the classic trade-off provided by the baby-step giant-step applies. The resulting time complexity is $O(\sqrt{|\operatorname{Cl}(\mathcal{O})|}) \approx O(\sqrt[4]{p})$. It is possible to provide a graphical interpretation for this algorithm: this is in fact equivalent to find a path between the curve E and $[\mathfrak{a}]E$ in the corresponding isogeny graph, building a breadth-first tree from each one, and looking for a collision, as shown in Figure 4.5.

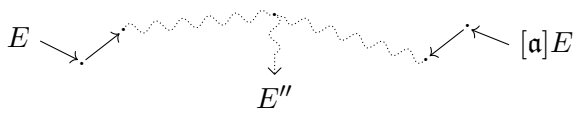


Figure 4.5: The meet-in-the-middle attack.

Pohlig-Hellman-Style Attack. The Pohlig–Hellman algorithm, sometimes credited as the Silver–Pohlig–Hellman algorithm, is a special-purpose algorithm for computing discrete logarithms in a finite abelian group whose order is a smooth integer [77].

Remark 4.20 (Pohlig-Hellman attack)**.** Given as input a finite cyclic abelian group G of order n with generator g, an element $h \in G$, and a prime factorization $n = \prod_{i=1}^{r} p_i^{e_i}$, the Pohlig-Hellman algorithm solves the discrete logarithm $g^x = h$ in the following way. Thanks to the classification of finite abelian groups, $G \cong \prod_i \mathbb{Z}_{p_i^{e_i}}$, so we can project g and g^x over each $p_i^{e_i}$-subgroup of G. The relative sub-problem is then solved, and finally the original solution is reconstructed with the Chinese remainder theorem. The worst-case input for the Pohlig–Hellman algorithm is a group of prime order: In that case, it degrades to the baby-step giant-step algorithm, hence the worst-case time complexity is $\mathcal{O}(\sqrt{n})$. However, it is much more efficient if the order is smooth: Specifically, if $\prod_i p_i^{e_i}$ is the prime factorization of n, then the complexity of the algorithm is

$$\mathcal{O}\left(\sum_i e_i(\log n + \sqrt{p_i})\right)$$

group operations. For this reason in classical algorithms, which could be weak to an attack of this type, it is always required that the group G contain at least one cyclic large order subgroup.

We observe that in our case the group we are acting on, namely $\mathrm{Ell}(\mathcal{O}, \pi)$, does not form a group whose operations are compatible with the action of the class group $\mathrm{Cl}(\mathcal{O})$, therefore there seems to be no way to apply a Pohlig-Hellman style algorithm and exploit the decomposition of finite abelian groups. In fact, the Pohlig-Hellman algorithm is based on the existence of an efficiently computable morphism towards some subgroups, which in our case would result in a morphism that projects a given curve on the orbit of E_0 under the action of a subgroup of $\mathrm{Cl}(\mathcal{O})$. Due to the impossibility of carrying out an attack of this type, the structure of the class group is not really relevant[7], in particular we do not require it to have a large-prime subgroup.

Quantum Security

We now present the state of quantum algorithms to solve the key recovery problem.

Remark 4.21 (The query model)**.** The query model is a computational model particularly suitable for analyzing quantum algorithms. Essentially we get a black-box function f and have to answer a question about it. Instead of measuring the time complexity of our algorithm, we measure the query complexity: the number of queries it makes to f. Why do we use the query model? Should not we only care about how much time an algorithm takes? It turns out that the query model has several advantages:

- It is simple to analyze.
- Often, the query complexity of an algorithm is the same as its time complexity. That is to say, often the function f is efficient to implement, and often the non-oracle parts of an algorithm are also efficient.
- All known interesting quantum algorithms fit in the query paradigm (for example, Shor's factorization algorithm is really a special use-case of a general period-finding query problem, but also Grover's algorithm and the hidden shift problem).

[7] Assuming it is big enough, anyway.

Grover's Algorithm and Claw Finding There is an easy exponential attack against our cryptosystem that improves upon exhaustive search [52], it lays on the claw finding problem and it leads to an attack on CSIDH with time complexity $O(\sqrt[6]{p})$.

Definition 4.22 (Claw finding problem)**.** Given two functions $f : A \to C$ and $g : B \to C$ with domain of equal size, the *claw problem* consist in finding a pair (a, b) such that $f(a) = g(b)$.

The claw problem can obviously be solved in $O(|A| + |B|)$ time and $O(|A|)$ space on a classical computer by building a hash table holding $f(a)$ for any $a \in A$ and looking for hits for $g(b)$ where $b \in B$. With a quantum computer, one can do better using the algorithm in [90], which has complexity $O(\sqrt[3]{|A||B|})$.

The attack has been presented for the first time against SIDH, and in this case it works as follows. To find an isogeny of degree $l_A^{e_A}$ between E and E_A, an attacker builds two trees of all curves isogenous to E (respectively, E_A) via isogenies of degree $l_A^{e_A/2}$. Once the trees are built, the attacker tries to find a curve lying in both trees. Since the degree of the isogeny φ_A is around $\sqrt{p}$ (much shorter than the size of the isogeny graph), it is unlikely that there will be more than one isogeny path (and thus more than one match) from E to E_A. This problem can be seen as an instance of the claw finding problem described above, therefore it can be solved with a classical computer in $O(l_A^{e_A/2}) = O(\sqrt[4]{p})$. With a quantum computer, the complexity comes down to $O(l_A^{e_A/3}) = O(\sqrt[6]{p})$ operations. Clearly the same kind of attack can easily be applied to CSIDH.

We observe that the classical subexponential attacks presented in the previous sections had already induced us to choose the parameter p *large enough* to withstand these threats. The size of p is quite large to protect us also against this latest attack. For example, suppose we want a scheme with 128-bit security for a quantum computer: an AES-128 key can be broken with 2^{64} quantum oracle queries, which would require us to instantiate CSIDH with $p > 2^{6 \cdot 64} = 2^{384}$. However we choose p much larger than this magnitude.

The Abelian-Hidden-Shift Problem

Definition 4.23 (Abelian hidden shift problem)**.** Let A be a finite abelian group, T a finite set and let $f_1, f_2 : A \to T$ be black-box functions. The functions f_1, f_2 are said to hide a shift $s \in A$ if f_1 is injective and $f_2(x) = f_1(xs)$ for all $x \in A$. The goal is then to recover s by evaluating the functions f_1 and f_2.

Childs-Jao-Shoukharev observed that construct an isogeny between E_0 and $E_A := [\mathfrak{a}]E_0$ can be easily formulated as an abelian hidden shift problem by defining the two functions $f_1, f_2 : \mathrm{Cl}(\mathcal{O}) \to \mathrm{Ell}(\mathcal{O})$ such that $f_1([\mathfrak{b}]) := [\mathfrak{b}]E_0$ and $f_2([\mathfrak{b}]) := [\mathfrak{b}]E_A$. Note that a solution $[\mathfrak{s}]$ to this problem implies that $E_A = [\mathfrak{1}]E_A = f_2([\mathfrak{1}]) = f_1([\mathfrak{1}\mathfrak{s}]) = [\mathfrak{s}]E_0$, that is $E_A = [\mathfrak{s}]E_0$. The hidden shift problem has suffered several attacks:

- Kuperberg [58] is the first that shows an algorithm capable of solving the hidden shift problem in a generic group H of order N with time, space and query complexity of $2^{O(\sqrt{\log N})}$. He also shows that each abelian hidden shift problem can be traced back to a dihedral hidden subgroup problem.
- Regev [79] shows a subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, in particular time and query complexity are upper

bounded by $2^{O(\sqrt{\log N \log\log N})}$. In this way we use less space, but we have worse time and query complexity.

- A follow-up algorithm by Kuperberg [59] uses $2^{O(\sqrt{\log N})}$ time, queries and classical space, but only $O(\log N)$ quantum space.

All these algorithms have subexponential time and space complexity. However, we observe that, beyond the number of estimated queries in the algorithms above, every single query requires evaluating the functions f_1, f_2 on arbitrary classes[8] of $\mathrm{Cl}(\mathcal{O})$, which is non-trivial. The time complexity of the algorithm must therefore take into account this important factor. Childs-Jao-Shoukharev show this can be done in subexponential time and space [17].

Remark 4.24. An important remark about all these quantum algorithms is that they do not immediately lead to estimates for runtime and memory requirements on concrete instantiations with $H = \mathrm{Cl}(\mathcal{O})$. Furthermore the algorithms by Kuperberg and Regev are shown to have subexponential complexity in the limit, and this asymptotic behavior is not enough to understand the space and time complexity on actual (small) instances. For example, Kuperberg's first paper [58, Theorem 3.1] mentions $O(2^{3\sqrt{\log N}})$ oracle queries to achieve a non-negligible success probability when N is a power of a small integer. He also presents a second algorithm that runs in $O(3^{\sqrt{2\cdot\log_3 N}}) = O(2^{1.8\sqrt{\log N}})$ [58, Theorem 5.1] (however the algorithm works with a generic group structure and does not compute a more accurate result). Of course, this does not contradict the time complexity of $2^{O(\log N)}$ as stated above, but for a concrete security analysis the hidden constants certainly matter a lot and ignoring the O typically underestimates the security.

Here we study in detail the complexity of quantum algorithms to solve the key recovery problem. We join the approaches presented above to solve the HSP, and the Childs-Jao-Shoukharev algorithm. We recall that by definition $L_a[1/2, b] := \exp\left[(b+1)\sqrt{\ln a \ln\ln a}\right]$.

- Childs-Jao-Shoukharev [17, Remark 4.8] show that Regev's algorithm, for solving D-HSP, has a query complexity of

$$L_N[1/2, \sqrt{2}] = \exp\left[(\sqrt{2}+1)\sqrt{\ln N \ln\ln N}\right] \tag{4.2}$$

  where $N = |\mathrm{Cl}(\mathcal{O})|$. They also present two algorithms to compute the isogeny oracle, the fastest of which has been implemented by Bisson [8], and has a time complexity of

$$L_p[1/2, 1/\sqrt{2}] = \exp\left[(1/\sqrt{2}+1)\sqrt{\ln p \ln\ln p}\right] \tag{4.3}$$

  Childs-Jao-Shoukharev compute the total cost to solve the key recovery problem as $L_p[1/2, 3/\sqrt{2}]$. Almost certainly, this result has been achieved multiplying (4.2) and (4.3) and imposing $N \approx p$. However, since $N \approx \sqrt{p}$, this turns out to be an overestimation, and the total complexity should be revisited as $L_p[1/2, 1+\sqrt{2}]$.

  We observe that, even if the space complexity of Regev's algorithm is polynomial, Galbraith and Vercauteren [37] show that this algorithm (Regev + Bisson) has superpolinomial space complexity, and this is due to the high memory usage in the computation of the isogeny oracle. We refer to [53] for further details.

[8] That is, without being given a representative that is a product of ideals of small prime norm.

- Childs-Jao-Shoukharev also compute the total complexity that would occur with joining Kuperberg and Bisson's methods; the result is $L_p[1/2, 1/\sqrt{2}]$. This method requires superpolinomial storage (also before considering the space required by the oracle). We observe that in this case the computational cost of the isogeny oracle dominates asymptotically.

As we have noted above, it is important to mention that asymptotically worse algorithms may provide practical improvements on our *small* instances over either of the algorithms studied by Childs–Jao–Soukharev: for example, Couveignes [23, Chapter 5] provides heuristic arguments that one can find smooth[9] representatives of ideal classes by computing the class-group structure (which can be done in polynomial time on a quantum computer [45]) and applying a lattice-basis-reduction algorithm such as LLL [63] to its lattice of relations. This might be more efficient than using Childs–Jao–Soukharev's subexponential oracle. However, note that this method makes evaluating the oracle several times harder for the attacker than for legitimate users, thus immediately giving a few additional bits of security, since users only evaluate the action of very smooth ideals by construction. Further research in this direction is necessary and important, since it will directly impact the cost of an attack, but we consider a detailed analysis of all these algorithms and possible trade-offs to be beyond the scope of this work.

Recent Attacks There have been several independent attempts to violate CSIDH after its first publication. We report a few of them.

- A recent article by Biasse, Iezzi and Jacobson [7] describes how to optimize the computation of the action of an element of the class group, and how to represent the elements of this group as a product of small prime ideals. In particular, they describe two algorithms to compute an isogeny between two elliptic curves E_1, E_2, defined on the same finite field and with the same endomorphism ring $\mathcal{O}$. With the approximation that $|\Delta| \approx p$, the first algorithm has a running time of $2^{O(\sqrt{p})}$, it needs polynomial quantum memory and $2^{O(\sqrt{p})}$ quantumly accessible classical memory. The second algorithm, which is based on a variant of Regev's Algorithm, has execution time $L_p[1/2, 1/\sqrt{2}]$, and requires polynomial space (classical and quantum). The analysis proposed in this paper is uniquely asymptotic, and a more precise study regarding its complexity is explicitly left to future work.

- Bonnetain and Schrottenloher [9] take up the Kuperberg algorithm (the faster of the two). The basic idea is to ignore the space complexity of the algorithm, which would make it impracticable, and take into account only its time complexity. The estimated number of oracle queries is $(5\pi^2/4)2^{1.8\sqrt{\log N}}$, where the factor 1.8 appears as an approximation of $\sqrt{2\log 3}$ in Kuperberg. Bonnetain-Schrottenloher estimate the number of qubits needed to implement this algorithm, i.e. the space complexity, such as $2^{1.8\sqrt{\log N}+2.3}$. For small instances of N, as for example CSIDH-512, the number of qubits just described might be physically achievable, however in these cases the overall complexity of the algorithm is dominated by the high computational cost of the oracle, which Childs–Jao–Soukharev placed at $L_p[1/2, 1/\sqrt{2}]$. To avoid this, Bonnetain-Schrottenloher do not use the method proposed by Childs–Jao–Soukharev, but take use the LLL-based method described by Couveignes, applying BKZ for a more efficient lattice-basis reduction. The Bonnetain-Schrottenloher attack is correlated with complexity estimates for CSIDH instantiation parameters, for which the

[9] In any case, not as smooth as the ones used by the participants of the protocol.

CSIDH-512 instance would not meet the NIST-1 security specifications. Unfortunately these estimates ignore a big part of the computational effort and therefore, although this attack is a significant improvement over the previous ones, this does not affect CSIDH's security claim when accounting precisely for the actual cost of oracle queries. In particular

- Algorithm 2 makes heavy use of input-dependent branches, which is impossible in superposition.
- The algorithm skips finding points of order l_i, which are needed as the kernel of the l_i-isogeny.
- The computational estimate applies a result for multiplication costs in $\mathbb{F}_{2^n}$ to multiplications in $\mathbb{F}_p$.

Therefore Bonnetain-Schrottenloher estimates do not imply that CSIDH-512 is broken under NIST-level 1. More analysis is certainly needed and it is unclear whether that will result in larger or smaller choices of p. In the estimates that we present in the next section we keep this attack in mind, with the theoretical complexity described by the original article.

Instantiations

Here are some estimates for CSIDH instances with a fixed size of p.

Security Estimates The basic idea is to instantiate the attacks defined above. We consider the best classical attack, which we have shown to have complexity $O(\sqrt[4]{p})$. For quantum attacks, we consider Regev and Kuperberg's approaches (which have complexity of $L_N[1/2, \sqrt{2}]$, $O(2^{3\sqrt{\log N}})$ and $O(2^{1.8\sqrt{\log N}})$, joined with Childs-Jao-Shoukharev (for an overall complexity of $L_p[1/2, 3/\sqrt{2}]$ and $L_p[1/2, 1/\sqrt{2}]$).

CSIDH-log p	Classical $\log \sqrt[4]{p}$	Regev [79] $\log L_N[1/2, \sqrt{2}]$	Kuperberg [58] $3\sqrt{\log N}$	Kuperberg [59] $1.8\sqrt{\log N}$	Table 7 in [9] $1.8\sqrt{\log N}$	[17]-Regev $\log L_p[1/2, 3/\sqrt{2}]$	[17]-Kuperberg $\log L_p[1/2, 1/\sqrt{2}]$	Table 8 in [9] $1.8\sqrt{\log N}$
CSIDH-512	128	62	48	29	32.5	139	47	71
CSIDH-1024	256	94	68	41	44.5	209	70	88
CSIDH-1792	448	129	90	54	57.5	288	96	104

Table 4.5: Estimated attack complexities ignoring limits on depth. The three rightmost columns state costs for the complete attack; the others state classical and quandum query complexities. All numbers are rounded to whole bits and use $N = |\operatorname{Cl}(\mathcal{O})| = \sqrt{p}$, $o(1) = 0$ and all hidden O-constant 1.

The results we present do not take into account any memory costs, and we do not bother to limit the maximum depth of quantum circuits. We stress that these results should be subjected to more in-depth analysis, which take into account the implicit constants, the non-feasibility of performing a long sequence of quantum operations, and the immense memory requests. Also, we emphasize that a recent analysis [1] shows that a classic attack

on SIDH (it is the same attack on CSIDH) is much slower in practice than in theory. This is due to the immense amount of memory used in the computation. Similarly, the cost of a quantum attack is higher than the theoretical one obtained as the product of query complexity times the cost of the group action because evaluating the oracle in superposition is significantly more expensive that a regular group action.

Remark 4.25. Observe that a public key of CSIDH consists of an element $A \in \mathbb{F}_p$, and therefore can be represented with $\lceil \log p \rceil$-bit. It is also possible to estimate the size of a private key $(e_1, ..., e_n)$. Indeed, as we have seen before, $n \log(2m+1) \approx \log \sqrt{p}$, so that

$$n \cdot \log m \approx n \cdot \log(2m+1) \approx \log \sqrt{p} = (\log p)/2$$

from which it immediately follows that the size of the private key is, with good approximation, $(\log p)/2$. Consequently, with reference to Table 4.5, the respective public keys have a size of $64, 128$ and 256 bytes, while private keys are approximately half the size of the public keys listed above.

Security Levels We briefly recall the security specifications for post quantum algorithms, as they are defined by NIST, after which we verify the security of CSIDH.
Recall that a k-bit security level means that the best attack is at least as difficult as performing a key-retrieval attack on a block cipher with a k-bit key, for example AES-k for $k \in \{128, 192, 256\}$. This request is equivalent to ask that, under the assumption that the attacks query an oracle on a circuit at least as costly as AES[10], we should have a query complexity of at least 2^{k-1} to a classical oracle, and $2^{\sqrt{k}}$ to a quantum oracle. The following table summarizes the security levels of a given protocol, as established by NIST.

Level	Security description
1	At least as hard to break as AES-128 (exhaustive key search)
2	At least as hard to break as SHA-256 (collision search)
3	At least as hard to break as AES-192 (exhaustive key search)
4	At least as hard to break as SHA-384 (collision search)
5	At least as hard to break as AES-256 (exhaustive key search)

Table 4.6: Security strength categories, as defined by NIST, which asked submitters to focus on levels $1, 2$, and 3 (levels 4 and 5 are for high security).

The parameters p of CSIDH-$\log p$ presented in Table 4.5 are chosen to match the query-complexity of Regev's attack on the hidden shift problem (check the third column of the table) for more or less $2^{k/2}$, in order to match the levels $1, 3, 5$ proposed by NIST, under the assumption that the computation of the group action has depth at least as large as that of AES.

Observe that adjusting the cryptosystem parameters consists essentially to change p, which is very simple. Therefore, in the case that the safety estimates above were not sufficient, we can quickly readjust the protocol so that it regains its security.

4.3.5 Implementation

Let us now discuss the low-level implementation.

[10] In our setting: the group action computation has depth at least as large as AES.

Montgomery Curves

Proposition 4.16, together with the request $p+1 \equiv 4 \pmod 8$ implies that all curves in $\mathrm{Ell}(\mathbb{Z}[\pi], \pi)$ are $\mathbb{F}_p$-isomorphic to a curve in Montgomery form, i.e. $y^2 = x^3 + Ax^2 + x$, for some $A \in \mathbb{F}_p$. We have already observed how this parameterization offers particularly efficient arithmetic on the x-line. These advantages also extend to the computation of isogenies, as shown in [21]. The implementation we describe uses curves in Montgomery form, and for isogeny computation uses a projectivized variant of the formulæ described in [19] and [80]. We work with projective spaces to avoid almost all inversions.

Suppose we want to compute an isogeny of degree $l \geq 3$. We first find a point P of order l. Then, defining $(X_k : Y_k : Z_k) := [k]P$, for each $k \in \{1, ..., l-1\}$, for [80] we have that the Montgomery coefficient of the image curve of E, under the action of an isogeny with kernel $\langle P \rangle$ is given by $\tau(A - 3\sigma)$, where

$$\tau = \prod_{i=1}^{l-1} \frac{X_i}{Z_i}, \quad \sigma = \sum_{i=1}^{l-1} \left(\frac{X_i}{Z_i} - \frac{Z_i}{X_i}\right)$$

it is possible to compute this result more quickly, indeed by defining $c_i \in \mathbb{F}_p$ in such a way that

$$\prod_{i=1}^{l-1} (Z_i w + X_i) = \sum_{i=0}^{l-1} c_i w^i \tag{4.4}$$

we observe that

$$\begin{aligned}
\tau(A - 3\sigma) &= \prod_{i=1}^{l-1} \frac{X_i}{Z_i} \left(A - 3 \sum_{i=1}^{l-1} \left(\frac{X_i}{Z_i} - \frac{Z_i}{X_i}\right) \right) \\
&= A \cdot \prod_{i=1}^{l-1} \frac{X_i}{Z_i} - 3 \cdot \prod_{i=1}^{l-1} \frac{X_i}{Z_i} \left(\sum_{i=1}^{l-1} \left(\frac{X_i}{Z_i} - \frac{Z_i}{X_i}\right) \right) \\
&= \left(\prod_{i=1}^{l-1} Z_i \right)^{-2} \left[A \left(\prod_{i=1}^{l-1} Z_i \right) \left(\prod_{i=1}^{l-1} X_i \right) - 3 \left(\prod_{i=1}^{l-1} Z_i \right) \left(\prod_{i=1}^{l-1} X_i \right) \left(\sum_{i=1}^{l-1} \left(\frac{X_i}{Z_i} - \frac{Z_i}{X_i}\right) \right) \right] \\
&= (\star)
\end{aligned}$$

We observe at this point that, by the contruction of (4.4), $\prod_i Z_i = c_{l-1}$. On the other hand $\prod_i X_i = c_0$. Furthermore $\sum_i \left(\frac{X_i}{Z_i} - \frac{Z_i}{X_i}\right) = \sum_i \frac{X_i}{Z_i} - \sum_i \frac{Z_i}{X_i}$, and $\sum_i \left(\frac{Z_i}{X_i}\right) = \frac{c_1}{\prod_i X_i} = \frac{c_1}{c_0}$; similarly $\sum_i \left(\frac{X_i}{Z_i}\right) = \frac{c_{l-2}}{c_{l-1}}$ and therefore

$$\begin{aligned}
(\star) &= c_{l-1}^{-2} \cdot \left[A c_0 c_{l-1} - 3 \cdot c_0 c_{l-1} \left(\frac{c_{l-2}}{c_{l-1}} - \frac{c_1}{c_0} \right) \right] \\
&= c_{l-1}^{-2} \cdot \left[A \cdot c_0 c_{l-1} - 3 \cdot \left(c_0 c_{l-2} - c_1 c_{l-1} \right) \right]
\end{aligned}$$

Beyond that, since $x([k])P = x([l-k]P)$ for each $k \in \{1, ..., (l-1)/2\}$, we can reduce the computation necessary to evaluate the points $(X_k : Z_k)$ by about half. With these premises it is possible to show that the computational effort required to compute $\tau(A - 3\sigma)$ is $5l\mathbf{M} + l\mathbf{S}$ operations[11].

[11] $\mathbf{M}$ e $\mathbf{S}$ denote respectively a multiplication and a squaring in $\mathbb{F}_p$.

Evaluating the Class Group Action

We show how evaluating the class group action can be made more efficient. Recall that the purpose is to evaluate the action of an element of the class group of the form $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]$ on a curve $E_A \in \mathrm{Ell}(\mathbb{Z}[\pi], \pi)$ of the form $y^2 = x^3 + Ax^2 + x$, where each $\mathfrak{l}_i = (l_i, \pi - 1)$ is a prime ideal of small odd norm l_i, and all the e_i are integer sampled in the range $\{-m, ..., m\}$. We propose two approaches.

- The most natural way to do this is to consider every single factor $\mathfrak{l}_i^{\pm 1}$, find the abscissa of a point P of order l_i on E, which will be (depending on the sign of the exponent of $\mathfrak{l}_i$) defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$. The existence of such a point is guaranteed to us by our constructive choices of p and l_i. To find such a point it is sufficient to randomly sample a coordinate in $\mathbb{F}_p$, check if $x^3 + Ax^2 + x$ is a root in $\mathbb{F}_p$ (in case we are working with $\mathfrak{l}_i^{+1}$, then we require that $x^3 + Ax^2 + x$ is a root in $\mathbb{F}_p$, otherwise we will require the opposite, which is a root in $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$). At this point we perform a scalar multiplication of the result found by $(p+1)/l_i$, and check that the result is not the neutral element O. In this case we repeat the whole operation.

  We observe an important fact: if $P \in E(\mathbb{F}_p)$ then necessarily $[(p+1)/l]P \in E(\mathbb{F}_p)$, but if $P \in E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p)$, are we sure that $[(p+1)/l]P \in E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p)$? This result is *true*, and follows from the fact that $E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p) = \ker(\pi+1) \setminus \{O\}$. Since $\ker(\pi + 1)$ is a subgroup, we obtain that $[(p+1)/k]P \in \ker(\pi + 1)$. The only case in which this point *falls* to the ground group $E(\mathbb{F}_p)$ is because it has been mapped to the neutral element, but this case is automatically discarded by the algorithm that we have just described. At this point the isogeny kernel can be computed using Vélu's formulæ. Repeating the result for all $\mathfrak{l}_i^{\pm 1}$ gives us the desired result.

  We observe that this method has some notable disadvantages: fixing the sign of the exponent before sampling points implies limiting ourselves to consider about half of the points (in addition, to realize that a point is not suitable, we also have to do a square-test). Also, decide which prime l_i to use before sampling leads to exclude other points (again, to realize that a point has order different from l_i we should do a square-test and a scalar multiplication for $(p+1)/l_i$). Let us see how the next algorithm allows to mitigate these problems.

- The second method we present does not fix a prime l_i a priori. On the contrary, we take a generic coordinate $x \in \mathbb{F}_p$, we determine if $x^3 + Ax^2 + x$ is an element of $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$, and use this element to compute as many isogenies as possible.

  In detail, as shown in Algorithm 2, after randomly choosing an element $x \in \mathbb{F}_p$, we denote with s the sign of the exponent. We stress that in the case $s = 1$ then the point is defined over $\mathbb{F}_p$, vice-versa it is defined over $\mathbb{F}_{p^2}$. Let us now take into account all the indices e_i that have a sign compatible with that of s and enclose all these values into a new variable S. At this point we enclose in k the product $\prod_{i \in S} l_i$ of all the orders that are fine for our choice of x . Therefore we set $Q = [(p+1)/k]P$. As shown above, asking that P lies in $E(\mathbb{F}_p)$ is equivalent to ask that Q lies in $E(\mathbb{F}_p)$, and asking that P lies in $E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p)$ is equivalent to ask that Q lies in $\big(E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p)\big) \cup \{O\}$. At this point we try to see if there is any order l_i that fit our point: we compute $R = [k/l_i]Q$ and we verify that it is not the neutral element. In this case $\langle R \rangle$ is the kernel of an isogeny of degree l_i, and is already *oriented* with respect to the sign of the exponent of the ideal $\mathfrak{l}_i$ (in the sense that if the exponent has a negative sign, then surely R is a point of $E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p)$, and vice versa if the

exponent has a positive sign, then surely R is an element of $E(\mathbb{F}_{p^2})$). At this point we compute the image curve, and also the image of Q, trying to reuse it as long as we can.

Algorithm 2 Evaluating the class group action

Input $A \in \mathbb{F}_p$ and a list of integers
Output B such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_A = E_B$, where $E_B : y^2 = x^3 + Bx^2 + x$.

function evalGroupAction$(A, e_1, ..., e_n)$
  while some $e_i \neq 0$ **do**
    sample a random $x \in F_p$.
    if $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$ **then** $s \leftarrow 1$
    else $s \leftarrow -1$
    Let $S := \{i | e_i \neq 0, \text{sign}(e_i) = s\}$.
    if $S = \emptyset$ **then** start over with a new x
    else
      Let $k \leftarrow \prod_{i \in S} l_i$ and compute $Q \leftarrow [(p+1)/k]P$.
      for each $i \in S$ **do**
        Compute $R \leftarrow [k/l_i]Q$
        if $R = O$ **then** skip this i.
        Compute the map $\varphi : E_A \to E_B : y^2 = x^3 + Bx^2 + x$ with $\ker(\varphi) = \langle R \rangle$.
        Set $A \leftarrow B$, $Q \leftarrow \varphi(Q), k \leftarrow k/l_i$ and finally $e_i \leftarrow e_i - s$.

Thanks to the commutativity of $\text{Cl}(\mathcal{O})$, and since the absolute value of each e_i decreases from time to time, the algorithm ends with the computation of $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E$.

We observe that, since the probability that a random point has order divisible by l_i (which therefore leads us to have an isogeny in the above algorithm) grows with l_i, the isogeny steps for large l_i are typically computed before those associated with small l_i. Among the various solutions that have been proposed to this problem, one suggests not to include any small l_i in the factorization of $p + 1$, in order to reduce the expected number of unnecessarily repeated cycles.

Remark 4.26. Algorithm 2, when implemented naively, is strongly variable-time. This is due to the fact that the number of cycles that the algorithm performs is directly linked to the degree of the isogeny we are trying to compute. It would not be difficult to create a constant time implementation: it would be enough to use the same algorithm, but always execute all the commands of each cycle, storing only useful results. This would lead to an algorithm that performs more operations than necessary and we leave the study of an optimized constant-time algorithm for future work.

Key Validation

In a previous section we introduced an algorithm for a public key validation, which computes $[(p+1)/l_i]P$ for each $i \in \{1, ..., n\}$, where P is a random point of E. It is possible to improve the efficiency of this algorithm by introducing a recursive variant, described below with Algorithm 3.

Algorithm 3 Batch cofactor multiplication [89, Algorithm 7.3]

Input An Elliptic curve point P and positive integers $(k_1, ..., k_n)$.
Output The points $(Q_1, ..., Q_n)$, where $Q_i = [\prod_{j \neq i} Q_j]P$.

1: **function** CofactorMultiplication$(P, k_1, ..., k_n)$
2: **if** $n = 1$ **then return** P.
3: Set $m \leftarrow \lceil \frac{n}{2} \rceil$ and let $u \leftarrow \prod_{i=1}^{m} k_i, v \leftarrow \prod_{i=m+1}^{n} k_i$.
4: Compute $L \leftarrow [v]P$ and $R \leftarrow [u]P$.
5: **Recurse** with input $L, (k_1, ..., k_m)$ giving $(Q_1, ..., Q_m)$.
6: **Recurse** with input $R, (k_{m+1}, ..., k_n)$ giving $(Q_{m+1}, ..., Q_n)$. **return** $(Q_1, ..., Q_n)$.

We observe that, similarly to its precursor, this algorithm only works on public parameters, and therefore need not to be constant-time in a side channel resistant implementation. We show how this algorithm is useful to us, that is, how it can be used to verify whether a curve is supersingular or not. The basic idea is that we still use Algorithm 2, but we exploit the pre-computed points with this new algorithm just described.

We first choose a random point $P \in E(\mathbb{F}_p)$ and run Algorithm 3 with input $[4]P$ and $(l_1, ..., l_n)$ in order to obtain the sequence $(Q_1, ..., Q_n)$ with $Q_i = [(p+1)l_i]P$. At this point can we run Algorithm 2 to verify that E is supersingular.

We observe that it is not necessary to run Algorithm 3, wait for it to return all points $Q_1, ..., Q_n$, and then proceed to the various checks, in fact the order check of Algorithm 2 can be performed as soon as a new point Q_i becomes available, i.e. in the base case of Algorithm 3. This reduces the overall memory usage, since each point Q_i is discarded immediately after use, it also decreases the total execution time, since it ends when it has collected enough information.

Remark 4.27. Compared to Algorithm 2, the combined action of these two algorithms essentially comes from a space-time trade-off. It is natural to think that the second algorithm is the most suitable to implement, and this is generally true, however on severely memory-constrained devices one may instead opt for the naive algorithm, which requires less space but is slower.

Performance

We now present a proof-of-concept implementation for CSIDH for a 512-bit prime p. The implementation we refer to is the one provided by the authors of the protocol themselves[12].

Remark 4.28. Although the implementation we refer to uses a 512-bit field arithmetic written in assembly, more specifically for a Skylake processor, the program also contains some generic C code, which supports other field-sizes. In this way the program can also be run on different computer architectures or with different parameter-sizes, if we wish so.

The implementation we present uses $p = 4 \cdot l_1 \cdots l_{74} - 1$, where the primes $l_1, ..., l_{73}$ are the 73 smallest odd primes, and l_{74} is the smallest odd prime such that $4 \cdot l_1 \cdots l_{74} - 1$ is prime. We can verify that $l_{74} = 587$. The parameters choice immediately implies (as we had already observed) that the size of the public key is 64 bytes. Private keys are stored in 37 bytes for simplicity, however an efficient version may come to store their

[12] All code is published in the public domain and is available for download at https://yx7.cc/code/csidh/csidh-latest.tar.xz.

information in 32 bytes. Table 4.7 summarizes the proof-of-concept implementation we have just described.

	Clock cycles	**Wall-clock time**	**Stack Memory**
Key validation	$5.5 \cdot 10^6$ cc	2.1 ms	4368 bytes
Group action	$106 \cdot 10^6$ cc	40.8 ms	2464 bytes

Table 4.7: Performance number for the described proof-of-concept implementation, averaged over 10000 runs on an Intel Skylake i5 processor clocked at 3.5 GHz.

We have not taken into account the execution times for generating the private key as this only consists in sampling n random integers in the range $\{-m, ..., m\}$, which has a negligible cost. However, the original authors emphasize that this implementation is intended only as a proof-of-concept, in particular it is not side-channel resistant and may contain some bugs. The design of hardened and more optimized implementations are left for future work. This proof-of-concept implementation carries out a non-interactive key exchange at a presumed classical security level of 128 bits and a conjectured post-quantum security level of 64 bits in about 80 milliseconds, while using key sizes of only 64 bytes. This is over 2000 times faster than the current state-of-the-art instantiation of the Couveignes-Rostovtsev-Stolbunov scheme by De Feo, Kieffer and Smith [27, 54], which itself presents many new ideas and speedups to even achieve that speed. For comparison, we remark that SIDH, which is the NIST submission with the smallest combined key and ciphertext length, uses public keys and ciphertexts of over 300 bytes each. More precisely SIKE's version p503 uses uncompressed keys of 378 bytes long [51] for achieving CCA security. The optimized SIKE implementation is about ten times faster than this proof-of-concept C implementation, but even at 80 ms, CSIDH is practical. Consider in this regard Figure 4.6, where we denoted the execution time of the major Nist PQC candidates, dividing their key-generation, encryption and decryption execution time.

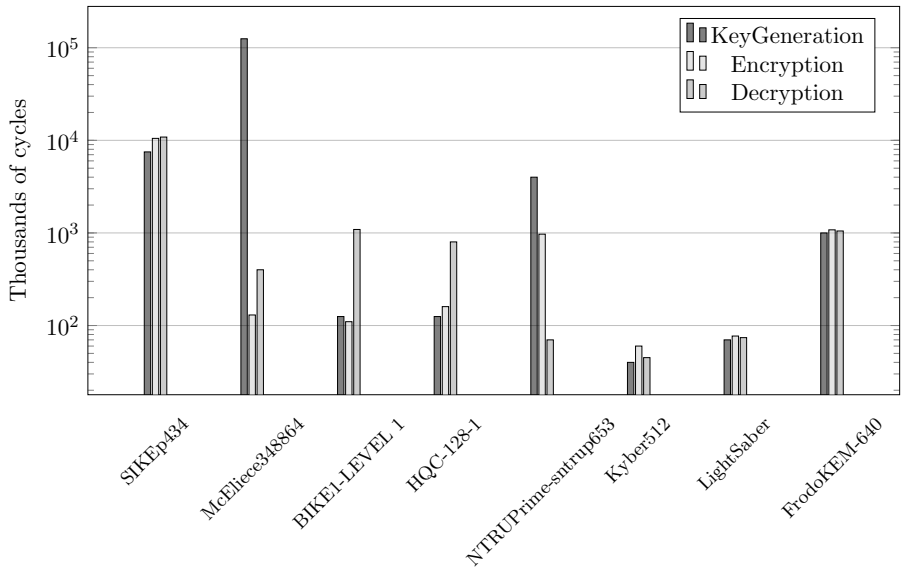


Figure 4.6: Speed - PKE/KEMs for NIST level 1

Remark 4.29. The first round SIKE submission offered three different security levels known as SIKEp503, SIKEp751, andSIKEp964. According to the best known quantum

attacks on solving supersingular isogeny problem by that time, the proposed security levels met NIST's level 1, 3, and 5 requirements, respectively. However, recent studies on the cost of solving isogeny problem on quantum computers revealed that the security assumptions for SIKE was too conservative. Accordingly, the second round SIKE offers a new set of security levels which are more realistic and provide significant improvement on the key encapsulation performance, decreasing the bit length of SIKE's primes to 434, 503, and 751-bit. This is the reason for which we find different notations in literature.

To conclude: CSIDH speed is practical while the public-key size is the smallest for key exchange or KEM in the portfolio of post-quantum cryptography. This makes CSIDH particularly attractive in the common scenario of prioritizing bandwidth over computational effort.

4.3.6 Conclusions

An order $\mathcal{O}$ of a quadratic field K is a subring $\mathcal{O} \subseteq \mathcal{O}_K$ that is also a free $\mathbb{Z}$-module of rank $2 = [K : \mathbb{Q}]$. The notion of ideal of $\mathcal{O}$ can be generalized to fractional ideals, which are sets of the form $\mathfrak{a} = \frac{1}{d}I$ where I is an ideal of $\mathcal{O}$ and $d \in \mathcal{O} \setminus \{O\}$. The invertible fractional ideals form a multiplicative group $\mathcal{I}$, which has a subgroup consisting of all principal invertible ideals $\mathcal{P}$. The ideal class group is by definition $\mathrm{Cl}(\mathcal{O}) := \mathcal{I}/\mathcal{P}$, so that in $\mathrm{Cl}(\mathcal{O})$ we identify two fractional ideals $\mathfrak{a}, \mathfrak{b}$ if there is $\alpha \in K$ such that $\mathfrak{b} = (\alpha)\mathfrak{a}$. We denote the resulting class of the fractional ideal $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$ as $[\mathfrak{a}]$. We have also shown that the ideal class group is finite and we have called its cardinality the class number of $\mathcal{O}$.
Given an elliptic curve E on a generic field, we have shown that its endomorphism ring satisfies $\mathbb{Z} \subseteq \mathrm{End}(E)$. For elliptic curves defined on a finite field, we know that $\mathbb{Z} \subsetneq \mathrm{End}(E)$. In this particular case the complete endomorphism ring $\mathrm{End}(E)$ is either an order in an imaginary quadratic field (in the case of ordinary curves) or an order in a quaternion algebra ramified at p and ∞ (in the case of supersingular curves). We have also shown that, when a supersingular curve is defined over $\mathbb{F}_p$, the ring of its $\mathbb{F}_p$-endomorphisms is isomorphic to an imaginary quadratic order, exactly as in the ordinary case.
We have seen that the endomorphism ring of an elliptic curve plays a crucial role in most algorithms for computing isogenies between curves. The class group of $\mathrm{End}(E)$ acts transitively on isomorphism classes of elliptic curves, which share the same endomorphism ring. More precisely, the class of an ideal $\mathfrak{a} \subseteq \mathcal{O}$ acts on the isomorphism class of a curve E with $\mathrm{End}(E) \cong \mathcal{O}$ through an isogeny of degree $N(\mathfrak{a})$. Likewise, every isogeny $\varphi : E \to E'$, where $\mathrm{End}(E) = \mathrm{End}(E') \cong \mathcal{O}$, corresponds (up to isomorphisms) to the class of some ideal in $\mathcal{O}$. We have shown it is possible to compute the action of an ideal in $\mathrm{Cl}(\mathcal{O})$ on a given curve: given an ideal $\mathfrak{a}$ and the l-torsion subgroup of E (where $l = N(\mathfrak{a})$), we can get the kernel of φ, and then we can derive the corresponding isogeny using Vélu's formulæ. We denote by $[\mathfrak{a}]E$ the action of the ideal class of $\mathfrak{a}$ on the isomorphism class of E. The typical strategy to evaluate the action of $[\mathfrak{a}]$ is to break it down as a product of classes of primary ideals of small norm, and evaluates the action of each prime ideal as l-isogenies. This strategy was first described by Rostovtsev and Stolbunov.

Couveignes suggested to use $\mathrm{Ell}_q(\mathcal{O})$ as an instance of a hard homogeneous space: the system parameters are a starting curve $E/\mathbb{F}_q$, and the associated class group $\mathrm{Cl}(\mathcal{O})$; the secret keys are random elements of $\mathrm{Cl}(\mathcal{O})$, and public keys are j-invariants of curves in $\mathrm{Ell}_q(\mathcal{O})$. However, given a generic element of $\mathrm{Cl}(\mathcal{O})$, the best algorithm to evaluate its action on $\mathrm{Ell}_q(\mathcal{O})$ has subexponential complexity in q, making the protocol infeasible. In-

stead, following Rostovtsev and Stolbunov, we may define a variant of Couveignes' HHS key exchange based on walks in a Cayley graph for $\mathrm{Cl}(\mathcal{O})$. The instantiation using a Schreirer graph of the HHS $\mathrm{Ell}_q(\mathcal{O})$ yields a usable variant of Couveignes' key exchange, but even with these adjustments, the protocol is still far from practical: Stolbunov managed to run a 108 bit secure implementation in around 5 minutes. Furthermore, Childs, Jao and Soukharev managed to reduce the CRS scheme to an instance of the well known abelian hidden-shift problem, for which subexponential quantum algorithms are known. Altough this represents a negative point for the protocol, its biggest limitation remains its inefficiency. Some later implementations, mainly due to De Feo, Kieffer and Smith, have improved the efficiency of this scheme, but several minutes are still needed for a single key exchange at a presumed classical security level of 128 bits.

These critical issues lead Jao and De Feo to define a protocol based on isogenies of supersingular curves, for which the corresponding endomorphisms ring is an order in a quaternion algebra, and therefore is non-commutative. The main technical idea is that we transmit the images of torsion bases under the isogeny in order to allow the parties to construct a shared commutative square despite the non-commutativity of the endomorphism ring. In the supersingular case, by contrast, the best known classical and quantum attacks against the underlying problem are both exponential in the size of the underlying finite field, since the non commutativity of the endomorphism ring means that the approach used in the ordinary case does not apply. The result is the scheme that goes by the name SIDH, and the price is the loss of a drop-in replacement for the pre-quantum Diffie-Hellman: first of all we lose the symmetry between Alice and Bob, whose roles are no longer interchangeable, and also now we share much more information than we were used to with a normal Diffie-Hellman style key exchange.

We have shown how to adapt the Couveignes-Rostovtsev-Stolbunov scheme so that it constitutes a feasible protocol. The basic idea is to use the family of supersingular elliptic curves defined over $\mathbb{F}_p$, for which the respective endomorphism ring, restricted only to maps defined on $\mathbb{F}_p$, it is still isomorphic to an order in an imaginary quadratic field. For this reason, these curves behave like ordinary curves, and the Couveignes-Rostovtsev-Stolbunov protocol carries over without modification. However, the fact that these curves necessarily have order $p + 1$ makes it extremely simple to control their group structure and class group size by appropriately choosing p from within the desired range. This close control means that we can force all of the small primes to be Elkies primes with $\lambda = 1$, which results in a speedup that beats ordinary-curve constructions by orders of magnitude. Compared to SIDH, with CSIDH we regain public-key validation and do not publish more extra points than we would expect for a Diffie-Hellman style protocol. Regarding classical and quantum attacks against this scheme, we have shown that the best classical attack has complexity $O(\sqrt[4]{p})$, while, for quantum attacks, we have considered Regev and Kuperberg's approaches which exploit the reduction of CSIDH to an instance of the abelian hidden shift problem, and solve it in subexponential time. Clearly the reduction to the hidden shift problem alone does not immediately give a subexponential-time algorithm for computing isogenies, because one must consider the time required to compute the isogeny oracle: Childs-Jao-Soukharev showed that it is possible to compute this function in subexponential time and thus obtain a subexponential-time reduction to the hidden shift problem, joining their approach with the ones described by Kuperberg or Regev. In particular: Kuperberg's algorithm for the abelian hidden shift problem uses superpolynomial space, i.e. a quantum computer with superpolynomially many qubits, so

the same is true of the most straightforward version of Childs-Jao-Soukharev algorithm. Since it is difficult to build quantum computers with many qubits, this feature could limit the applicability of this result. However, Childs-Jao-Soukharev also obtain an algorithm using polynomial space by taking advantage of an alternative approach to the abelian hidden shift problem due to Regev. Regev only explicitly considered the case of the hidden shift problem in a cyclic group whose order is a power of 2, and even in that case did not compute the constant in the exponent of the running time. Childs-Jao-Soukharev fill both of these gaps, showing that the hidden shift problem in any finite abelian group G can be solved in time $L_{|G|}[1/2, \sqrt{2}]$ by a quantum computer using only polynomial space. With these premises, we have shown that Regev and Kuperberg's approaches, joined with Childs-Jao-Soukharev, yield an overall complexity respectively of $L_p[1/2, 3/\sqrt{2}]$ and $L_p[1/2, 1/\sqrt{2}]$.

To summarize, CSIDH is a new cryptographic primitive that can serve as a drop-in replacement for the (EC)DH key-exchange protocol while maintaining security against quantum computers. Up to now we are unaware of any impact on security, negative or positive, stemming from the use of supersingular curves as opposed to ordinary curves. It provides a non-interactive (static–static) key exchange with full public-key validation. The speed is practical while the public-key size is the smallest for key exchange or KEM in the portfolio of post-quantum cryptography. This makes CSIDH particularly attractive in the common scenario of prioritizing bandwidth over computational effort, though the development of efficient side-channel-aware implementations of commutative isogeny protocols remains an open problem. At the same time we do not have to forget that all these isogeny-based protocols are more than a decade younger than all other post quantum schemes, so their security still have to withstand the tests of time and of a wide cryptanalytic effort.

Appendix A

Quantum Computing

Quantum computers are much more expressive than a normal classical computer: they do not only perfectly emulate a Turing machine, but also manage to perform some mathematical operations in much less time than a classical computer. This section has the explicit task of introducing the reader to the quantum computational model. This is not essential to understand the thesis, but it provides a more complete picture of the subject we are dealing with, making the reader more comfortable to compare with the results presented in the main discussion.

We start by describing the basic unit information on which these computers are based, namely the qubit; we then describe the way in which it is possible to perform operations on them, through objects that we will call quantum gates. With these notions we are able to understand the main quantum algorithms. First we describe the quantum Fourier transform, which can be computed in polynomial time with respect to the input size[1]. This algorithm establishes the basis for building a polynomial algorithm for the phase estimation[2], which in turns allows to solve the order finding problem in a finite group in polynomial time. Finally, the resolution of the order finding problem allows us to build an algorithm which solves the factorization problem in polynomial time. We are talking about Shor's algorithm.

A.1 Qubit

The quantum information unit is called *qubit*.

Single Qubit

In classical computer science the information unit is the *bit*, which can have a value of 0 or 1. Unlike this, the quantum computational model is based on two base states, which we denote by $|0\rangle$ and $|1\rangle$, and form an orthonormal system for the states that a qubit can assume. The value that a single qubit can take is a linear combination with complex coefficients of these two values, more precisely the value encoded by a single qubit is $|\psi\rangle$ given by

$$\begin{cases} |\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle \\ |\psi_0|^2 + |\psi_1|^2 = 1 \end{cases}$$

[1] Note that in the classical case this complexity is exponential.

[2] That is, an algorithm which estimates the eigenvalues of a matrix with particular features.

where $\psi_0, \psi_1 \in \mathbb{C}$.

Remark A.1. We do not describe how a qubit is actually implemented, in the same way as we didn't bother to describe how a classical bit is implemented, and we content ourselves only to know how it is modeled. The only thing we consider appropriate to keep in mind is that such a model is actually physically achievable, for example with the spin of a photon, which manifests the properties of the quantum mechanics: spin can be ± 1 (which our model represents with $|0\rangle$ and $|1\rangle$) or one superposition of these states (a non-trivial formulation of $\psi_0|0\rangle + \psi_1|1\rangle$). In the exact moment we observe the spin of a photon, the latter is projected onto one of its fundamental states, and it is precisely this peculiarity that our model keeps track of, thanks to the request $|\psi_0|^2 + |\psi_1|^2 = 1$, which we can interpret as follows: the probability that by measuring the spin of a photon we get 0 is $|\psi_0|^2$, conversely, the probability of measuring 1 is $|\psi_1|^2$. With these premises the variables ψ_0, ψ_1 are also called the *probability amplitudes* of the states to which they refer, as they represent the probability that measuring a qubit $|\psi\rangle$ we obtain the respective base state.

When a qubit is not in one of its fundamental states, we say that it is in *superposition*. It is important to note that, once the value of a qubit has been measured, all the information it contained is lost, as its state collapses on a state of the base. This fact is unavoidable and intrinsic in quantum mechanics.

Multiple Qubits

We describe the case of a 2 qubit system; this case generalizes to an arbitrary number of qubits in a natural way. In the case of 2 qubit, their measurement will give rise to 4 possible cases, which we identify with $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. A system of 2 qubit can therefore represent a state $|\psi\rangle$, with:

$$\begin{cases} |\psi\rangle = \psi_{00}|00\rangle + \psi_{01}|01\rangle + \psi_{10}|10\rangle + \psi_{11}|11\rangle \\ |\psi_{00}|^2 + |\psi_{01}|^2 + |\psi_{10}|^2 + |\psi_{11}|^2 = 1 \end{cases}$$

We observe that the base states described above, namely $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, are just a way to denote the state that we actually measure. Sometimes for convenience we refer to these states denoting them respectively with their decimal notation, that is $|0\rangle, |1\rangle, |2\rangle, |3\rangle$.

In general a value $|\psi\rangle$ encoded with n qubit can be expressed by a computational basis with 2^n states, for example

$$|\psi\rangle = \sum_{k=0}^{2^n-1} \psi_k |k\rangle$$

The description of a multi-qubit system is intrinsically linked to the notions of Hilbert space and tensor product, in fact any quantum system of dimension 2^n can be associated to a Hilbert space of dimension 2^n. Let us briefly recall that a Hilbert space is a vector space of finite dimension, with an inner product, complete with respect to the metric induced by the canonical norm. Among the most important properties of Hilbert spaces it should be remembered that they are unique (up to isomorphisms) for each fixed dimension, furthermore given two Hilbert spaces V and W, of dimension n and m respectively, their tensor product $V \otimes W$, that is the space whose elements are the linear combinations

of tensor products $|v\rangle \otimes |w\rangle$ where $|v\rangle \in V$ and $|w\rangle \in W$, constitutes a new Hilbert space of dimension nm. Also a base of the new space is given by the tensor products of the elements of the basis of the starting spaces. Finally, we remember that the tensor product satisfies the following important properties:

- $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$
- $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle$
- $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle$

Example A.2. Earlier we have denoted a basis for a 2 qubit system with the elements $|00\rangle, |01\rangle, |10\rangle$ and $|11\rangle$. For the results just stated, a basis for this space is also given by

$$|0\rangle \otimes |0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$$

$$|0\rangle \otimes |1\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

$$|1\rangle \otimes |0\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$$

$$|1\rangle \otimes |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

We will therefore associate the elements $|00\rangle, |01\rangle, |10\rangle$ and $|11\rangle$ to the four elements of the base described above. We will use these definitions interchangeably, thus choosing each time the most convenient to use. Clearly this construction applies to spaces of all sizes.

A.2 Quantum Gates

Just like in the classical model, also in the quantum one low-level manipulation of information takes place via gates, which are circuits that alter the information encoded inside each qubit. Since the constraints on the probabilities ψ_i of each qubit must hold both before and after the application of a gate, these conditions must be preserved.

Single Quantum Gates

Clearly a gate acting on a single qubit can be represented as a square matrix U of size 2, in addition to ensure compliance with the above condition, it is necessary and sufficient to request this matrix to be unitary, that is invertible with inverse $U^\dagger$ obtained by conjugating and transposing U.

Hadamard Gate. The Hadamard gate is the operator defined by the (unitary) matrix

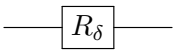
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Furthermore, the matrix that defines this operator is self-adjoint, therefore applying two times this gate to a qubit leaves the final value unchanged. The representation circuit of this operator is as follows.

Phase gate. The phase operator is defined by the (unitary) matrix

$$R(\delta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{bmatrix}$$

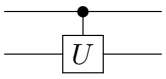
We immediately notice that the action of this gate leaves the $|0\rangle$ state unchanged and adds a phase factor $e^{i\delta}$ to $|1\rangle$. The factor phase has modulus equal to one so it does not influence the probability of the qubit to be in state $|1\rangle$. The circuit representation of this operator is the following.



Multiple Quantum Gates

Similarly, a gate acting on n qubits can be represented as a square and unitary matrix U of dimension 2^n.

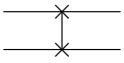
Controlled operator. Given a generic unitary matrix $U \in M_{2^n}(\mathbb{C})$, it can be seen as a quantum gate acting on n qubits. Starting from U we can define a new operator, which we denote with U^c (controlled U), which acts on $n+1$ qubit (the extra qubit is called the control qubit, the others the target). The operation works as follows: if the control qubit is zero, then the target qubit are not modified, vice-versa they undergo the action of U. The circuit representation of this operator is the following



Swap. Another operator that can be very useful is the $SWAP$ operator, associated with the matrix

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Its operation is simple: it just swap the value of two qubits. In other words $|00\rangle \mapsto |00\rangle, |01\rangle \mapsto |10\rangle, |10\rangle \mapsto |01\rangle, |11\rangle \mapsto |11\rangle$. This operator is described by the following circuit



Quantum Fourier transform. The Fourier transform is undoubtedly the most important transformation in the quantum world: on one hand its implementation is much faster than its classical version, on the other hand it forms the basis of many other quantum algorithms, first of all the Shor's one. The classical Fourier transform acts on a vector $(\psi_0, \psi_1, \ldots, \psi_{N-1}) \in \mathbb{C}^N$ and sends it to the vector $(\varphi_0, \varphi_1, \ldots, \varphi_{N-1}) \in \mathbb{C}^N$ defined by

$$\varphi_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \psi_k \omega_N^{-jk}, \quad j = 0, 1, 2, \ldots, N-1,$$

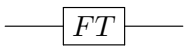
where $\omega_N = e^{\frac{2\pi i}{N}}$ is a primitive N-th roots of unity. Similarly, the quantum Fourier transform acts on a quantum state $|\psi\rangle = \sum_{k=0}^{N-1} \psi_k |k\rangle$ and maps it to another quantum state $|\varphi\rangle = \sum_{j=0}^{N-1} \varphi_j |j\rangle$ defined by

$$\varphi_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \psi_k \omega_N^{jk}, \quad j = 0, 1, 2, \ldots, N-1,$$

Conventions for the sign of the phase factor exponent vary; here we use the convention according to which the quantum Fourier transform has the same effect of the inverse discrete Fourier transform, and vice-versa. For simplicity of notation, in case there is no ambiguity we will denote the phase factor w_N simply by w. It can be shown that the (unitary) matrix associated with this transformation is

$$QFT = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)^2} \end{bmatrix} \tag{A.1}$$

Its circuit representation is given by



Given the importance of this transform, we show its circuit implementation in detail, in order to understand why the execution complexity is polynomial. To do this we must first rewrite the expression $QFT(|j\rangle)$ for a generic quantum state $|j\rangle = |j_1, ..., j_n\rangle$, in such a way that it is transparent how to implement it with the gates described above. We observe that thanks to (A.1) we have

$$\begin{aligned} \mathrm{QFT}(|j\rangle) &= \frac{1}{\sqrt{N}} \sum_{k=0}^{2^n-1} \omega^{jk} |k\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \omega^{j \sum_{l=1}^{n} k_l 2^{n-l}} |k_1 \cdots k_n\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} \omega^{j k_l 2^{n-l}} |k_l\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \omega^{j k_1 2^{n-1}} |k_1\rangle \otimes \bigotimes_{l=2}^{n} \omega^{j k_l 2^{n-l}} |k_l\rangle \end{aligned}$$

$$= \frac{1}{\sqrt{N}} \left(\sum_{k_1=0}^{1} \omega^{jk_1 2^{n-1}} |k_1\rangle \right) \otimes \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=2}^{n} \omega^{jk_l 2^{n-l}} |k_l\rangle$$

$$\vdots$$

$$= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(\sum_{k_l=0}^{1} \omega^{jk_l 2^{n-l}} |k_l\rangle \right)$$

$$= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(|0\rangle + w^{j2^{n-l}} |1\rangle \right) = (\star)$$

We now observe that $w^{j2^{n-l}} = (e^{\frac{2\pi i}{N}})^{j2^{n-l}} = \left(e^{\frac{2\pi i}{2^n}}\right)^{j2^{n-l}} = e^{2\pi i(j2^{-l})}$, also rewriting j in binary basis we get

$$j2^{-l} = 2^{-l} \sum_{r=1}^{n} j_r 2^{n-r} = \sum_{r=1}^{n} j_r 2^{n-r-l} = \sum_{r=1}^{n-l} j_r 2^{n-r-l} + \sum_{r=n-l+1}^{n} j_r 2^{n-r-l} = \alpha(l) + \beta(l)$$

Since in the first sum $r \leq n - l$, then $n - l - r \geq 0$ and $\alpha(l) \in \mathbb{N}$. This obviously does not apply to $\beta(l)$. It will be convenient to rewrite the latter quantity representing it as a binary fraction, that is, as an expression of the form $0.j_l ... j_m$ with coefficients in $\{0, 1\}$, which uniquely encodes the quantity $\frac{1}{2} j_l + \frac{1}{2^2} j_{l+1} + \cdots + \frac{1}{2^{m-l+1}} j_m$. With this premise $\beta(l) = 0.j_{n-l+1} j_{n-l+2} \cdots j_n$ and therefore

$$e^{2\pi i(j2^{-l})} = e^{2\pi i(\alpha(l))} e^{2\pi i(\beta(l))} = e^{2\pi i[0.j_{n-l+1} j_{n-l+2} \cdots j_n]}$$

We can now go back to work on the main expression, obtaining:

$$(\star) = \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{2\pi i[0.j_{n-l+1} ... j_n]} |1\rangle \right)$$

$$= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{2\pi i[0.j_n]} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i[0.j_1 ... j_n]} |1\rangle \right)$$

This reformulation of the quantum Fourier transform allows us to deduce its circuit representation. The latter will in fact be made up by Hadamard, phase, and swap gates. Let us see how this can be done.

We first observe that given a register of n qubit $|j_1\rangle \otimes \cdots \otimes |j_n\rangle$, if we apply a Hadamard gate to the first qubit we get

$$H(|j_1\rangle) \otimes |j_2 ... j_n\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i[0.j_1]} |1\rangle) \otimes |j_2 ... j_n\rangle \tag{A.2}$$

Indeed if $j_1 = 1$ then $e^{2\pi i[0.1]} = e^{\frac{2\pi i}{2}} = e^{\pi i} = -1$, otherwise if $j_1 = 0$ then $e^{2\pi i[0.0]} = e^0 = 1$.

Recall now that the R_k phase operator leaves the $|0\rangle$ state unchanged and adds a phase factor to the state $|1\rangle$ equal to $e^{\frac{2\pi i}{2^k}}$. Starting from this gate we consider the controlled operator $C - R_k$ (which now acts on two qubits) defined by the matrix

$$C - R_k := \begin{bmatrix} Id & 0 \\ 0 & R_k \end{bmatrix}$$

which applies a phase factor to the target qubit if and only if the control qubit is in state $|1\rangle$. We observe that applying $C - R_2$ to the expression (A.2) we get

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i[0.j_1 j_2]}|1\rangle) \otimes |j_2 ... j_n\rangle$$

Indeed, since $e^{2\pi i[0.j_1 j_2]} = e^{2\pi i[0.j_1]} e^{2\pi i \frac{j_2}{4}}$, if $j_2 = 0$ then no phase factor is applied, while if $j_2 = 1$ we apply a phase factor equal to $e^{\frac{2\pi i}{2^2}}$.
In a similar way we can apply in succession the gates $C - R_3, C - R_4, ..., C - R_n$ and the final state of the system will be

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i[0.j_1 j_2 ... j_n]}|1\rangle) \otimes |j_2 ... j_n\rangle$$

We can repeat the same steps for the other qubit as well and thus obtain

$$\frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i[0.j_1 j_2 ... j_n]}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i[0.j_n]}|1\rangle)$$

This expression is essentially identical (except for the qubit order) to our reformulation of $QFT(|j\rangle)$, and with at most $n/2$ swap gate we can swap the qubit of our circuit to get exactly the same expression. The work we have done so far allows us to give a circuit representation of the QFT, which we present in Figure A.1.

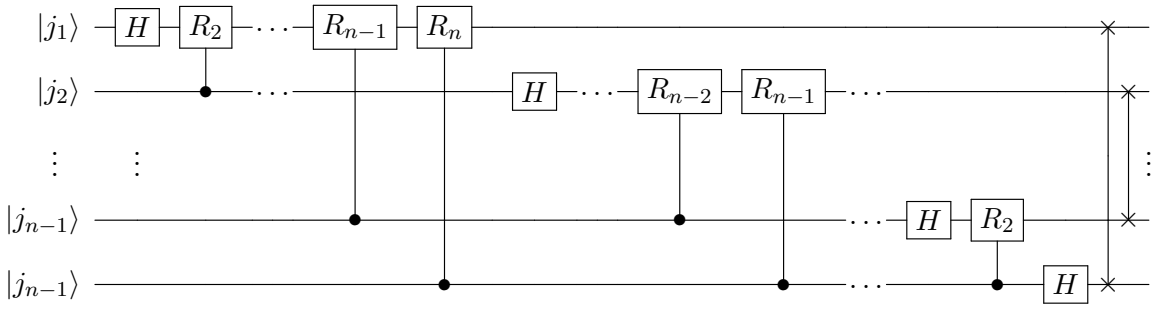


Figure A.1: Circuit representation of the QFT.

Observe that to make a circuit that allows us to compute the quantum Fourier Transform we needed $n + (n-1) + \cdots + 1 = n(n+1)/2$ gates (both Hadamard and phase gates). We also used $n/2$ swap gates. Total complexity is therefore $O(n^2)$. Consider that in the classical case, always for $N = 2^n$ elements, the best algorithms to compute the discrete Fourier transform work in $O(n2^n)$ operations.

A.3 Quantum Algorithms

The Fourier transform we just showed is the key for a lot of quantum algorithms. We briefly expose those which are relevant for our discussion.

Phase estimation

This algorithm allows to obtain an estimate of an eigenvalue of a unitary matrix starting from the corresponding eigenvector. Let us see why the algorithm has this name, or what the phase has to do with the eigenvalues of a unitary matrix. Given a generic unitary

operator U on n qubit, suppose λ is the eigenvalue corresponding to an eigenvector $|u\rangle$. Since U is unitary, its eigenvalues are complex numbers with modulo equal to 1, and

$$\lambda = e^{2\pi i\varphi} \quad \varphi \in [0, 1)$$

Therefore estimating λ is equivalent to estimate φ, which is the phase. We will not go further into this topic, and let us just remember that the circuit implementation of this algorithm is possible thanks to the QFT implementation we described earlier. Phase estimation plays a major role in the next algorithm.

Order estimation

The problem we would like to solve is to find the order r of a given element x in $\mathbb{Z}_N$. This problem has exponential complexity in $\log N$ for a classical computer, however in the quantum computing world there is an algorithm which solves the problem in $O((\log N)^3)$ operations. This algorithm essentially consists in applying the phase estimation algorithm to the unitary operator U defined by

$$U|y\rangle = |xy \pmod N\rangle$$

where $|y\rangle$ is a generic state of a qubit register, indeed we observe that each vector of the form

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i k \frac{s}{r}} |x^k \pmod N\rangle$$

is an eigenvector of U with eigenvalue $\lambda_s = e^{2\pi i \frac{s}{r}}$, indeed:

$$\begin{aligned}
U|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i k s}{r}\right] |x^{k+1} \pmod N\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=1}^{r} \exp\left[\frac{-2\pi i (k-1) s}{r}\right] |x^k \pmod N\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i (k-1) s}{r}\right] |x^k \pmod N\rangle \\
&= \exp\left[\frac{2\pi i s}{r}\right] \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i k s}{r}\right] |x^k \pmod N\rangle \\
&= \exp\left[\frac{2\pi i s}{r}\right] |u_s\rangle
\end{aligned}$$

Therefore, by applying the the phase estimation algorithm, we obtain an accurate approximation of the phase s/r. Since the resulting phase φ_s is an approximation of a rational number, if we could compute the closest fraction to φ_s, we would have a chance of getting r. There are some classical algorithms that, using the continued fractions, allow to solve this problem. Again, we do not describe the algorithm in detail, and we limit ourselves to its essential description. Total complexity occurs to be $O((\log N)^3)$.

Shor

We have laid the basis for understanding Shor's algorithm. The starting problem is the following: given a positive integer N, what is the prime factorization of N? We show that this problem is equivalent to the order finding problem discussed above.

Theorem A.3. *Let N be a composite number and let $x \in \mathbb{Z}_N$ be a solution of*

$$\begin{cases} x^2 - 1 \equiv 0 \pmod{N} \\ x \pm 1 \not\equiv 0 \pmod{N} \end{cases}$$

Then at least one of mcd$(x-1, N)$ and mcd$(x+1, N)$ is a non trivial factor of N and can be computed in $O((\log N)^3)$ operations with Euclide's algorithm.

We can use this result to our advantage in the following way: given $x \in \mathbb{Z}_N$, we know how to compute its order t in polynomial time. If its order is even, and if $x^{\frac{t}{2}}$ satisfies the equation above, i.e. $x^{\frac{t}{2}} \pm 1 \not\equiv 0 \pmod{N}$ (the condition can be relaxed with $x^{\frac{t}{2}} + 1 \not\equiv 0 \pmod{N}$, since t is the order of x), then

$$(x^{\frac{t}{2}})^2 - 1 \equiv (x^{\frac{t}{2}} - 1)(x^{\frac{t}{2}} + 1) \equiv 0 \pmod{N}$$

Then at least one between $\text{mcd}(x^{\frac{t}{2}} - 1, N)$ and $\text{mcd}(x^{\frac{t}{2}} + 1, N)$ provides a non-trivial factor of N.

The second result we recall indirectly determines the probability of finding a non-trivial factor of any composite number, and composes the last piece to be able to state and understand Shor's algorithm.

Theorem A.4. *Let N be an odd composite integer number, whose prime factorization is $N = p_1^{\alpha_1} \cdots p_m^{\alpha_m}$. If $x \in \mathbb{Z}_N$ is a random integer such mcd$(x, N) = 1$, then, denoting with r its order in $\mathbb{Z}_N$ we have that $\mathbb{P}(r$ is even and $x^{\frac{r}{2}} + 1 \not\equiv 0 \pmod{N}) > 1 - \frac{1}{2^m}$.*

With these premises Shor's algorithm naturally follows: given a composed number N we proceed as follows.

1. Choose a random integer $x \in \mathbb{Z}_N \setminus \{0\}$.
2. With Euclide we compute $\text{mcd}(x, N)$. If this value is greater of 1 then we have found a non-trivial factor of N, otherwise we proceed with the next step.
3. With the quantum algorithm to solve the order-finding problem we compute r, the order of x in $\mathbb{Z}_N$.
4. If r is odd, or if $x^{\frac{r}{2}} + 1 \equiv 0 \pmod{N}$ then we go back to step 1, otherwise we proceed.
5. With Euclide we compute $\text{mcd}(x^{\frac{r}{2}} - 1, N)$ and $\text{mcd}(x^{\frac{r}{2}} + 1, N)$. If either one turns out to be a non-trivial factor of N then the algorithm ends successfully, otherwise we start over from step 1 with a new x.

The fact that the algorithm could fail at the last step is due to the outcome of the quantum algorithm in step 3, which could give an incorrect estimate on the order r.

Bibliography

[1] G. Adj, D. Cervantes-Vázquez, J. Chi-Domínguez, A. Menezes & F. Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves, 2018. IACR Cryptology ePrint Archive 2018/313. https://ia.cr/2018/313. To appear at SAC 2018.

[2] A. Antipa, D. R. L. Brown, A. Menezes, R. Struik, and S. A. Vanstone. Validation of elliptic curve public keys. In Y. Desmedt, editor, Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings, volume 2567 of Lecture Notes in Computer Science, pages 211-223. Springer, 2003.

[3] E. Barker, W. Barker, W. Burr, W. Polk & M. Smid, Recommendation for key management: Part 1: General. National Institute of Standards and Technology, Technology Administration, 2006.

[4] E. Bellini, N. Murru, A.J. Di Scala, & M. Elia. Group law on affine conics and applications to cryptography. Applied Mathematics and Computation, 125537. https://doi.org/10.1016/j.amc.2020.125537, 2020.

[5] D.J. Bernstein, Introduction to post-quantum cryptography. Post-quantum cryptography. Springer, Berlin, Heidelberg, 2009. 1-14.

[6] D. J. Bernstein, T. Lange, and P. Schwabe. On the correct use of the negation map in the Pollard rho method. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings, volume 6571 of Lecture Notes in Computer Science, pages 128-146. Springer, 2011.

[7] J.F. Biasse, A. Iezzi, and M.J. Jacobson. A note on the security of CSIDH, 2018. https://arxiv.org/abs/1806.03656.

[8] G. Bisson. Computing endomorphism rings of elliptic curves under the GRH. J. Mathematical Cryptology, 5(2):101–114, 2012.

[9] X. Bonnetain and A. Schrottenloher. Quantum security analysis of CSIDH and ordinary isogeny-based schemes, 2018. IACR Cryptology ePrint Archive 2018/537, version 20180621:135910. https://eprint.iacr.org/2018/537/20180621:135910.

[10] R. Bröker. Constructing Elliptic Curves of Prescribed Order. PhD thesis, Universiteit Leiden, 2006.

[11] R. Bröker and P. Stevenhagen. Efficient CM-constructions of elliptic curves over finite fields. Math. Comput., 76(260):2161–2179, 2007.

[12] J. Buchmann, T. Takagi, and U. Vollmer (2004). Number field cryptography. High Primes & Misdemeanours: Lectures in Honour of the 60th Birthday of Hugh Cowie Williams, van der Poorten and Stein, eds, 41, 111-125.

[13] J. A. Buchmann and H. C. Williams. (1989, August). A key exchange system based on real quadratic fields Extended abstract. In Conference on the Theory and Application of Cryptology (pp. 335-343). Springer, New York, NY.

[14] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm. J. Cryptology, 11(2):141-145, 1998.

[15] W. Castryck, T. Lange, C. Martindale, L. Panny & J. Renes. CSIDH: an efficient post-quantum commutative group action. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2018.

[16] L. Chen, S. Jordan, Y.K. Liu, D. Moody, R. Peralta, ... & D. Smith-Tone. Report on post-quantum cryptography. Vol. 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

[17] A. M. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. J. Mathematical Cryptology, 8(1):1–29, 2014. https://arxiv.org/abs/1012.4019.

[18] H. Cohen and H. W. Lenstra, Jr. Heuristics on class groups of number fields. In Hendrik Jager, editor, Number Theory Noordwijkerhout 1983, pages 33–62. Springer, 1984.

[19] C. Costello and H. Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In ASIACRYPT (2), volume 10625 of Lecture Notes in Computer Science, pages 303–329. Springer, 2017. https://ia.cr/2017/504.413.

[20] C. Costello. Supersingular Isogeny Key Exchange for Beginners. International Conference on Selected Areas in Cryptography. Springer, Cham, 2019.

[21] C. Costello, P. Longa, and M. Naehrig. Efficient algorithms for Supersingular Isogeny Diffie–Hellman. In CRYPTO (1), volume 9814 of Lecture Notes in Computer Science, pages 572–601. Springer, 2016. https://ia.cr/2016/

[22] D.A. Cox, Primes of the form $x^2 + ny^2$, Wiley, 1989.

[23] J. Couveignes. Hard Homogeneous Spaces, 2006. IACR Cryptology ePrint Archive 2006/291. https://ia.cr/2006/291.

[24] L. De Feo. Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017).

[25] L. De Feo. Isogeny Graphs in Cryptography. Graph Theory Meets Cryptography. Wurzbürg, Germany. July 29 - August 2, 2019.

[26] C. Delfs, and S.D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Designs, Codes and Cryptography 78.2 (2016): 425-440.

[27] L. De Feo, J. Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs, 2018. IACR Cryptology ePrint Archive 2018/485. https://ia.cr/2018/485.

[28] C. Diem and E. Thomè. Index calculus in class groups of non-hyperelliptic curves of genus three. J. Cryptology, 21(4):593-611, 2008.

[29] W. Diffie & M. Hellman, 1976. New directions in cryptography. IEEE transactions on Information Theory, 22(6), 644-654.

[30] A. Enge, P. Gaudry, and E. Thomè. An L(1/3) discrete logarithm algorithm for low degree curves. J. Cryptology, 24(1):24–41, 2011.

[31] F.R.K. Chung. Diameters and eigenvalues. Journal of the American Mathematical Society, 2(2):187–196, 1989.

[32] M. Fouquet and F. Morain. Isogeny volcanoes and the SEA algorithm. In Claus Fieker and David R. Kohel, editors, Algorithmic Number Theory Symposium, volume 2369 of Lecture Notes in Computer Science, pages 47–62, Berlin, Heidelberg, 2002. Springer Berlin / Heidelberg.

[33] G. Frey, M. Müller, and H. Rück. The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. IEEE Trans. Information Theory, 45(5):1717-1719, 1999.

[34] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In CRYPTO, volume 1666 of Lecture Notes in Computer Science, pages 537–554. Springer, 1999.

[35] S.D. Galbraith. Mathematics of public key cryptography. Cambridge University Press, 2012.

[36] S.D. Galbraith, C. Petit, B. Shani, and Y.B. Ti. On the security of supersingular isogeny cryptosystems. In ASIACRYPT (1), volume 10031 of Lecture Notes in Computer Science, pages 63-91. Springer, 2016. IACR Cryptology ePrint Archive 2016/859. https://ia.cr/2016/859.

[37] S.D. Galbraith and F. Vercauteren. Computational problems in supersingular elliptic curve isogenies. Quantum Information Processing, 17, 2018. IACR Cryptology ePrint Archive 2017/774. https://ia.cr/2017/774.

[38] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. Improving the parallelized Pollard lambda search on anomalous binary curves. Math. Comput., 69(232):1699-1705, 2000.

[39] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. J. Symb. Comput., 44(12):1690–1702, 2009.

[40] P. Gaudry, F. Hess, and N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. J. Cryptology, 15(1):19-46, 2002.

[41] P. Gaudry, E. Thomè, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. Math. Comput., 76(257):475-492, 2007.

[42] O. Goldreich. Basic Facts about Expander Graphs, pages 451-464. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[43] L.K. Grover. "A fast quantum mechanical algorithm for database search." Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. 1996.

[44] A. Guillevic, F. Morain. Discrete Logarithms. N. El Mrabet; M. Joye. Guide to pairing-based cryptography, CRC Press - Taylor and Francis Group, pp.42, 2016, 9781498729505. hal-01420485v2.

[45] S. Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In STOC, pages 468–474. ACM, 2005. http://cse.psu.edu/ sjh26/unitgroup.pdf.

[46] J.L. Hafner and K.S. McCurley. A rigorous subexponential algorithm for computation of class groups. J. Amer. Math. Soc., 2(4):837–850, 1989.

[47] J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: A ring-based public key cryptosystem. International Algorithmic Number Theory Symposium. Springer, Berlin, Heidelberg, 1998.

[48] D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki–Okamoto transformation. In TCC (1), volume 10677 of Lecture Notes in Computer Science, pages 341-371. Springer, 2017. https://ia.cr/2017/604.

[49] D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. International Workshop on Post-Quantum Cryptography. Springer, Berlin, Heidelberg, 2011.

[50] D. Jao, S.D. Miller, and R. Venkatesan. Expander graphs based on GRH with an application to elliptic curve cryptography. Journal of Number Theory, 129(6), 2009.

[51] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, J. Renes, V. Soukharev, and D. Urbanik. SIKE. Submission to [49]. http://sike.org.

[52] D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In PQCrypto, volume 7071 of Lecture Notes in Computer Science, pages 19–34. Springer, 2011. https://eprint.iacr.org/2011/506/20110918:024142.

[53] D. Jao, J. LeGrow, C. Leonardi, and L. Ruiz-Lopez. A subexponential-time, polynomial quantum space algorithm for inverting the CM group action, 2018. To appear at MathCrypt 2018.

[54] J. Kieffer. étude et accélération du protocole d'échange de clés de Couveignes-Rostovtsev-Stolbunov. Mémoire du Master 2, Université Paris VI, 2017. https://arxiv.org/abs/1804.10128.

[55] N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48:203-209, 1987.

[56] N. Koblitz. Hyperelliptic cryptosystems. J. Cryptology, 1(3):139–150, 1989.

[57] D. Kohel. Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California at Berkley, 1996.

[58] G: Kuperberg. A subexponential-time quantum algorithm for the Dihedral Hidden Subgroup Problem. SIAM J. Comput., 35(1):170–188, 2005. https://arxiv.org/abs/quant-ph/0302112.

[59] G. Kuperberg. Another subexponential-time quantum algorithm for the Dihedral Hidden Subgroup Problem. In TQC, volume 22 of LIPIcs, pages 20–34. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. https://arxiv.org/abs/1112.3333.

[60] S. Lang. Algebraic number theory, volume 110 of Graduate Texts in Mathematics. Springer-Verlag, 1994.

[61] S. Lang, Elliptic functions, 2nd ed., GTM, vol. 112, Springer, 1987.

[62] A. K. Lenstra and H. W. Lenstra, Jr., editors. The development of the number field sieve, volume 1554 of Lecture Notes in Mathematics. Springer, Berlin, Heidelberg, 1993.

[63] H.W. Lenstra, A.K. Lenstra, and L.Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen, 261:515–534, 1982.

[64] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. Combinatorica, 8(3), 1988.

[65] R.J. McEliece, 1978. A public-key cryptosystem based on algebraic. Coding Thv, 4244, 114-116.

[66] A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Trans. Information Theory, 39(5):1639-1646, 1993.

[67] J.F. Mestre. La méthode des graphes. Exemples et applications. In Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata, 1986), Nagoya, 1986. Nagoya University.

[68] V. S. Miller. Use of elliptic curves in cryptography. In H. C. Williams, editor, Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings, volume 218 of Lecture Notes in Computer Science, pages 417-426. Springer, 1985.

[69] V. K. Murty. Abelian varieties and cryptography. In S. Maitra, C. E. V. Mad- havan, and R. Venkatesan, editors, Progress in Cryptology - INDOCRYPT 2005, 6th International Conference on Cryptology in India, Bangalore, India, Decem- ber 10-12, 2005, Proceedings, volume 3797 of Lecture Notes in Computer Science, pages 1-12. Springer, 2005.

[70] W. Nagao, M. Yoshifumi, and T. Okamoto. On the Equivalence of Several Security Notions of Key Encapsulation Mechanism. IACR Cryptol. ePrint Arch. 2006 (2006): 268.

[71] National Institute of Standards and Technology. Post-quantum cryptography standardization, December 2016. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

[72] P. Q. Nguyen and B. Vallée, editors. The LLL Algorithm. Springer, 2010.

[73] C. Petit. Faster algorithms for isogeny problems using torsion point images. In ASIACRYPT (2), volume 10625 of Lecture Notes in Computer Science, pages 330–353. Springer, 2017. https://ia.cr/2017/571.

[74] S. Pohlig, and M. Hellman. An improved algorithm for computing logarithms over GF (p) and its cryptographic significance (Corresp.). IEEE Transactions on information Theory 24.1 (1978): 106-110.

[75] A.K. Pizer. Ramanujan graphs and Hecke operators. Bulletin of the American Mathematical Society (N.S.), 23(1), 1990.

[76] A.K. Pizer. Ramanujan graphs. In Computational perspectives on number theory (Chicago, IL, 1995), volume 7 of AMS/IP Stud. Adv. Math. Amer. Math. Soc., Providence, RI, 1998.

[77] S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). IEEE Trans. Information Theory, 24(1):106-110, 1978.

[78] J. M. Pollard. Monte Carlo methods for index computation (mod p). Mathematics of Computation, 32(143):918-924, 1978.

[79] O. Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, 2004. https://arxiv.org/abs/quant-ph/0406151.

[80] J. Renes. Computing isogenies between Montgomery curves using the action of (0, 0). In PQCrypto, volume 10786 of Lecture Notes in Computer Science, pages 229–247. Springer, 2018. https://ia.cr/2017/1198.

[81] D. Robert. Theta functions and cryptographic applications. PhD thesis, Université Henri Poincaré - Nancy I, July 2010.

[82] D. Shanks. Class number, a theory of factorization, and genera. Proc. of Symp. Math. Soc., 1971. Vol. 20. 1971.

[83] S. A. Shirali. Groups associated with conics. The Mathematical Gazette 93.526 (2009): 27-41.

[84] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review 41.2 (1999): 303-332.

[85] J.H. Silverman. The arithmetic of elliptic curves. Vol. 106. Springer Science & Business Media, 2009.

[86] N. P. Smart. The discrete logarithm problem on elliptic curves of trace one. J. Cryptology, 12(3):193–196, 1999.

[87] B. Smith. Isogenies and the discrete logarithm problem in Jacobians of genus 3 hyperelliptic curves. J. Cryptology, 22(4):505–529, 2009.

[88] B. Smith. Pre-and post-quantum Diffie–Hellman from groups, actions, and isogenies. International Workshop on the Arithmetic of Finite Fields. Springer, Cham, 2018.

[89] A. V. Sutherland. Order computations in generic groups. PhD thesis, Massachusetts Institute of Technology, 2007.

[90] S. Tani. Claw Finding Algorithms Using Quantum Walk. arXiv:0708.2584, March 2008.

[91] T. Tao. Expansion in groups of Lie type basic theory of expander graphs. https://terrytao.wordpress.com/2011/12/02/245b-notes-1-basic-theory-of-expander-graphs/, 2011.

[92] E. Thormarker. Post-Quantum Cryptography: Supersingular Isogeny Diffie-Hellman Key Exchange. PhD thesis, Stockholm University, 2017.

[93] J. Vélu. Isogénies entre courbes elliptiques. Comptes Rendus de l'Académie des Sciences de Paris, 273:238–241, 1971.

[94] J. Voight. Quaternion algebras, 2018.

[95] L.C. Washington. Elliptic curves: number theory and cryptography. CRC press, 2008.

[96] W.C. Waterhouse. Abelian varieties over finite fields. Ann. Sci. Ecole Norm. Sup, 2(4):521-560, 1969.

[97] M. J. Wiener and R. J. Zuccherato. Faster attacks on elliptic curve cryptosystems. In S. E. Tavares and H. Meijer, editors, Selected Areas in Cryptography '98, SAC'98, Kingston, Ontario, Canada, August 17-18, 1998, Proceedings, volume 1556 of Lecture Notes in Computer Science, pages 190-200. Springer, 1998.