# Supersingular Group Actions and Post-quantum Key-exchange

Mathilde Chenu-de La Morinerie

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX120

Thèse de doctorat

# Supersingular Group Actions and Post-quantum Key Exchange

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 17 décembre 2021, par

**MATHILDE CHENU–DE LA MORINERIE**

Composition du Jury :

| | |
|---|---|
| Anne Canteaut<br>Directrice de recherche Inria Paris - COSMIQ | Présidente |
| Sylvain Duquesne<br>Professeur des universités Université Rennes 1 - IRMAR | Rapporteur |
| Damien Robert<br>Chargé de recherche Inria Bordeaux - Sud-Ouest - LFANT | Rapporteur |
| Frédéric Chyzak<br>Chargé de recherche Inria Saclay - SPECFUN | Examinateur |
| Kirsten Eisenträger<br>Professeur Pennysylvania State University - Department of Mathematics | Examinateur |
| Nicolas Sendrier<br>Directeur de recherche Inria Paris - COSMIQ | Examinateur |
| François Morain<br>Professeur École polytechnique - LIX | Directeur de thèse |
| Benjamin Smith<br>Chargé de recherche Inria Saclay - GRACE | Co-directeur de thèse |

# Contents

# Résumé en français

La cryptographie à clés publiques, ou asymétrique, découverte il y a 50 ans par Whitfield Diffie et Martin Hellman, utilise des paires de clés (une clé privée et une clé publique) pour construire des protocoles sécurisés. Elle est devenue une part essentielle des systèmes quotidiennement utilisés, en particulier pour construire des protocoles d'échanges de clés. Ces protocoles sont essentiels pour établir des clés secrètes dans le cadre de la cryptographique symétrique.

Cependant, les ordinateurs quantiques sont capables d'attaquer efficacement les problèmes de théorie des nombres garantissant la sécurité des systèmes à clés publiques les plus courants aujourd'hui, particulièrement la factorisation (sur lequel repose notamment RSA) et le logarithme discret (sur lequel repose la cryptographie basée sur les courbes elliptiques). Pour anticiper cette menace, des algorithmes post-quantiques sont actuellement développés, qui résistent à la fois aux attaques classiques et aux attaques quantiques.

Une des familles de cryptosystèmes post-quantiques repose sur les isogénies, c'est-à-dire des homomorphismes entre les courbes elliptiques. En particulier, deux protocoles d'échange de clés basés sur les isogénies sont en cours d'étude: SIDH (Supersingular Isogeny Diffie-Hellman) et CSIDH (Commutative Supersingular Isogeny Diffie-Hellman). Par ailleurs le protocole SIKE (Supersingular Isogeny Key Encapsulation), dérivé de SIDH, est actuellement en phase d'étude en vue d'une possible standardisation pour la cryptologie post-quantique. Notre problématique est la suivante: comment renforcer la confiance dans la sécurité et la faisabilité des protocoles d'échanges de clés post-quantiques basés sur les isogénies ?

Nous proposons quatre axes de réponse dans cette thèse: développer les forces de la cryptographie basée sur les isogénies, pallier ses faiblesses, généraliser les protocoles existants, et construire des attaques pour éprouver leur robustesse. Cette thèse développe ces axes dans trois parties.

La première concerne le protocole CSIDH. Nous en proposons une implémentation en temps constant, construite avec des contre-mesures envers les attaques par étude du temps d'exécution, de la consommation de courant, et par injection de fautes. Pour y parvenir, les paramètres publiques et l'espace de clé autorisés sont soigneusement choisis afin que chaque calcul soit nécessaire à l'obtention d'une clé valide. Nous proposons également une variante de ces paramètres qui permet de réaliser le protocole sans avoir recours à un générateur d'aléa.

Nous proposons dans une seconde partie une généralisation du protocole d'échange de clés de CSIDH. Pour cela nous utilisons des ensembles de courbes ayant une isogénie de degré $d$ vers leur conjuguée. Nous nommons ces couples (courbe, $d$-isogénies) des $(d, \epsilon)$-structures. Nous prouvons l'existence d'une action libre et transitive du groupe de classe d'idéaux sur des sous-ensembles des $(d, \epsilon)$-structures supersingulières, et nous utilisons cette action pour étudier la structure des graphes d'isogénies obtenus. Par la suite nous dérivons de cette étude un protocole d'échange de clés baptisé HD CSIDH pour Higher Degree Commutative Supersingular Isogeny Diffie-Hellman. Nous décrivons concrètement son utilisation, et nous en analysons la sécurité. Finalement nous développons des techniques de compression et de validation des clés spécifiquement pour HD CSIDH, et nous les comparons avec CSIDH.

Dans une troisième partie nous étudions les applications cryptanalytiques de cette nouvelle action libre et transitive, en particulier sur les protocoles SIDH et SIKE. Nous montrons qu'elle amène une généralisation de l'attaque de Delfs et Galbraith ([DG16]) sur SIDH, et nous évaluons sa complexité et son impact concret sur la sécurité. Enfin nous identifions un ensemble de courbes faibles particulièrement vulnérables à cette attaque dans le cas spécifique de SIKE et des paramètres choisis pour le premier niveau de sécurité de la spécification. Cependant nous montrons que ces attaques ne sont pour l'instant pas suffisamment efficaces pour menacer la robustesse de SIDH et SIKE.

# Chapter 1

# Introduction

## 1.1 Landscape of cryptology

Cryptology is literally the science of secrecy. It aims to ensure some or all of the following guarantees on the information exchanged:

- confidentiality: nobody other than the recipients of a message can have access to its content;
- authenticity: someone cannot pretend to send a message as someone else;
- integrity: the message cannot be modified by a third party.

Cryptography invisibly surrounds us in our every-day life such as in encrypted chats, storage of sensitive information, payments on the internet, among other examples. Considering the importance of the digital world nowadays, cryptography is more necessary than ever, for states, companies and private individuals, to ensure protection against spying and attacks on their digital data.

**Symmetric and asymmetric settings** Cryptology is often divided into two main branches: symmetric and asymmetric.

In the symmetric setting, the sender and the receiver share a common secret key that enables them to encrypt and decrypt their messages. This branch of cryptography allows fast encryption with block ciphers and stream ciphers. However some cryptographic primitives, such as signatures, cannot be achieved with symmetric cryptography. Besides, the problem of securely establishing the secret key between two parties remains. Two encryption standards chosen by the American National Institute of Standards and Technology (NIST) are symmetric block ciphers, namely the Data Encryption Standard (DES) from 1976 to 2001, replaced by the Advanced Encryption Standard (AES) since 2001.

In asymmetric settings, each party, sender and receiver, has a private key and an associated public key. Asymmetic cryptography is often slower than

its symmetric counterpart, however it also provides a different and complementary range of primitives, such as signatures, multiparty computation, key encapsulation, and especially key exchange protocols. For example, the RSA algorithm, from the name of its inventors (Rivest Shamir and Adleman) in 1977, is a famous asymmetric cryptosystem, and the basis of a widely-used signature scheme, whose security relies on the hardness of factoring. The RSA algorithm can be used for encryption, but due to its relative slowness, it is mostly used to encapsulate and exchange secret keys before being used in a symmetric protocol.

**Key exchange protocols** Key exchange protocols are crucial to ensure that two parties Alice and Bob can create a shared secret key from their respective private and public key, and later use this shared secret in symmetric encryption and/or authentification. In this sense, key exchange protocols are the bridge between the asymmetric and symmetric worlds.

Key exchange protocols have first been introduced by Diffie and Hellman in their 1976 article "New directions in cryptography" [DH76]. In this revolutionary paper, they introduce key exchange protocols as a way to provide a secure method for two parties to obtain a shared secret.

**Elliptic curves** Modern cryptography is heavily based on mathematical theory and computer science practice. In particular, a huge part of contemporary asymmetric cryptology relies on elliptic curves. These curves first appeared in cryptology in 1986, and now benefit from years of research, both in mathematics (where they have been used and studied since the $19^{th}$ century) and computer science area. Thanks to these developments, they provide fast and compact protocols that are widely used in encryption systems like Signal, Telegram or WhatsApp, signatures for e-commerce, or information encryption in biometric passports.

## 1.2 Computationally hard problems

Protocols in cryptography rely on computationally hard problems, i.e. problems that are *assumed* not to be solvable efficiently by a computer unless the underlying secret is known. This is crucial to ensure that the secret key in symmetric cryptography, or the private key in asymmetric cryptography, remain secret. Otherwise the protocol is corrupted and an attacker can decrypt messages or usurp the identity of someone else. An example of a hard computational problem used in symmetric protocols is to compute preimages of hash functions, but from now on we will focus on asymmetric protocols.

Widely used computationally hard problems for asymmetric cryptography include factorization and discrete logarithm. Note that algorithms to solve these two problems are known, but that their requirements in time or memory grow sub-exponentially or exponentially with the size of the input, making them unpractical for the sizes used in cryptography.

**Factorization** Let $p$ and $q$ be two (large) primes. Given their product $pq$ only, the *factorization problem* is to recover the factors $p$ and $q$.

For this problem, the computational effort needed to find the answer grows subexponentially with the size of the integer $pq$ to be factored. Hence, for $p$ and $q$ sufficiently large, factorizing their product becomes computationally infeasible (in the sense that the time needed would be greater than the age of the universe). The factorization problem is the underlying building block for the security of the widely used RSA scheme.

### Discrete logarithm problem

**Definition 1** (Discrete Logarithm Problem)**.** Let $(G, \times)$ be a group and $g$ a generator. Let $e$ be a secret integer. Given $g^e$ only, *the discrete logarithm problem* is to find $e$.

For this problem, the computational effort needed to find the answer depends on the underlying group $G$: it is quasi-polynomial or subexponential in finite fields, but exponential on elliptic curves. The discrete logarithm problem is the building block for elliptic curve cryptography.

**Hard Homogeneous Spaces** The discrete logarithm problem has been generalized by Couveignes [Cou06] as an instance of a Hard Homogeneous Space (HHS). Hard Homogeneous Spaces are the kind of settings that allow key exchange protocols.

**Definition 2** (Homogeneous space)**.** Let $G$ be a finite commutative group. A homogeneous space $H$ for $G$ is a finite set $H$ of the same cardinality $S = \#H = \#G$ which is acted on freely and transitively by $G$.

This definition means that there is a single orbit and for any $g \in G$ not the identity, the permutation of $H$ induced by $g$ has no fixed points. In other words, for two elements $h_1, h_2 \in H$ there is a unique $g$ in $G$ that maps $h_1$ to $h_2$. The homogeneous spaces of interest for us are the ones where the following computational problems are easy:

- Group operations for $G$: Given strings encoding of group elements $g_1$ and $g_2$, decide if they represent elements in $G$ and if these elements are equal. Given $g_1$, $g_2 \in G$ compute $g_1 g_2$, $g_1^{-1}$ and decide if $g_1 = g_2$.
- Random element for $G$: Find a random element in $G$ with uniform probability.
- Membership for $H$: Given a string $h$ decide if $h$ represents an element in $H$.
- Equality in $H$: Given $h_1$, $h_2 \in H$ decide if $h_1 = h_2$.
- Action of $G$ on $H$: Given $g \in G$ and $h \in H$ compute the action of $g$ on $h$.

For cryptographic purposes, we are interested in homogeneous spaces having additional hard computational problems. We consequently define the notion of hard homogeneous spaces.

**Definition 3** (Hard homogeneous space or HHS)**.** A hard homogeneous space $H$ for $G$ is a homogeneous space for which the following problems are hard:

- Vectorization: Given $h_1$, $h_2 \in H$, find $g \in G$ such that $h_2$ is the result of the action of $g$ on $h_1$ .
- Parallelization: Let $\delta(h_2, h_1)$ be the unique group element mapping $h_1$ to $h_2$. Given $h_1$, $h_2$, $h_3 \in H$, compute the unique $h_4$ such that $\delta(h_4, h_3) = \delta(h_2, h_1)$ .

It is conjectured (and proven in quantum settings) that parallelization and vectorization are equivalent, in the sense that if we can solve one problem efficiently, we can then use it to solve the other problem efficiently too.

## 1.3 Original Diffie–Hellman

We now present the original version of the Diffie–Hellman key exchange from [DH76] in Figure 1.1. It requires a finite cyclic group $G$ of order $n$, and a generating element $g$ in $G$.

Alice and Bob both choose random integers as private keys. They derive their public keys by exponentiating the group generator by their private keys. Both of them can compute a shared secret by applying their own private key to the counterpart's public key, thanks to the group commutativity.

Note that a passive attacker observing the information exchanged between Alice and Bob would not be able to obtain any information on the private keys. The security depends on the hardness of the Diffie–Hellman problem, which is analogous to Parallelization in a HHS, and on the Discrete Logarithm Problem, which is analogous to Vectorization.

## 1.4 Quantum revolution

Contemporary cryptography faces a major threat: the arrival of quantum computers. The publication in 1994 of Shor's algorithm [Sho94] has been a game changer. Shor proves that *with a quantum computer*, his algorithm can solve the factorization and discrete logarithm problem in polynomial time in the size of the input. This means that while these two building blocks problems remain hard against an attacker having only classical resources, they are no longer safe to be used against an attacker having access to a quantum computer.

Note that when Shor first published his algorithm, there were no quantum computers available yet. However after years of research, the development of quantum computers is rapidly growing. While bits on a classical computer have two distinct states 0 or 1, quantum bits, or qubits, can be in a superposition of

**Public parameters:**
A finite cyclic group $G$ of order $n$ (here $G$ is written multiplicatively).
A generating element $g$ in $G$.

| $\mathcal{A}$lice | | $\mathcal{B}$ob |
|---|---|---|
| **Private key:** | | **Private key:** |
| $a \in \mathbb{N}, a < n$ | | $b \in \mathbb{N}, b < n$ |
| **Public key:** | | **Public key:** |
| $PK_A = g^a$ | | $PK_B = g^b$ |
| | $PK_A \rightharpoonup$ | |
| | $\rightharpoondown PK_B$ | |
| **Shared secret computation:** | | **Shared secret computation:** |
| $(PK_B)^a \mod n$ | | $(PK_A)^b \mod n$ |
| **Shared secret:** | | **Shared secret:** |
| $g^{ab}$ | | $g^{ab}$ |

Figure 1.1: Original Diffie–Hellman protocol

both states 0 and 1. This allows new type of algorithms to be developed, namely quantum algorithms, that outperform classical ones on several computational problems, including several problems on which current cryptography is based.

Today, these quantum computers are not powerful enough to break currently used cryptography, but they might be in the near future. Current attempts are far from being enough to implement the quantum algorithm of Shor on integers of cryptographic size, which would need about 100 logical qubits, scaling up to thousands of physical qubits due to the need for error corrections. Nonetheless the power of quantum machines is rapidly growing, and large quantum machines capable of running interesting instances of Shor's algorithm could be operational in five to ten years according to some experts, rendering obsolete many of the algorithms used in cryptology. While most symmetric cryptosystems can be patched by roughly doubling the size of the keys, the current state-of-the-art in asymmetric cryptography, including elliptic-curve based cryptography, will completely collapse, because the factorization and discrete logarithm problems would be rendered easy enough to solve.

The consequences of the availability of a fully operational quantum computer would be disastrous: secure communications, digital signatures, and online payments, among others would not be safe to use any more. Considering this threat, there is an urgent need to find new protocols that would be resistant against quantum attacks. This is exactly what post-quantum cryptography is: algorithms, possibly running on classical computers, that can resist both classical and quantum adversaries.

The potential post-quantum cryptosystems have five dominant underlying mathematical techniques:

- ◇ Lattice-based systems, which rely on the hardness of problems such as finding a short vector in a given lattice;
- ◇ Code-based systems, which rely on hard problems from the theory of error correcting codes;
- ◇ Multivariate systems, which rely on the difficulty of solving various kinds of polynomial systems;
- ◇ Hash-based systems, which rely on the difficulty of inverting cryptographic hash functions;
- ◇ **Isogeny-based systems**, which take advantage of the hardness of finding paths in the graph of isogenies between ordinary or supersingular elliptic curves.

These problems are currently believed to be hard even for an attacker equipped with a quantum computer. To encourage efforts in post-quantum research, NIST (the American National Institute of Standards and Technology) has launched in 2017 a five-year-program, aiming to standardize a portfolio of quantum-resistant cryptosystems. The isogeny-based key encapsulation candidate, SIKE [JAC+17], is one of the alternate third round finalists.

## 1.5 Isogeny history

Elliptic curve cryptography has been used for years due to its efficiency and compactness. However it relies on the discrete logarithm problem which can be solved efficiently by quantum computers, triggering the need for a replacement.

Isogenies are morphisms between elliptic curves (preserving the point at infinity). In that sense isogeny-based systems naturally evolve from elliptic curve cryptography. Isogenies have been historically studied for point counting algorithms or endomorphism ring computation on elliptic curves. However, while the underlying hard problems for elliptic-curve-based systems are easily attackable by a quantum computer, the problem of finding an isogeny between two given elliptic curves remains conjecturally hard for both classical and quantum attackers. This makes isogenies a suitable candidate for post-quantum cryptography.

Isogeny-based cryptography is the youngest of post-quantum paradigm. However it benefits from years of studies made on elliptic curve cryptography, which prepared a fertile soil for its growth. It first appeared in 1996, when Couveignes proposed a key-exchange protocol based on the action of the ideal class group on an isogeny class of ordinary elliptic curves [Cou06]. Although his discovery did not spark much interest at the time, a few years later in 2004 the same scheme was independently rediscovered by Rostovstev and Stolbunov [RS06] who claimed its post-quantum security. This time it captured more attention, or at least enough attention for a quantum subexponential attack on the scheme

to be published. Indeed, Childs, Jao and Soukharev showed in 2010 the existence of a quantum subexponential attack [CJS14]. This attack, added with the fact that the scheme is inconveniently slow, despite recent steps towards greater practicability of the scheme in [DKS18], seemed to temporarily end interest on the use of isogenies between ordinary elliptic curves for cryptographic purposes.

In order to avoid the quantum subexponential attack on the ordinary case, De Feo, Jao and Plût proposed in [JD11] and [DJP14] to use isogenies between supersingular elliptic curves over $\mathbb{F}_{p^2}$ instead of ordinary ones. Indeed, the attack of [CJS14] strongly relies on the fact that the endomorphism ring of ordinary elliptic curves is commutative, which is not the case for supersingular curves over $\mathbb{F}_{p^2}$. Using a commutative diagram to replace the missing commutativity, they provide a quantum-resistant key exchange protocol *à la* Diffie–Hellman, named SIDH for Supersingular Isogeny Diffie–Hellman.

The SIDH protocol later lead to SIKE (Supersingular Isogeny Key Encapsulation), the isogeny-based proposal for the post-quantum NIST contest. It has moved on through the competition to reach the alternate third-round pool. It offers the shortest key sizes, perhaps the only one being in accessible range for practical use in some applications (less than kilobytes versus megabytes).

Attempting to improve the ordinary case key exchange protocols of CRS, Castryck, Lange, Martindale, Panny and Renes had the idea of using supersingular curves defined over $\mathbb{F}_p$ (instead of $\mathbb{F}_{p^2}$ in SIKE). The endomorphism ring over $\mathbb{F}_p$ then happens to be an order in a quadratic field, which is commutative, exactly as in the ordinary case. Using ideas from De Feo, Kieffer and Smith initially intended to accelerate the ordinary case protocol [DKS18], they proposed in [CLM$^+$18] a key exchange protocol with efficient public-key validation, and without sending additional torsion points. This scheme, named CSIDH for Commutative Supersingular Isogeny Diffie–Hellman still suffers from the subexponential quantum attack, but it offers a nice and complementary alternative to SIKE with an acceptable running time and the hope that it offers more flexibility to derive other primitives. Note that the existence of a subexponential quantum attack does not necessarily mean that the protocol is unusable: the extensively used RSA protocol also has a subexponential attack, and it is the most currently-used cryptographic protocol.

A growing number of isogeny-based protocols are being developed and studied, offering a portfolio of quantum-resistant cryptographic primitives. We give a non-exhaustive list of such primitives to give an idea of the variety of possibilities:

- Hash functions: Charles–Goren–Lauter [CLG09];
- Key exchange protocols: Couveignes [Cou06], Rostovtsev–Stolbunov [RS06], SIDH [JD11], CSIDH [CLM$^+$18], CSURF [CD20], BSIDH [Cos20], OSIDH [CK20], CTIDH [BBC$^+$21];
- Key encapsulation protocol: SIKE [JAC$^+$17];
- Signatures: SeaSign [DG18], CSI–FiSh [BKV19], SQISign [DKL$^+$20];

- Verifiable delay functions: [Wes20], [DMPS19];
- Oblivious Transfer: [Vit19];
- Threshold schemes: [DM20];

Isogeny-based cryptosystems benefit from being the natural successor of elliptic curves cryptography, taking full advantage of the years of research and confidence on curves. It is also the only post-quantum system offering a close analogue of the Diffie–Hellman key exchange protocol, as opposed to key encapsulation only.

On the downside, isogeny-based cryptography is criticised for being slower than other post-quantum families, meaning that much effort on implementations is needed to make protocols practicable. Since isogeny-based cryptography is a young field, more research is needed to increase the confidence in the underlying hard problems.

## 1.6 Problematic

Considering the advantages and drawbacks of isogeny-based cryptography described above, our problematic is the following: **how can we increase confidence in the security and practicability of isogeny-based key exchange protocols?**

We base our argumentation on four axes of confidence:

- Mitigating weaknesses: *implementation* to make key exchange fast and secure;
- Re-enforcing strengths: *key management* to provide protocols with efficient public key compression and validation;
- Constructive approach: *generalization* of existing protocols to cover the different needs of cryptography;
- Destructive approach: *cryptanalysis* to test and estimate the resistance of isogeny-based key exchanges.

## 1.7 Overview

We start by recalling in Chapter 2 the necessary mathematical preliminaries to study isogeny-based protocols. We first gather notions on quadratic fields in order to define the ideal class group in Section 2.1. Then, from algebraic plane curves in Section 2.2, we gather the tools to properly define elliptic curves in Section 2.3. We then define the morphisms between these curves, namely isogenies, in Section 2.4, and the classification of curves that arise from the structure of their endomorphism ring in Section 2.5. We make precise the action of the ideal class group in the case of ordinary and supersingular elliptic curves

in Section 2.6, which allows us to detail the structure of isogeny graphs in Section 2.7.

In Chapter 3 we introduce isogeny-based key exchanges. We detail three existing key exchange protocols : the ordinary case (CRS [Cou06, RS06, Sto10]) in Section 3.1 , the supersingular case of $\mathbb{F}_{p^2}$ (SIDH [JD11][DJP14]) in Section 3.2 and the supersingular case over $\mathbb{F}_p$ (CSIDH [CLM$^+$18]) in Section 3.3. Eventually, we compare these schemes in Section 3.5 and we briefly introduce notions of key management in cryptography in Section 3.4.

After these introductory chapters we study in Chapter 4 the constant-time implementation of the CSIDH key exchange protocol. We start by recalling several models of side-channel attacks in Section 4.1, and previous works on this subject in Section 4.2. Having gathered the necessary notions, we present a dummy-free fault-attack-resistant constant-time implementation of CSIDH in Section 4.3, as well as a derandomized variant implementation in Section 4.4. These two results are part of the joint work in [CCC$^+$19]. For completeness, we present the results published by the research community after [CCC$^+$19] in Section 4.5.

We then move on to a generalization of the CSIDH group action in Chapter 5. The results of this section have been published in [CS21]. The chapter begins with the study of curves having a degree-$d$ isogeny to their Galois conjugate in Section 5.1. We call these $(d, \epsilon)$-structures. Next we prove in Section 5.2 that there is a free and transitive action of the ideal class group of $\mathbb{Q}(\sqrt{-dp})$ on isogeny classes of $(d, \epsilon)$-structures. This result allows us to prove and describe the structure of the isogeny graph of $(d, \epsilon)$-structures in Section 5.3. We eventually show how these structures can be parametrized via modular curves in Section 5.5 and Section 5.6.

The study of the properties of $(d, \epsilon)$-structures paves the way for our Higher Degree Commutative Supersingular Isogeny Diffie–Hellman (HD CSIDH) presented in Chapter 6. We describe the key exchange protocol in Section 6.1 and detail the practical computation and related algorithms in Section 6.2, illustrated by a concrete key exchange example in Section 6.3. We then focus on key management for this new isogeny-based key exchange, in particular public key compression in Section 6.4, and public key validation in Section 6.5.

Finally we study cryptanalytic aspects of SIDH in Chapter 7. We start by recalling the Delfs–Galbraith attack from [DG16] in Section 7.1, before generalizing the approach in Section 7.2 using the tools developed for $(d, \epsilon)$-structures. We study the consequences for SIDH in Section 7.3

# Notations and conventions

- $\mathcal{K}$ is a perfect field, i.e. a field $\mathcal{K}$ in which every irreducible polynomial over $\mathcal{K}$ has distinct roots. In most applications it will be a finite field.
- $\bar{\mathcal{K}}$ is the algebraic closure of $\mathcal{K}$.
- $k$ is a quadratic field.
- $\mathcal{O}_k$ is the maximal order of $k$.
- $\mathbb{A}^n$ is the affine space over $\mathcal{K}$ of dimension $n$, $\mathbb{A}^n(\mathcal{K})$ is the set of points defined over $\mathcal{K}$, and $\mathbb{A}^n(\overline{\mathcal{K}})$ is the set of points defined over $\overline{\mathcal{K}}$.
- $\mathbb{P}^n$ is the projective space over $\mathcal{K}$ of dimension $n$, $\mathbb{P}^n(\mathcal{K})$ is the set of points defined over $\mathcal{K}$, and $\mathbb{P}^n(\overline{\mathcal{K}})$ the set of points defined over $\overline{\mathcal{K}}$.
- $n = (n_{k-1}...n_0)_2$ is the decomposition in base 2 for an integer $n$, written with least significant bits on the right.
- $(\frac{n}{p})$ is the Legendre symbol, equal to 0 if $p$ divides $n$, 1 if $n$ is a nonzero square modulo $p$, and $-1$ otherwise.
- log for the logarithm in base 2.

# Part I

# Preliminaries

# Chapter 2

# Mathematical preliminaries for isogeny-based cryptography

The building block for isogeny-based cryptography is elliptic curves. An elliptic curve is a smooth curve of genus one with a distinguished rational point. To understand this definition we start by recalling notions about quadratic fields and ideal class groups. We then study affine and projective plane curves, along with the notions of non-singularity, dimension, function fields, divisors and genus. Having gathered the tools to properly define elliptic curves, we turn to their properties: an elliptic curve is an algebraic variety but also an algebraic group. We define the addition in this group using divisors, then scalar multiplication and torsion subgroups. Finally, we introduce the invariant differential.

After studying properties of elliptic curves we focus on morphisms between such curves, namely isogenies. We define the notions of separable isogeny, dual and degree. We give Vélu's formulae, which are used to compute isogenies in practice, as well as a concrete example. We conclude the section with an introduction to modular curves an their link to isogenies. We then focus on isogenies from one curve to itself, i.e. endomorphisms, and recall how the endomorphism ring of ordinary and supersingular curves differs. We describe the Deuring correspondence which links the world of isogenies with the world of fractional ideals, and introduce the action of the ideal class group on different subset of elliptic curves. Finally we introduce isogeny graphs to describe the structure of isogenies linking curves from a given set. With graphical example we highlight the differences that arise depending on the endomorphism ring properties.

## 2.1 Quadratic imaginary order and class groups

We start by recalling notions of quadratic fields, orders, (integral) ideals, invertible and principal fractional ideals, gathering the tools to formally define the ideal class group itself.

### 2.1.1 Quadratic fields, orders and ideals

A *quadratic field* is $\mathbb{Q}(\sqrt{d})$ where $d \neq 0, 1$ is a squarefree integer. If $d < 0$ then the field is called an *imaginary quadratic field*. The discriminant of $\mathbb{Q}(\sqrt{d})$ is $D = d$ if $d \equiv 1 \pmod 4$ or $D = 4d$ otherwise.

An *order* in a field $k$ containing $\mathbb{Q}$ is a subring $R$ of $k$ that is finitely generated as a $\mathbb{Z}$-module and is such that $R \otimes_{\mathbb{Z}} \mathbb{Q} = k$. An order $\mathcal{O}$ is *maximal* if every order $\mathcal{O}'$ such that $\mathcal{O} \subset \mathcal{O}' \subset k$ is such that $\mathcal{O}' = \mathcal{O}$. Any order of a quadratic field is contained in a unique maximal order (see [BV07] Theorem 8.1.4).

**Proposition 1.** *Let $\mathcal{O}$ be an order of an imaginary quadratic field $k = \mathbb{Q}(\sqrt{d})$. The maximal order is $\mathcal{O}_k = \mathbb{Z} + \omega\mathbb{Z}$, where $\omega = \frac{1}{2}(1 + \sqrt{d})$ if $d$ is congruent to 1 modulo 4 or $\omega = \sqrt{d}$ otherwise. Moreover $\mathcal{O}$ is a submodule of the maximal order, and can be written as $\mathcal{O} = \mathbb{Z} + f\omega\mathbb{Z}$, where $f = [\mathcal{O} : \mathcal{O}_k]$ is called the* conductor.

*Proof.* See [BV07] Proposition 7.2.6 and [Gal12] Section A.12.. □

**Proposition 2.** *Let $I$ be an integral ideal of an order $\mathcal{O} = \mathbb{Z} + f\omega\mathbb{Z}$ in $\mathbb{Q}(\sqrt{d})$, where $\omega = \sqrt{d}$ if $d \not\equiv 1 \pmod 4$ and $\omega = \frac{1}{2}(1 + \sqrt{d})$ otherwise. We have $I = c(a\mathbb{Z} + (b + f\omega)\mathbb{Z})$, where $a, b$ and $c$ are integers such that $c > 0$, $a > b \geqslant 0$, and*

- *$a$ divides $b^2 - d$ if $d \not\equiv 1 \pmod 4$ ;*
- *$a$ divides $b(b + 1) - \frac{d-1}{4}$ if $d \equiv 1 \pmod 4$ .*

*Proof.* See [BV07], Equation (8.8) and Proposition 8.4.5, with $q = 1$, and $\sqrt{\Delta} = f\sqrt{-d}$. □

Recall that the *norm* of an ideal I is defined as $N(I) = |\mathcal{O}/I|$ . The norm is multiplicative, i.e. $N(IJ) = N(I)N(J)$ . An ideal $I$ strictly included in a ring $R$ is said to be a *prime ideal* if for every element $a$ and $b$ in $R$ such that $ab$ belongs to $I$, then $a$ or $b$ belongs to $I$. Using the link between integral ideals and integral binary quadratic forms, it is possible to show that every ideal whose norm is coprime to the conductor has *unique factorization* into a product of invertible prime ideals (see [BV07] Theorem 8.6.8).

**Example 1.** The order $\mathcal{O}_k = \mathbb{Z} + \frac{(1+\sqrt{-3})}{2}\mathbb{Z}$ is a maximal order of the quadratic field $k = \mathbb{Q}(\sqrt{-3})$. The suborder $\mathcal{O} = \mathbb{Z} + 5\sqrt{-3}\mathbb{Z}$ has conductor 10. Set $\omega = \frac{(1+\sqrt{-3})}{2}$. The ideal $I = 5(21\mathbb{Z} + (5 - \omega)\mathbb{Z})$ is an ideal of $\mathcal{O}_k$, with norm 525. It is the product of three prime ideals, namely $3\mathbb{Z} + (1 - 2\omega)\mathbb{Z}$ with norm 3, $7\mathbb{Z} + (4 + 2\omega)\mathbb{Z}$ with norm 7, and $5\mathbb{Z}$ with norm 25.

### 2.1.2 Fractional ideals

From integral ideals we move on to *fractional ideals* of an order $\mathcal{O}$. Since fractional ideals are not ideals, we will reserve the term ideal for integral ideals in order to avoid confusion. Fractional ideals will always be named as such.

A fractional ideal of an order $\mathcal{O}$ in a field $k$ is a subset $\mathfrak{a}$ of $k$ such that $a\mathfrak{a}$ is an (integral) ideal $\mathcal{O}$ of $k$ for some positive integer $a \in \mathbb{Z}$. A fractional ideal of an order $\mathcal{O}$ in a quadratic field $k$ is said to be *principal* if it can be written as $\mathfrak{a} = \alpha\mathcal{O}$ for some $\alpha$ in $k$. It is said to be *invertible* if there exist a fractional ideal $\mathfrak{b}$ such that $\mathfrak{ab} = \mathcal{O}$ .

In the maximal order of a quadratic field, all nonzero fractional ideals are invertible. Moreover every principal fractional ideal is invertible (see [BV07] Corollary 8.4.15, with the second point following from the definition).

The set of fractional ideal forms an abelian semigroup under multiplication (with usual multiplication of ideals). The set of invertible fractional ideals $I(\mathcal{O})$ is a subgroup, in which the set of principle ideals $Pr(\mathcal{O})$ is a normal subgroup (see [BV07] Proposition 8.4.10 and Theorem 8.4.13).

**Example 2.** Consider the quadratic imaginary field $k = \mathbb{Q}(\sqrt{-3})$ and its maximal order $\mathcal{O}_k = \mathbb{Z} + \sqrt{-3}\mathbb{Z}$. Consider the ideal $I = 6\mathbb{Z} + (3 + \sqrt{-3})\mathbb{Z}$ of $\mathcal{O}_k$. Taking $\alpha = 2 + \frac{3}{4}\sqrt{-3} \in \mathbb{Q}(\sqrt{-3})^*$, $\alpha\mathcal{O}_k$ and $\alpha I$ are fractional ideals (and not an integral ideal) of $\mathcal{O}_k$.

### 2.1.3 Ideal class group

**Definition 4.** Let $\mathcal{O}$ be an order of an imaginary quadratic field. The *ideal class group* is

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

Intuitively, this means that we will consider *equivalence classes* of invertible fractional ideals up to multiplication by an non-zero element of $\mathbb{Q}(\sqrt{d})$. For example, the fractional ideals $I$ and $\alpha I$ belong to the same *class*.

**Proposition 3.** *The order of the ideal class group of $\mathcal{O}_k$ asymptotically satisfies*

$$\log(\#\,\mathrm{Cl}(O)) \sim \log\sqrt{|d|} .$$

*Proof.* The is a special case of the Brauer–Siegel theorem (see [Lan94]). See also [BV07] Theorem 9.3.10. □

## 2.2 Algebraic plane curves

This section aims to gather all the elements needed to define elliptic curves, namely the notion of affine and projective spaces, plane curves, dimensions, smoothness and genus. See [Gal12] for references of the results in this section.

### 2.2.1 Affine plane curves

Let $\mathcal{K}$ be a perfect field. The affine 2-space over $\mathcal{K}$ is the plane $\mathbb{A}^2(\mathcal{K}) = \{(x, y) : x, y \in \mathcal{K}\}$.

An affine plane curve is defined by a single polynomial equation $f(x, y)$. An algebraic plane curve $C$ is defined over $\mathcal{K}$ if its defining polynomial is defined over $\mathcal{K}$ . We denote this by $C/\mathcal{K}$. If $C/\mathcal{K}$ is a curve defined by $f(x, y) = 0$ with $f$ a polynomial in $\bar{\mathcal{K}}[x, y]$, and $\mathcal{K}'/\mathcal{K}$ is an extension, then $C(\mathcal{K}') = \{(x, y) \in \mathbb{A}^2(\mathcal{K}') : f(x, y) = 0\}$.

### 2.2.2 Projective plane curves

Let $\mathcal{K}$ be a perfect field. The projective 2-space (over $\mathcal{K}$), denoted by $\mathbb{P}^2$ or $\mathbb{P}^2(\bar{\mathcal{K}})$ is the set of all triplets $(X, Y, Z) \in \mathbb{A}^3$ such that at least one parameter is nonzero, modulo the equivalence relation $(X, Y, Z) \simeq (X', Y', Z')$ if there exists a $\lambda \in \bar{\mathcal{K}}^*$ such that $X = \lambda X'$, $Y = \lambda Y'$, $Z = \lambda Z'$.

An equivalence class $(\lambda X, \lambda Y, \lambda Z) : \lambda \in \bar{\mathcal{K}}^*$ is denoted by $(X : Y : Z)$, and the individual $X, Y, Z$ are called homogeneous coordinates for the corresponding point in $\mathbb{P}^2$ . The set of $\mathcal{K}$-rational points in $\mathbb{P}^2$ is the set $\mathbb{P}^2(\mathcal{K}) = \{(X : Y : Z) \in \mathbb{P}^2 : X, Y, Z \in \mathcal{K}\}$.

A projective plane curve is defined by a single homogeneous polynomial equation $f(X, Y, Z) = 0$.[1] It is defined over $\mathcal{K}$ if its generating polynomial is defined over $\mathcal{K}$ . We denote this by $C/\mathcal{K}$. If $C/\mathcal{K}$ is a projective plane curve defined by $f(X, Y, Z) = 0$ with $f$ a homogeneous polynomial in $\bar{\mathcal{K}}[X, Y, Z]$, and $\mathcal{K}'/\mathcal{K}$ is an extension, then $C(\mathcal{K}') = \{(X : Y : Z) \in \mathbb{P}^2(\mathcal{K}') : f(X, Y, Z) = 0\}$.

**Example 3.** We start with an example of an affine plane curve. Let $f = x^3+7x+21-y^2 \in \mathcal{K}[x, y]$. Then $f$ defines an affine plane curve $C/\mathcal{K} = \{(x, y) \in \mathbb{A}^2 : x^3 + 7x + 21 - y^2 = 0\}$. The polynomial $F = X^3 + 7XZ^2 + 21Z^3 - Y^2Z = Z^3 f(X/Z, Y/Z) \in \mathcal{K}[X, Y, Z]$ is homogeneous, and defines a projective closure of $C$ in $\mathbb{P}^2$.

### 2.2.3 Function field

#### 2.2.3.1 Affine case

**Definition 5.** Let $C$ be a affine algebraic plane curve defined over $\mathcal{K}$ generated by a polynomial $f$. The coordinate ring of $C$ over $\mathcal{K}$ is $\mathcal{K}[C] = \mathcal{K}[x, y]/(f)$. The function field is $\mathcal{K}(C) = \{f_1/f_2 : f_1, f_2 \in \mathcal{K}[C], f_2 \notin (f)\}S$ with the equivalence relation $f_1/f_2 \equiv f_3/f_4$ if and only if $f_1f_4 - f_2f_3 \in (f)$, the ideal of $\mathcal{K}[C]$ generated by $f$.

In other words, $\mathcal{K}(C)$ is the field of fractions of the affine coordinate ring $\mathcal{K}[C]$ over $\mathcal{K}$. Elements of $\mathcal{K}(C)$ are called rational functions. For $a \in \mathcal{K}$ the rational function $f : V \mapsto k$ given by $f(P) = a$ is called a constant function.

[1] A polynomial $f \in \bar{\mathcal{K}}[X, Y, Z]$ is homogeneous of degree $d$ if $f(\lambda X, \lambda Y, \lambda Z) = \lambda^d f(X, Y, Z)$ for all $\lambda \in \bar{\mathcal{K}}$.

#### 2.2.3.2 Projective case

**Definition 6.** Let $C$ be a projective algebraic set defined over $\mathcal{K}$. The homogenous coordinate ring of $C$ over $\mathcal{K}$ is $\mathcal{K}[C] = \mathcal{K}[X, Y, Z]/f$. The function field is $\mathcal{K}(C) = \{f_1/f_2 : f_1, f_2 \in \mathcal{K}[C] \text{ homogeneous of the same degree }, f_2 \notin (f)\}$ with the equivalence relation $f_1/f_2 \equiv f_3/f_4$ if and only if $f_1 f_4 - f_2 f_3 \in (f)$.

In other words, $\mathcal{K}(C)$ is the field of fractions of the projective coordinate ring $\mathcal{K}[C]$ over $\mathcal{K}$. Elements of $\mathcal{K}(C)$ are called rational functions.

### 2.2.4 Smooth algebraic plane curves

We study the regularity, or smoothness, of a curve. As an introduction to this notion, we provide two examples in figures 2.1 and 2.2. The first former is regular whilst the latter presents a singularity at the origin. We then formally define these two notions.

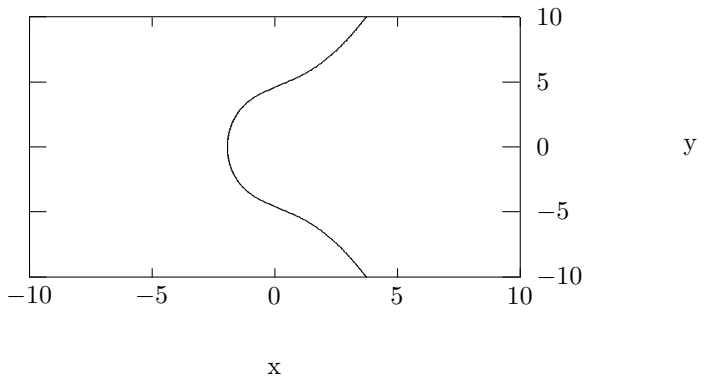



Figure 2.1: Smooth curve $y^2 = x^3 + 7x + 21$ over $\mathbb{R}$.

#### 2.2.4.1 Affine case

Let $C$ be a plane curve defined by the polynomial $f(x, y)$, and $P \in C$. Then $C$ is singular at $P$ if the partial derivatives

$$\frac{\partial(f(x,y))}{\partial x}(P) \quad \text{and} \quad \frac{\partial(f(x,y))}{\partial y}(P)$$

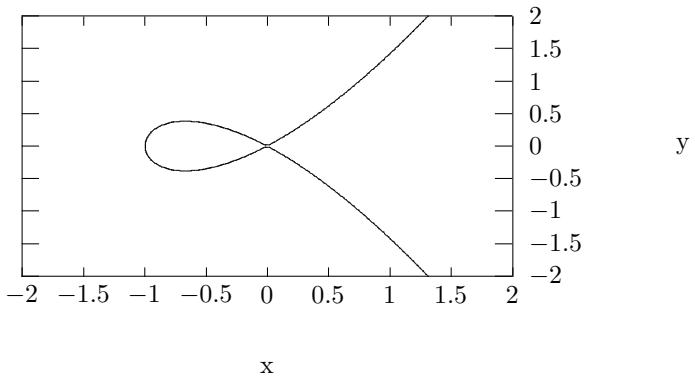



Figure 2.2: The curve over $\mathbb{R}$ defined by $y^2 = x^3 + x^2$ is singular curve at $(0, 0)$.

are both zero at $P$. If $C$ is nonsingular at every point, then we say that $C$ is nonsingular.

**Example 4.** Let $C = \{(x, y) \in \mathbb{A}^2 : x^3 + 7x + 21 - y^2 = 0\}$ and $P \in C$. The plane curve $C$ is smooth at $P$, since

$$\begin{pmatrix} \frac{\partial(x^3+7x+21-y^2)}{\partial x}(P) \\ \frac{\partial(x^3+7x+21-y^2)}{\partial y}(P) \end{pmatrix} = \begin{pmatrix} 7 + 3x^2 \\ 2y \end{pmatrix}$$

has rank 1.

#### 2.2.4.2 Projective case

Let $C$ be a projective plane curve, let $P \in C$ , and choose $\mathbb{A}^2 \subset \mathbb{P}^2$ with $P \in \mathbb{A}^2$. Then $C$ is nonsingular (or smooth) at $P$ if $C \cap \mathbb{A}^2$ is nonsingular at $P$. If $C$ is nonsingular at every point, then we say that $C$ is nonsingular.

**Example 5.** The curve $C = \{(X : Y : Z) \in \mathbb{P}^2 : X^3 + 7XZ^2 + 21Z^3 - Y^2Z = 0$ is a smooth projective plane curve. Indeed let $P \in C$ different from the point at the infinity. Then taking an affine plane $S$ with $Z \neq 0$, $C \cap S = \{(x, y) \in \mathbb{A}^2 : x^3 + 7x + 21 - y^2 = 0)$, which is smooth. For the point at the infinity, we choose another affine plane $S$ with $Y \neq 0$ and proceed similarly.

### 2.2.5 Morphisms of plane curves

Let $C$ be a plane curve and let $f \in \mathcal{K}(C)$. Then $f$ is defined or *regular* at $P$ if it can be written as $f_1/f_2$ with $f_2(P) \neq 0$ with $f_1, f_2 \in \mathcal{K}[C]$.

**Definition 7.** Let $C$ and $C'$ be two plane curves defined over $\mathcal{K}$. A rational map $\varphi : C \to C'$ over $\mathcal{K}$ which is regular at every point $P \in C(\mathcal{K})$ is called a *morphism*. If there exists a morphism $\psi : C' \to C$ over $\mathcal{K}$ such that $\varphi \circ \psi$ and $\psi \circ \varphi$ are the identity on $C'$ and $C$, respectively, then $\varphi$ is a plane curve *isomorphism*.

### 2.2.6 Divisor of a function

The divisor of a function is a formal sum representing its zeros and poles counted with multiplicities. The formal definition of a divisor requires the notion of the valuation of a function at a point. The valuation carries the information of the behaviour of the function at this point: if it is a zero (resp. a pole), the valuation is equal to its multiplicity (resp. minus its multiplicity); otherwise, the valuation is simply zero. To formally define divisors of functions, we first introduce the local ring of a variety and its maximal ideal.

**Definition 8.** Let $C$ be a plane curve over $\mathcal{K}$. The local ring over $\mathcal{K}$ of $C$ at a point $P \in C(\mathcal{K})$ is $\mathcal{O}_{P,\mathcal{K}(x,y)} = \{f \in \mathcal{K}(x, y) : f \text{ is regular at } P\}$. We write $\mathfrak{m}_{P,\mathcal{K}(x,y)} = \{f \in \mathcal{O}_{P,\mathcal{K}(x,y)} : f(P) = 0\} \subseteq \mathcal{O}_{P,\mathcal{K}(x,y)}$ for the maximal ideal of $\mathcal{O}_{P,\mathcal{K}(x,y)}$. A uniformizer for $C$ at $P$ is any generator of $\mathfrak{m}_{P,\mathcal{K}(x,y)}$.

**Definition 9.** Let $\mathcal{K}$ be a field. A discrete valuation on $\mathcal{K}$ is a function $v : \mathcal{K}^* \mapsto \mathbb{Z}$ such that:

1. for all $f, g \in \mathcal{K}^*$, $v(fg) = v(f) + v(g)$;
2. for all $f, g \in \mathcal{K}^*$ such that $f + g \neq 0$, $v(f + g) \geq \min(v(f), v(g))$;
3. there is some $f \in \mathcal{K}^*$ such that $v(f) = 1$ (equivalently, $v$ is surjective to $\mathbb{Z}$).

Let $C$ be a plane curve over $\mathcal{K}$ and $P \in C(\overline{\mathcal{K}})$. Let $\mathfrak{m}_P = \mathfrak{m}_{P,\mathcal{K}(C)}$ be as in Definition 8 and define $\mathfrak{m}_P^0 = \mathcal{O}_{P,\mathcal{K}(X)}$. Let $f \in \mathcal{O}_{P,\mathcal{K}(X)}$ be such that $f \neq 0$. Then the function $f \mapsto v_P(f) = \max\{m \in \mathbb{N} : f \in \mathfrak{m}_P^m\}$ defines a discrete valuation. We say that $v_P(f)$ is the *order* of $f$ at $P$. If $v_P(f) = 1$ then $f$ has a simple zero at $P$.

For each point $P$ on the curve, let $g_P$ and $h_P$ be functions in $\mathcal{O}_{P,\mathcal{K}(X)}$ such that $g_P/h_P = f$. The *divisor of* $f$ is $\mathrm{Div}(f) = \sum_{P \in C(\mathcal{K})} v_P(g_P)(P) - \sum_{P \in C(\mathcal{K})} v_P(h_P)(P)$. The divisor of a function is also called a principal divisor. We write $\mathrm{Prin}(C) = \{\mathrm{Div}(f) : f \in \mathcal{K}(C)^*\}$.

### 2.2.7 Divisor class group

The divisor group of a curve $C$ defined over $\mathcal{K}$, denoted by $\mathrm{Div}(C)$, is the free abelian group generated by the points of $C$. Thus a *divisor* $D \in \mathrm{Div}(C)$ is a formal sum

$$D = \sum_{P \in C(\overline{\mathcal{K}})} n_P(P) \, ,$$

where $n_P \in \mathbb{Z}$ and $n_P = 0$ for all but finitely many $P \in C(\overline{\mathcal{K}})$. The *degree* of $D$ is defined by

$$\deg D = \sum_{P \in C(\overline{\mathcal{K}})} n_P \, .$$

We write

$$\mathrm{Div}^0(C) = \{D \in \mathrm{Div}(C) : \deg(D) = 0\} \, .$$

A divisor $D = \sum_{P \in C(\overline{\mathcal{K}})} n_P(P)$ is *effective*, denoted by $D \geq 0$, if $n_P \geq 0$ for every $P \in C$. Similarly, for any two divisors $D_1, D_2 \in \mathrm{Div}(C)$, we write $D_1 \geq D_2$ to indicate that $D_1 - D_2$ is effective.

**Definition 10.** Let $C$ be a curve defined over $\mathcal{K}$ and let $D = \sum_{P \in C(\overline{\mathcal{K}})} n_P(P)$ be a divisor on $C$. For $\sigma \in \mathrm{Gal}(\overline{\mathcal{K}}/\mathcal{K})$ define $\sigma(D) = \sum_{P \in C(\overline{\mathcal{K}})} n_P(\sigma(P))$. Then $D$ is defined over $\mathcal{K}$ if $\sigma(D) = D$ for all $\sigma \in \mathrm{Gal}(\overline{\mathcal{K}}/\mathcal{K})$. We write $\mathrm{Div}_{\mathcal{K}}(C)$ for the set of divisors on $C$ that are defined over $\mathcal{K}$.

**Lemma 4.** $\mathrm{Prin}(C)$ *is a subgroup of* $\mathrm{Div}^0_{\mathcal{K}}(C)$.

*Proof.* See [Gal12] Chapter 7 Section 7 Lemma 7.7.6. □

The *degree zero divisor class group* of a curve $C$ over $\mathcal{K}$ is $\mathrm{Pic}^0(C) = \mathrm{Div}^0(C)/\mathrm{Prin}(C)$. We call two divisors $D_1, D_2 \in \mathrm{Div}^0(C)$ *linearly equivalent* and write $D_1 \equiv D_2$ if $D_1 - D_2 \in \mathrm{Prin}(C)$. The equivalence class (called a divisor class) of a divisor $D \in \mathrm{Div}^0(C)$ under linear equivalence is denoted $[D]$.

### 2.2.8 Genus

Over $\mathbb{C}$, the genus represents the number of "holes" on a curve viewed as a Riemann surface. It is formally defined for curves over any field using the divisor class group.

**Definition 11.** Let $D \in \mathrm{Div}(C)$. We associate to $D$ the set of functions $\mathcal{L}(D) = \{f \in \bar{\mathcal{K}}(C)^* : \mathrm{Div}(f) \geq -D\} \cup 0$. The set $\mathcal{L}(D)$ is a finite-dimensional $\bar{\mathcal{K}}$-vector space, and we denote its dimension by $\ell(D) = \dim_{\bar{\mathcal{K}}} \mathcal{L}(D)$.

**Theorem 5** (Riemann[2])**.** *Let $C$ be a plane curve over a field $\mathcal{K}$. There is an integer $g \geq 0$ such that $\ell(D) \geq deg(D) + 1 - g$ for all divisors $D$ on $C$ . The smallest such $g$ is called the genus of $C$ .*

*Proof.* See [Gal12] Chapter 8 Theorem 8.4.7. □

[2]this theorem is a weak form of the Riemann–Roch theorem, but we do not need the full Riemann–Roch in what follows.

## 2.3 Elliptic curves

Elliptic curves have been used in cryptography since Miller and Koblitz published independent triggering papers in 1985 [Mil85][Kob87]. Elliptic curve protocols provide speed and compact keys. Moreover they benefit from enhanced security compared to analogous algorithms for finite fields, since the discrete logarithm problem is believed to be harder to solve on elliptic curves of the same size.

We start by introducing two common representations of elliptic curves, and the $j$-invariant that allows us to identify curves up to isomorphism. Using divisors, we then show that elliptic curves are algebraic groups. From the additive group law we define scalar multiplication. We also categorize elliptic curves defined over a finite field into ordinary and supersingular ones.

### 2.3.1 Representation of elliptic curves

**Definition 12.** An *elliptic curve* defined over a field $\mathcal{K}$ is a smooth, projective, algebraic plane curve of genus one defined over $\mathcal{K}$, on which there is a distinguished point $\mathcal{O}_E$ called the infinity.

Let $E$ and $E'$ be two elliptic curves defined over $\mathcal{K}$. A *morphism* of elliptic curves $\varphi : E \to E'$ over $\mathcal{K}$ is a plane curve morphism with $\varphi(\mathcal{O}_E) = \mathcal{O}_{E'}$. If there exists a morphism $\psi : E' \to E$ over $\mathcal{K}$ such that $\varphi \circ \psi$ and $\psi \circ \varphi$ are the identity on $E'$ and $E$ respectively then $\varphi$ is an elliptic curve *isomorphism*. The curves $E$ and $E'$ are said to be isomorphic, written $E \simeq E'$.

**Proposition 6.** *Every elliptic curve is isomorphic to a curve in the projective space* $\mathbb{P}^2$ *given by the following* Weierstrass equation *:*

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

*where the coefficients* $a_1, ..., a_6$ *belong to the field* $\mathcal{K}$*. If the characteristic of the field* $\mathcal{K}$ *is different than* 2 *or* 3*, the equation can even be simplified to an affine* short Weierstrass equation *as:*

$$y^2 = x^3 + ax + b \ ,$$

*where* $a$ *and* $b$ *belong to* $\mathcal{K}$*, and* $x = X/Z$*,* $y = Y/Z$ *.*

*Proof.* See [Sil09] Chapter III section 1. □

In addition to the short Weierstrass form, in this thesis we also use extensively elliptic curves in Montgomery form.

**Definition 13.** Let $p \geq 5$ be a prime and $\mathbb{F}_p$ the finite field of order $p$. For $A, B$ in $\mathbb{F}_p$, an elliptic curve defined by

$$By^2 = x^3 + Ax^2 + x$$

is called an elliptic curve in *Montgomery* form.

Not every short Weierstrass curve has a Montgomery equivalent, however the following proposition gives a useful criteria to determine an isomorphism.

**Proposition 7.** *A Weierstrass form elliptic curve $E : y^2 = x^3 + ax + b$ is transformable to the Montgomery form if and only if*

1. *the equation $x^3 + ax + b = 0$ has at least one root in $\mathbb{F}_p$, and*
2. *the number $3\alpha^2 + a$ is quadratic residue in $\mathbb{F}_p$, where $\alpha$ is a root of the equation $x^3 + ax + b = 0$ in $\mathbb{F}_p$.*

*Let $s$ be one of the square roots of $(3\alpha^2 + a)^{-1}$ in $\mathbb{F}_p$, and set $B = s$, $A = 3\alpha s$. Then, the function mapping point $(x, y)$ on $E$ to $(s(x - \alpha), sy)$ gives an isomorphism $E$ to the Montgomery form elliptic curve defined by $By^2 = x^3 + Ax^2 + x$.*

*Proof.* See [OKS00] Proposition 1. □

The Montgomery form of curve provides numerous algorithmic improvements when using elliptic curves in protocols, including compact representation of the points by dropping the $y$ coordinate with optimized fast multiplication using the so-called Montgomery ladder (see [Mon87] for the original article and [CS18] for a survey on Montgomery curves).

**Definition 14.** In both the short Weierstrass and Montgomery models, using the notation above, we define the *$j$-invariant* of an elliptic curve $E$ defined over a field $\mathcal{K}$ by:

$$j(E) = 256\frac{(A^2 - 3)^3}{A^2 - 4} = \frac{-1728(4a)^3}{-16(4a^3 + 27b^2)} .$$

Note that in the Montgomery case, the $j$-invariant does not depend on the coefficient $B$ of the curve. Two elliptic curves defined over $\mathcal{K}$ are isomorphic over $\bar{\mathcal{K}}$ if and only if they have the same $j$-invariant (see [Sil09] chapter III section 1). We warn the reader that curves isomorphic over $\overline{\mathcal{K}}$ are not necessarily isomorphic over $\mathcal{K}$.

**Definition 15.** Let $\alpha$ be an element of $\overline{\mathbb{F}}_q \setminus \{0\}$. For each elliptic curve $E : y^2 = x^3 + ax + b$ defined over $\mathbb{F}_q$, there is a curve

$$E^\alpha/\mathbb{F}_q(\alpha^2) : y^2 = x^3 + \alpha^4 ax + \alpha^6 b$$

and an $\mathbb{F}_q(\alpha)$-isomorphism $\tau_\alpha : E \to E^\alpha$ defined by $(x, y) \mapsto (\alpha^2 x, \alpha^3 y)$. Abusing notation, we write $\tau_\alpha$ for this map on *every* elliptic curve; with this convention, $\tau_\beta \circ \tau_\alpha = \tau_{\alpha\beta}$. If $\delta$ is a nonsquare in $\mathbb{F}_q$ then $E^{\sqrt{\delta}}$ is the *quadratic twist* (which, up to $\mathbb{F}_q$-isomorphism, is independent of the choice of nonsquare $\delta$) and $\tau_{\sqrt{\delta}}$ is the twisting isomorphism.

Let $E$ be an elliptic curve over $\mathbb{F}_q$ with $char(\mathbb{F}_q) \neq 2, 3$, and $j$ its $j$-invariant. If $j \neq 0, 1728$, then the quadratic twist is the only twist up to isomorphism (see [Gal12] Lemma 9.5.7).

### 2.3.2 Algebraic group

One of the many interesting mathematical properties of elliptic curves is that they are commutative algebraic groups. Indeed, it is possible to define an addition law on the set of points of the curve. We introduce the group law using properties of divisors, before giving explicit formulae described with a graphical interpretation via the chord-and-tangent rule.

**Theorem 8.** *There is a one-to-one correspondence between the points of $E$ and $\mathrm{Pic}^0(E)$, namely $P \to (P)-(\mathcal{O}_E)$. It follows that $E$ is an algebraic group for the law induced by pulling back the divisor class group operations via this bijection.*

*Proof.* See [Sil09], Chapter III, Proposition 3.4. □

Under the bijection, the point at infinity $\mathcal{O}$ maps to 0, making $\mathcal{O}$ the neutral element of the group. It follows that if $P+Q=\mathcal{O}$ then $P=-Q$, which defines the negation of a point. It corresponds to an involution $(x,y) \to (x,-y)$ on the curve.

#### 2.3.2.1 Link with "chord-and-tangent" rule

In practice explicit formulae are derived from Theorem 8 using the "chord-and-tangent" rule. Let $P$ and $Q$ be two points on an elliptic curve. Let $l(x,y)=0$ be the line through $P$ and $Q$ (chord if $P \neq Q$, tangent to the curve otherwise).

- If $P=-Q$, then $l(x,y)=x-x_P$ is the vertical line passing through $P$. It has zeroes $P$ and $-P$, and one pole $\mathcal{O}$ with multiplicity two. Hence $(P)-(\mathcal{O})+(Q)-(\mathcal{O})=\mathrm{Div}(l)$, so $P+Q=\mathcal{O}$ under the bijection.
- If $P \neq -Q$, then the line $l(x,y)$ cuts the elliptic curve at a third point $R$. The line has zeroes $P$, $Q$ and $R$, and one pole $\mathcal{O}$ with multiplicity three. Now let $v(x)=0$ be the vertical line through $R$. It has zeroes $R$ and $-R$ and pole $\mathcal{O}$ with multiplicity two. Hence $\mathrm{Div}(l/v)=(P)+(Q)+(R)-3(\mathcal{O})-(R)-(-R)+2(\mathcal{O})=(P)+(Q)-(R)-(\mathcal{O})$. Then $(P)-(\mathcal{O})+(Q)-(\mathcal{O})=\mathrm{Div}(l/v)+(-R)-(\mathcal{O})$, so $P+Q=-R$ under the bijection.

By expressing the lines $l(x,y)$ and $v(x)$ depending on $P$, $Q$, and the model of the curve we obtain explicit formulae for the addition.

#### 2.3.2.2 Weierstrass curves

We first consider formulae for the short Weierstrass model $E: y^2=x^3+ax+b$ where $E$ is defined over a field $\mathcal{K}$. The point at infinity $\mathcal{O}$ is the identity element for the addition: for all $P \in E(\mathcal{K})$ we have $P+\mathcal{O}=\mathcal{O}+P=P$. The negation of a point $P=(x,y)$ is $-P=(x,-y)$. Let $P,Q \in E$ such that $P,Q \neq \mathcal{O}$ with $P=(x_P,y_P)$ and $Q=(x_Q,y_Q)$. If $Q=-P$ then $P+Q=\mathcal{O}$. In the remaining cases let

$$\lambda = \begin{cases} \frac{3x_P^2+a}{2y_P} & \text{if } P=Q\ , \\ \frac{y_Q-y_P}{x_Q-x_P} & \text{if } P \neq \pm Q\ , \end{cases}$$

and set $x_{P+Q} = \lambda^2 - x_P - x_Q$ and $y_{P+Q} = -\lambda(x_{P+Q} - x_P) - y_P$. Then $P + Q = (x_{P+Q}, y_{P+Q})$.

#### 2.3.2.3 Montgomery curves

We now consider formulae for the Montgomery model $E : By^2 = x^3 + Ax^2 + x$ where $E$ is defined over a field $\mathcal{K}$. As usual, $\mathcal{O}$ is the point at infinity.

**Addition** The point at infinity is the identity element for the addition: for all $P \in E(\mathcal{K})$ we have $P + \mathcal{O} = \mathcal{O} + P = P$. The negation of a point $P = (x, y)$ is $-P = (x, -y)$. Let $P, Q \in E$ such that $P, Q \neq \mathcal{O}$ with $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$. If $Q = -P$ then $P + Q = \mathcal{O}$. For the remaining cases let

$$\lambda = \begin{cases} \frac{(3x_P^2 + 2Ax_P + 1)}{(2By_P)} & \text{if } P = \pm Q \ , \\ \frac{(y_Q - y_P)}{(x_Q - x_P)} & \text{if } P \neq \pm Q \ , \end{cases}$$

and set $x_{P+Q} = B\lambda^2 - (x_P + x_Q) - A$ and $y_{P+Q} = (2x_P + x_Q + A)\lambda - B\lambda^3 - y_P = \lambda(x_P - x_{P+Q}) - y_P$. Then $P + Q = (x_{P+Q}, y_{P+Q})$ .

**Pseudo-addition** The Montgomery model allows us to compute an $x$-coordinate-only *pseudo-addition*, i.e. compute $x_{P+Q} = x(P + Q)$ in terms of $x_P = x(P)$, $x_Q = x(Q)$, and $x_{P-Q} = x(P - Q)$. Assume $P \neq Q$ and $P - Q \neq (0, 0)$ . Since $P - Q \notin \{\mathcal{O}, (0, 0)\}$, we know that $x_{P-Q} \neq 0$. If we set

$$\begin{aligned} X_{P+Q} &= [(x_P - 1)(x_Q + 1) + (x_P + 1)(x_Q - 1)] \ , \\ Z_{P+Q} &= x_{P-Q}[(x_P - 1)(x_Q + 1) - (x_P + z_P)(x_Q - z_Q)] \ , \end{aligned}$$

then $x_{p+Q} = X_{P+Q}/Z_{P+Q}$.

Most isogeny-based protocols use elliptic curves in the Montgomery form, which provides efficient arithmetic and compact data representation since points can be represented by their $x$-coordinate only. First introduced by Montgomery in [Mon87], this idea has been improved over the years as Costello and Smith outlined in [CS18]. An optimized version of the Montgomery ladder has also been introduced by Faz-Hernández, López, Ochoa-Jiménez and Rodríguez-Henríquez in [FLOR18].

### 2.3.3 Torsion

Having an addition law on the set of points of the curve, written $E(\mathcal{K})$, we can also define a multiplication by integer scalars. For an integer $m$ we define the endomorphism:

$$[m] : E(\mathcal{K}) \longrightarrow E(\mathcal{K}) \text{ by } [m]P = P + P + ... + P$$

that is the sum of $m$ copies of P. If $m < 0$, then $[m]P = [-m](-P)$. The kernel of the multiplication by $m$, i.e.

$$\ker([m]) = \{P \in E(\bar{\mathcal{K}}) : [m]P = \mathcal{O}_E\} \ ,$$

is called the *m-torsion group* of $E$, written $E[m]$. Points in this subgroup are called *m-torsion points*. If we want to restrict to the torsion group over a subfield $\mathcal{L} \subset \bar{\mathcal{K}}$, we write $E[m](\mathcal{L})$ for the *$\mathcal{L}$-rational m-torsion* subgroup.

### 2.3.4 Invariant differential

Let $C$ be a curve. The space of differential forms on $C$, denoted by $\Omega_C$ , is the $\overline{\mathcal{K}}$-vector space generated by symbols of the form $dx$ for $x \in \overline{\mathcal{K}}(C)$, subject to the usual relations:

1. $d(x+y) = dx + dy$ for all $x, y \in \overline{\mathcal{K}}(C)$.
2. $d(xy) = xdy + ydx$ for all $x, y \in \overline{\mathcal{K}}(C)$.
3. $da = 0$ for all $a \in \overline{\mathcal{K}}$

**Definition 16.** Let $E$ a elliptic curve over a field $\mathcal{K}$ defined by the Weierstrass equation $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ . The *invariant differential* is defined to be

$$\omega = \frac{dx}{2y + a_1x + a_3} \in \Omega_C \ .$$

For both short Weierstrass and Montgomery forms, we have

$$\omega = \frac{dx}{2y} \ .$$

## 2.4 Isogenies

After studying elliptic curves, we now study morphisms between these curves. They will be at the heart of the quantum-resistant cryptographic protocols. We introduce definitions and characteristic of isogenies, before providing computational details and examples. We then focus on the set of isogenies from a curve to itself, namely the endomorphism ring.

### 2.4.1 Definitions

**Definition 17.** An *isogeny* $\varphi$ is a non-constant morphism between two elliptic curves $E_1$ and $E_2$ that maps $\mathcal{O}_{E_1}$, the point at infinity of $E_1$, to $\mathcal{O}_{E_2}$, the point at infinity of $E_2$.

**Proposition 9.** *An isogeny of elliptic curves is necessarily (geometrically) surjective, and must have finite kernel. It is also a homomorphism.*

*Proof.* See [Sil09] Chap.3 Remark 4.3. □

Let $E_1/\mathcal{K}$ and $E_2/\mathcal{K}$ be curves and let $\varphi : E_1 \mapsto E_2$ be a nonconstant rational map defined over $\mathcal{K}$. Then composition with $\varphi$ induces an injection of function fields fixing $\mathcal{K}$, $\varphi^\# : \mathcal{K}(E_2) \mapsto \mathcal{K}(E_1)$, $\varphi^\# f = f \circ \varphi$, called *pullback*.

An isogeny is said to be *separable, inseparable*, or *purely inseparable* if the field extension $\mathcal{K}(E_1)/\varphi^{\#}(\mathcal{K}(E_2))$ is separable[3], inseparable, or purely inseparable respectively. A separable isogeny is defined by its kernel up to isomorphism (see [Gal12] Theorem 9.6.19).

**Definition 18.** An isogeny $\varphi : E \to E'$ is *normalised* if $\varphi^{\#}(\omega_E) = \omega_E$.

**Definition 19.** Let $G$ be a finite subgroup of an elliptic curve $E$. We define $E/G$ to be the codomain of the normalized separable isogeny with kernel $G$.

The *degree* of an isogeny is the degree of the finite extension $[\mathcal{K}(E_1) : \varphi^{\#}(\mathcal{K}(E_2))]$. For separable isogenies, the degree is also the cardinality of the kernel defining the isogeny. For a positive integer $d$, a $d$-isogeny is an isogeny of degree $d$. For every $d$-isogeny $\varphi$ from a curve $E_1$ to a curve $E_2$, there exists a *dual isogeny* with degree $d$ from $E_2$ to $E_1$ such that $\widehat{\varphi} \circ \varphi = [d]$.

Recall the definition of the quadratic twist (Definition 15). Let $E$ and $E'$ be two elliptic curves defined over $\mathbb{F}_{p^2}$. For each isogeny $\varphi : E \to E'$ defined over $\mathbb{F}_{p^2}$, and $\alpha$ an element of $\overline{\mathbb{F}}_p \setminus \{0\}$, there is an $\mathbb{F}_{p^2}(\alpha^2)$-isogeny, called the *twisted isogeny*

$$\varphi^{\alpha} := (\tau_\alpha \circ \varphi \circ \tau_{1/\alpha}) : E^{\alpha} \longrightarrow (E')^{\alpha} .$$

where $\tau_\alpha$ is the $\mathbb{F}_{p^2}(\alpha)$-isomorphism $\tau_\alpha : E \to E^{\alpha}$ defined by $(x, y) \mapsto (\alpha^2 x, \alpha^3 y)$.

Every separable isogeny $\varphi : E \to E'$ defined over $\mathcal{K}$ can be split as a (generally not unique) composition $\varphi = \varphi_1 \circ \cdots \circ \varphi_m \circ [n]$ where $(\varphi_i)_{1 \le i \le m}$ are prime degree isogenies defined over $\mathcal{K}$ and $\deg(\varphi) = n^2 \prod_{i=1}^{m} \deg(\varphi_i)$ (see [Gal12] Theorem 25.1.2).

**Definition 20.** Let $E$ be an elliptic curve over $\mathbb{F}_q$. The *isogeny class* of $E$ is the set of $\mathbb{F}_q$-isomorphism classes of elliptic curves over $\mathbb{F}_q$ that are isogenous to $E$ over $\mathbb{F}_q$.

**Theorem 10** (Tate)**.** *Two elliptic curves $E$ and $E'$ over $\mathbb{F}_q$ are $\mathbb{F}_q$-isogenous if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.*

*Proof.* See [Tat66]. □

### 2.4.2 Vélu's formulae

Now we recall Vélu's formulae [Vé71] for computing explicit normalized separable isogenies. For proofs, see [Gal12] Theorem 25.1.6. We focus on the case of prime degree $\ell$. We need different formulae for odd and even $\ell$.

Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over $\mathcal{K}$. Let $\ell$ be an odd prime and $G$ a subgroup of order $\ell$. The map $\varphi$ defined by

$$\varphi(P) = (x_P + \sum_{Q \in G \setminus \mathcal{O}_E} (x_{P+Q} - x_P), y_P + \sum_{Q \in G \setminus \mathcal{O}_E} (y_{P+Q} - y_P))$$

[3] An algebraic field extension $E \supseteq F$ is separable if for every $\alpha \in E$, the minimal polynomial of $\alpha$ over $F$ is a separable polynomial.

is invariant under translation by elements of $G$, and the kernel of $\varphi$ is $G$. Using the group law on the curve, we also see that $\varphi$ can be written in terms of rational functions. Indeed let $G^* = G \setminus \mathcal{O}_E$. Partitioning $G$ into two sets $G^+$ and $G^-$ such that $G^* = G^+ \cup G^-$, and $P \in G^+$ iff $-P \in G^-$ and for each point $P \in G^+$, we define the following quantities:

$$\begin{aligned} g_P^x &= 3x_P^2 + a \\ g_P^y &= -2y_P \\ v_P &= 2g_P^x \\ u_P &= (g_P^y)^2 \\ v &= \sum_{P \in G^+} v_P \\ w &= \sum_{P \in G^+} (u_P + x_P v_P) \ . \end{aligned}$$

Then the $\ell$-isogeny $\varphi : E \mapsto E'$ is given by

$$\varphi(x,y) = (x + \sum_{P \in G^+} (\frac{v_P}{x - x_P} - \frac{u_P}{(x-x_P)^2}), y - \sum_{P \in G^+} \frac{2yu_P}{(x-x_P)^3} + v_P \frac{y - y_P - g_P^x g_P^y}{(x-x_P)^2}) \ .$$

The equation for the image curve is $E' : y^2 = x^3 + (a - 5v)x + (b - 7w)$.

For even-order subgroup we need different formulae. Let $\ell = 2$ and $Q$ a point of order 2. The map $\varphi$ defined by

$$\varphi(P) = (x_P + (x_{P+Q} - x_P), y_P + (y_{P+Q} - y_P))$$

is invariant under translation by elements of $G = \langle Q \rangle$, and the kernel of $\varphi$ is $G$. Using the group law on the curve, we also see that $\varphi$ can be written in terms of rational functions. We define the following quantities:

$$\begin{aligned} g_G^x &= 3x_G^2 + a \\ v &= g_G^x \\ w &= x_G v_G \ . \end{aligned}$$

Then the 2-isogeny $\varphi : E \mapsto E'$ is given by

$$\varphi(x,y) = (x + \frac{v}{x - x_G}, y - v \frac{y - y_G}{(x - x_G)^2}) \ .$$

The equation for the image curve is

$$E' : y^2 = x^3 + (a - 5v)x + (b - 7w) \ .$$

### 2.4.3 Example

Let $E_1$ and $E_2$ be the elliptic curves defined over $\mathbb{F}_p$ given by the equations $y^2 = x^3 + x$ and $y^2 = x^3 - 4x$, respectively. Then

$$\begin{array}{rccc} \varphi: & E_1 & \longrightarrow & E_2 \\ & (x,y) & \longmapsto & (x+\frac{1}{x}, y-\frac{y}{x^2}) \end{array}$$

defines an isogeny of degree two with kernel generated by $(0,0)$. The curves $E_1$ and $E_2$ have the same $j$-invariant, hence they are isomorphic over $\overline{\mathbb{F}}_p$, but not necessarily over $\mathbb{F}_p$: the curves are $\mathbb{F}_p$-isomorphic if and only if $\sqrt{-1}$ is in $\mathbb{F}_p$, that is if and only if $p \equiv 1 \pmod 4$. One isomorphism $E_2 \to E_1$ is defined by $(x,y) \mapsto (-i/2x, (i+1)/4y)$, where $i = \sqrt{-1}$. It follows that the isogeny $\varphi$ is an endomorphism over $\mathbb{F}_p$ if $p \equiv 1 \mod 4$, or $\mathbb{F}_{p^2}$ if $p \not\equiv 1 \pmod 4$. This isogeny is illustrated in the case $p = 7$ in Figure 2.3.

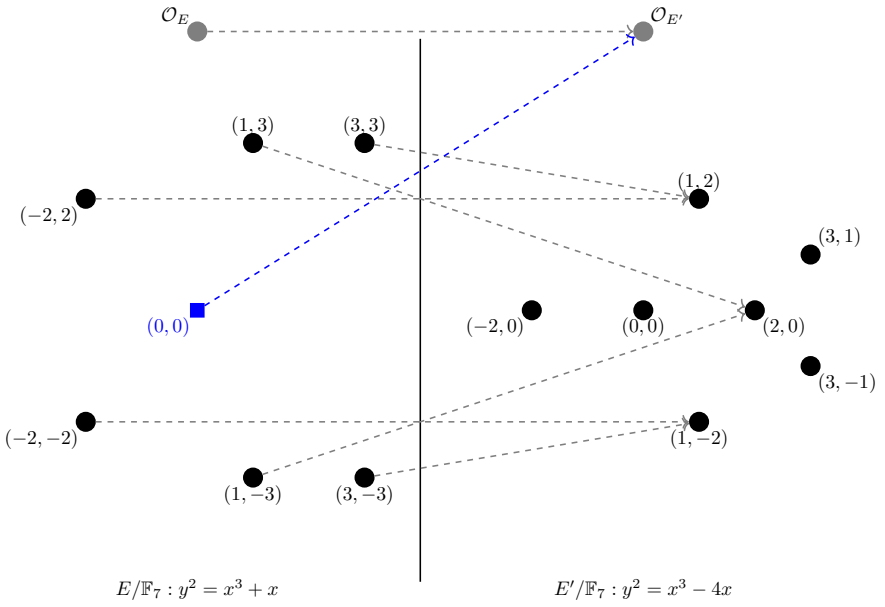



Figure 2.3: Homomorphism $E_1(\mathbb{F}_7) \to E_2(\mathbb{F}_7)$ induced by the isogeny from Section 2.4.3 with kernel $(0,0)$ between curves $E_1/\mathbb{F}_7 : y^2 = x^3 + x$ (on the left) and $E_2/\mathbb{F}_7 : y^2 = x^3 - 4x$ (on the right). The kernel $(0,0)$ is indicated with a blue square node. Arrows indicate the images of points in $E_1$ through the isogeny. Although the isogeny is surjective over $\overline{\mathbb{F}}_7$, note that over $\mathbb{F}_7$ some points of $E_2$ do not have preimages.

The dual isogeny from $E_2$ to $E_1$, with kernel generated by $(0,0)$ on $E_2$, is given by

$$\begin{array}{rccc} \widehat{\varphi}: & E_2 & \longrightarrow & E_1 \\ & (x,y) & \longmapsto & (\frac{1}{4}(x-\frac{4}{x}), \frac{1}{8}(y+4\frac{y}{x^2})) \,. \end{array}$$

### 2.4.4 Modular curves

This section introduces the correspondence between modular curves and elliptic curves with a cyclic subgroup of order $d$. Let $n$ be a positive integer.

The $n^{th}$ modular polynomial $\Phi_n(X, Y)$ defined over $\mathbb{Z}$ parametrizes pairs of elliptic curves up to isomorphism with a cyclic isogeny of degree $n$ between them, i.e

$\Phi_n(j, j') = 0 \Leftrightarrow \{$there exists an $n$-isogeny between curves of $j$-invariant $j$ and $j'$ $\}$ .

Note that $\Phi_n(X, Y) = \Phi_n(Y, X)$ from the definition and the existence of the dual isogeny. The value of $\Phi_n$ for a positive integer $n$ can be precomputed. For a given $j$-invariant $j$ in $\mathbb{F}_p$, we can also efficiently compute the polynomial $\Phi_n(j, Y)$ using the algorithm from [Sut13].

**Example 6.**

$$\begin{aligned}\Phi_2(X, Y) =& X^3 - X^2Y^2 + 1488X^2Y - 162000X^2 + 1488XY^2 + 40773375XY \\ &+ 8748000000X + Y^3 - 162000Y^2 + 8748000000Y - 157464000000000\end{aligned}$$

Let $n$ be a positive integer. The classical modular curve, written $X_0(n)$, is a completion of the affine plane curve $Y_0(N)$ defined by the classical modular polynomial $\Phi_n(X, Y)$. On this curve there is an Atkin–Lehner operator, $\omega_n$, which sends an isogeny to its dual. In terms of the modular polynomial $\Phi_n(X, Y)$, the operator $\omega_n$ swaps the coordinates $X$ and $Y$ (see [Che10] Proposition 3.6 and following discussion).

## 2.5 Endomorphisms and curve classification

### 2.5.1 The endomorphism ring

Endomorphisms are homomorphisms that map a curve to itself. The ring formed by all endomorphisms carries information about the curve itself.

**Definition 21.** The *endomorphism ring* $\mathrm{End}(E)$ (resp. $\mathcal{K}$-rational endomorphism ring $\mathrm{End}_{\mathcal{K}}(E)$) of an elliptic curve $E$ defined over a field $\mathcal{K}$ is the set of all the isogenies over $\overline{\mathcal{K}}$ (resp. over $\mathcal{K}$) from the curve to itself, with the ring operations being the pointwise addition and composition.

The scalar multiplication by any integer $m$ is an endomorphism. Moreover $[m] \neq [n]$ if and only if $m \neq n$, that is, the map from $\mathbb{Z}$ to $\mathrm{End}(E)$ is injective: $\mathbb{Z}$ is always a subring of the endomorphism ring. The *Frobenius endomorphism* of an elliptic curve $E$ over a finite field $\mathbb{F}_q$ is given by

$$\begin{aligned}\pi_q : E &\longrightarrow E \\ (x, y) &\longmapsto (x^q, y^q) .\end{aligned}$$

**Proposition 11.** *Every endomorphism of $E$ satisfies a quadratic integer polynomial.*

*Proof.* See [Gal12] Theorem 9.9.3. □

The quadratic integer polynomial satisfied by the Frobenius endomorphism is called the *characteristic polynomial of Frobenius*.

**Theorem 12** (Hasse)**.** *The characteristic polynomial of Frobenius has the form* $P(X) = X^2 - tX + q$*, where* $|t| \leq 2\sqrt{q}$*.*

*Proof.* See [Gal12] Chapter 9 Theorem 9.10.7. □

**Corollary 13.** *For a curve* $E$ *defined over a field* $\mathbb{F}_q$*, we obtain*

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

**Theorem 14** (Waterhouse)**.** *Let* $q = p^m$ *where* $p$ *is prime and let* $t \in \mathbb{Z}$ *be such that* $|t| \leq 2\sqrt{q}$*. Then there is an elliptic curve over* $\mathbb{F}_q$ *with* $\#E(\mathbb{F}_q) = q - t + 1$ *if and only if one of the following conditions holds:*

1. $\gcd(t, p) = 1$*;*
2. $m$ *is even and* $t = \pm 2\sqrt{q}$*;*
3. $m$ *is even,* $p \not\equiv 1 \pmod 3$ *and* $t = \pm\sqrt{q}$*;*
4. $m$ *is even and* $p \not\equiv 1 \pmod 4$ *and* $t = 0$*;*
5. $m$ *is odd,* $p = 2, 3$ *and* $t = \pm p^{\frac{(m+1)}{2}}$*;*
6. $m$ *is odd and* $t = 0$*.*

*Proof.* See [Gal12] Theorem 9.10.12. □

### 2.5.2 Supersingular and ordinary cases

A consequence of Proposition 11 is that the possible endomorphism rings are orders in quadratic fields and quaternion algebras. The endomorphism ring gives some information about the curve itself, since its type allows to classify the curves as *ordinary* or *supersingular*. Each type has special properties that are used in different cryptographic protocols.

**Definition 22.** Let $E$ be an elliptic curve defined over a field $\mathcal{K}$ of characteristic $p$. Then $E$ is *supersingular* if and only if $E[p] = E[p](\overline{k}) = \mathcal{O}_E$. Otherwise it is *ordinary*.

**Proposition 15** (Ordinary curves)**.** *The endomorphism ring of an ordinary elliptic curve defined over a finite field* $\mathbb{F}_q$ *is an order in an imaginary quadratic field* $k$*, i.e.* $End(E) \otimes \mathbb{Q} \simeq \mathbb{Q}(\sqrt{d})$*, where* $d = |t^2 - 4q|$ *and* $t$ *is the trace of the Frobenius endomorphism.*

*Proof.* See [Koh96] Chapter 4. □

Let $\mathcal{O}$ be an order of an imaginary quadratic field $k = \mathbb{Q}(\sqrt{d})$. The *Hilbert class polynomial* is the monic polynomial $h_d$ whose roots are the distinct $j$-invariants of all elliptic curves with endomorphism ring isomorphic to $\mathcal{O}$.

**Example 7.** The Hilbert class polynomial for $\mathbb{Q}(\sqrt{-23})$ is $h_{-23}(x) = x^3 + 3491750x^2 - 5151296875x + 12771880859375$ .

**Proposition 16** (Supersingular curves)**.** *The endomorphism ring* over $\overline{\mathbb{F}}_p$ *of a supersingular elliptic curve* [4] *is an order in a quaternion algebra, i.e.* $End(E) \otimes \mathbb{Q} \simeq \mathbb{Q}\langle i, j\rangle$ *where* $ij = -ji$*, and* $i^2, j^2$ *belong to* $\mathbb{Q}$*.*

*Proof.* See [Gal12] Theorem 9.11.2. □

## 2.6 Deuring correspondence and the action of the ideal class group

### 2.6.1 Action of the ideal class group on elliptic curves

For $\mathcal{O}$ an order of an imaginary quadratic field $k$, the ideal class group $\mathrm{Cl}(\mathcal{O})$ acts on the set of ordinary elliptic curves with $\mathbb{F}_p$-rational endomorphism ring $\mathcal{O}$, but also on special subsets of the class of supersingular elliptic curves. We start by recalling notions on group actions, before describing the specific properties of the ideal class group action for each type of elliptic curves.

**Definition 23** (Action)**.** Let $G$ be a group with identity element $e$. Let $X$ be a set. A (left) group action $\alpha$ of $G$ on $X$ is a function

$$\alpha : G \times X \to X ,$$

(with $\alpha(g, x)$ often shortened to $gx$ or $g \cdot x$ when the action being considered is clear from context), that satisfies the following two properties:

1. Identity: $e \cdot x = x$ for all $x$ in $X$
2. Compatibility: $g \cdot (h \cdot x) = (gh) \cdot x$ for all $g$ and $h$ in $G$ and all $x$ in $X$.

A set $X$ together with an action of $G$ is called a (left) $G$-set.

The action is *free* if, given $g$, $h$ in $G$, the existence of an $x$ in $X$ with $g \cdot x = h \cdot x$ implies $g = h$. The action is *transitive* if $X$ is non-empty and if for each pair $x$, $y$ in $X$ there exists a $g$ in $G$ such that $g \cdot x = y$.

For any elliptic curve $E$ defined over a finite field, and for any order $\mathcal{O}$ of a quadratic field such that $\mathcal{O} \subset \mathrm{End}(E)$, we can define an action of $\mathrm{Cl}(\mathcal{O})$ on $E$. However, what will be important for us is to determine on which set of elliptic curves the action is free and transitive, since these are useful properties when building key exchange protocols.

Let $E$ be an elliptic curve defined over a finite field $\mathbb{F}_q$, with $q = p^n$. Let $\pi_p$ be the $p$-power Frobenius. Let $\mathfrak{p}$ be a prime ideal over $p$ corresponding to the isogeny $\pi_p$. Let $\mathcal{O}$ be an order of a quadratic field $k$ such that $\mathcal{O} \subset \mathrm{End}(E)$. Let $[\mathfrak{a}]$ be an element of $\mathrm{Cl}(\mathcal{O})$ with integral representative $\mathfrak{a}$ and let $r$ and $\mathfrak{a}'$ be such that $\mathfrak{a} = (\mathfrak{p})^r\mathfrak{a}'$, where $\mathfrak{a}'$ is integral and not contained in $\mathfrak{p}$ (the existence of $r$ and $\mathfrak{a}'$ follow from unique factorization of ideals in $\mathcal{O}$).

[4] When considering endomorphisms defined over $\mathbb{F}_p$ only, the restricted endomorphism ring is not a quaternion algebra any more, but a quadratic imaginary order. See Theorem 19.

**Definition 24.** With the notation above: Let $E[\mathfrak{a}] := \bigcap_{\alpha \in \mathfrak{a}} \ker(\alpha)$ for the ideal $\mathfrak{a} \subset \mathrm{End}(E)$. We define $\phi_{\mathfrak{a}} : E \to E/E[\mathfrak{a}]$ to be the isogeny (up to isomorphism) whose separable part is the isogeny with kernel $\bigcap_{\alpha \in \mathfrak{a}'} \ker \alpha$ , and whose purely inseparable part is $r$ iteration of the Frobenius $\pi_p$. The image of $E$ under the action of $\mathfrak{a}$ is the codomain of the isogeny $\phi_{\mathfrak{a}}$.

**Theorem 17.** *We keep the same notation as above, and write $[E]$ for the isomorphism class of curves with $\mathcal{O} \subset \mathrm{End}(E)$. The map $(\mathfrak{a}, [E]) \mapsto [E/E[\mathfrak{a}]]$ defines an action of $\mathrm{Cl}(\mathcal{O})$ on the set of isomorphism classes.*

*Proof.* See [Wat69] Section 3.2 Kernel ideals. □

#### 2.6.1.1 The ordinary case

Let $\mathcal{O}$ be an order of $\mathbb{Q}(\sqrt{d})$. In the ordinary case, the action of the ideal class group $\mathrm{Cl}(\mathcal{O})$ is free and transitive on the set of curves having same cardinality over their base field and their endomorphism ring isomorphic to $\mathcal{O}$.

**Theorem 18.** *Let $\mathrm{Ell}_p(\mathcal{O}, \pi_p)$ be the set of elliptic curves $E$ defined over $\mathbb{F}_p$ with $\mathrm{End}_p(E) \simeq \mathcal{O}$ such that $\pi_p$ corresponds to the $\mathbb{F}_p$-Frobenius endomorphism of $E$. Let $\mathcal{O}$ be an order in an imaginary quadratic field that $\mathrm{Ell}_p(\mathcal{O}, \pi_p)$ is non-empty. Then the ideal-class group $\mathrm{Cl}(\mathcal{O})$ acts freely and transitively on the set $\mathrm{Ell}_p(\mathcal{O}, \pi_p)$ via the map*

$$\begin{aligned} \mathrm{Cl}(\mathcal{O}) \times \mathrm{Ell}_p(\mathcal{O}, \pi_p) &\longrightarrow \mathrm{Ell}_p(\mathcal{O}, \pi_p) \\ ([\mathfrak{a}], E) &\longmapsto E/E[\mathfrak{a}] \end{aligned}$$

*in which $\mathfrak{a}$ is chosen as an integral representative.*

*Proof.* See [Wat69] Theorem 4.5. □

#### 2.6.1.2 The supersingular case over $\mathbb{F}_p$

In the supersingular case, we can define a free and transitive group action on (sub)set of supersingular elliptic curves defined over $\mathbb{F}_p$.

**Theorem 19.** *Let $\mathcal{O}$ be $\mathbb{Z}[\pi_p]$ or $\mathbb{Z}[\frac{1+\pi_p}{2}]$ ) Let $\mathcal{S}$ be the set of supersingular elliptic curves over $\mathbb{F}_p$ with endomorphism ring over $\mathbb{F}_p$ equal to $\mathcal{O}$ . The ideal class group $Cl(\mathcal{O})$ acts freely and transitively on $\mathcal{S}$.*

*Proof.* See [Wat69] Theorems 4.5. □

#### 2.6.1.3 The supersingular case over $\mathbb{F}_{p^2}$

In the supersingular case over $\mathbb{F}_{p^2}$, the endomorphism ring is a maximal order in a non-commutative quaternion algebra. Hence we cannot have a free and transitive action from the ideal class group of a quadratic order on the entire set of curves. However there are subsets in which the action is free and transitive. See Chapter 5 and [Onu21]. In Chapter 6 we extend this action to subsets of supersingular elliptic curves equipped with distinguished isogenies.

### 2.6.2 Deuring correspondence

For elliptic curves whose endomorphism ring is an order $\mathcal{O}$ in a quadratic field, Deuring established the following correspondence between isogenies up to isomorphism and elements of the ideal class group, allowing another representation of isogenies.

**Theorem 20.** *Let $E$ be an elliptic curve defined over a field $\mathcal{K}$ whose endomorphism ring over $\mathcal{K}$ is isomorphic to an order $\mathcal{O}$ in a quadratic field. Let $E'$ be an elliptic curve isogenous to $E$. We have the following dictionary between fractional ideals and isogenies.*

| | |
|---|---|
| *Endomorphism ring* $\mathrm{End}(E)$ | *Order $\mathcal{O}$ of a quadratic field* |
| *Isogenies from $E$ to $E'$* | *Invertible fractional ideals $I(\mathcal{O})$* |
| *Endomorphisms of $E$* | *Principal fractional ideals $Pr(\mathcal{O})$* |
| *Isogeny composition* | *Ideal multiplication* |
| *Dual isogeny* | *Inverse* |

In this thesis, we will use Theorem 20 in the case of ordinary curves and supersingular curves defined over $\mathbb{F}_p$, for which the endomorphism ring *over their base field* is a order of a (commutative) quadratic field.

*Proof.* [Wat69] Chapters 4 for the general case, and [Voi17] Chapter 42 in the supersingular case over $\mathbb{F}_p$. □

Since principal fractional ideals correspond to endomorphisms of the curve, and invertible fractional ideals are associated with isogenies, we quotient the abelian subgroup of invertible fractional ideals $I(\mathcal{O})$ by the normal subgroup of principal fractional ideals $Pr(\mathcal{O})$ in order to keep only isogenies and "kill" endomorphisms. The equivalence classes of the ideal class group $\mathrm{Cl}(\mathcal{O})$ hence corresponds to isogenies up to endomorphism.

## 2.7 Isogeny graphs

**Definition 25.** Let $\mathcal{K}$ be a finite field of characteristic $p$, or its algebraic closure. Let $L$ be a set of primes not including $p$. The isogeny graph $\Gamma(\mathcal{K}, L)$ is the directed graph where the vertices are $\mathcal{K}$-isomorphism classes of elliptic curves defined over $\mathcal{K}$, and the edges are classes of $\mathcal{K}$-isogenies with degree $\ell \in L$ between the curves. We write $\Gamma(\mathcal{K}, \ell)$ when we consider only $L = \{\ell\}$.

### 2.7.1 Ordinary case

In the ordinary case, the $\ell$-isogeny graph for a prime $\ell$ resembles a volcano, as defined by [FM02]. An $\ell$-volcano $V$ is a connected undirected graph whose

vertices are partitioned into one or more levels $V_0, ..., V_d$ such that the following hold:

1. The subgraph on $V_0$ (the surface) is a regular graph of degree at most 2.
2. For $i > 0$, each vertex in $V_i$ has exactly one neighbour in level $V_{i-1}$, and this accounts for every edge not on the surface.
3. For $i < d$, each vertex in $V_i$ has degree $\ell + 1$.

Level $V_d$ is called the floor of the volcano; the floor and surface coincide when $d = 0$.

**Theorem 21** ([Koh96])**.** *Let $\mathbb{F}_q$ be a finite field, let $\ell \nmid q$ be a prime, let $\mathfrak{l}$ be an ideal above $\ell$, and let $V$ be an ordinary component of $\Gamma(\mathbb{F}_q, \ell)$ that does not contain curves with $j$-invariants 0 or 1728. We write $t_{\pi_p}$ for the trace of the Frobenius (every curve in the graph having the same cardinality over $\mathbb{F}_q$, hence the same trace). Then $V$ is an $\ell$-volcano for which the following hold:*

1. *The depth of $V$ is $d$, where $d$ is such that $4q = (t_{\pi_p})^2 - \ell^{2d} v^2 disc(\mathcal{O}_0)$ with $\ell \nmid v$.*
2. *The vertices in level $V_i$ all have endomorphism ring isomorphic to the same order $\mathcal{O}_i$.*
3. *$\ell \nmid [\mathcal{O}_d : \mathcal{O}_0]$ and $[\mathcal{O}_i : \mathcal{O}_{i+1}] = \ell$ for $0 \leq i < d$.*
4. *The subgraph on $V_0$ has degree $1 + (\frac{disc(\mathcal{O}_0)}{\ell})$.*
5. *If $(\frac{disc(\mathcal{O}_0)}{\ell}) \geq 0$, then $|V_0|$ is the order of $[\mathfrak{l}]$ in $\mathrm{Cl}(\mathcal{O}_0)$; otherwise $|V_0| = 1$.*

*Proof.* Although the theorem is originally from [Koh96] Proposition 23, the volcano terminology first appear in [FM02]. See also [FM02] in particular Lemmas 2.3, 2.4, and 2.5 and [Gal12] 25.4.6. □

Figure 2.4 illustrates a 2-isogeny volcano of depth 3.

### 2.7.2 Supersingular case over $\mathbb{F}_p$

In the supersingular case over $\mathbb{F}_p$, the $\ell$-isogeny graph for a prime $\ell$ is again a volcano, but with limited depth: for $\ell = 2$ it has at most two levels, and for any other $\ell$ only one.

**Theorem 22** ([DG16])**.** *Let $p > 3$ be a prime.*

1. *If $p \equiv 1 \pmod 4$, then there are $h(-4p)$ $\mathbb{F}_p$-isomorphism classes of supersingular elliptic curves over $\mathbb{F}_p$, all having the same endomorphism ring $\mathbb{Z}[\sqrt{-p}]$. From every one there is one $\mathbb{F}_p$-rational horizontal 2-isogeny as well as two horizontal $\ell$-isogenies for every prime $\ell > 2$ with $(\frac{-p}{\ell}) = 1$.*

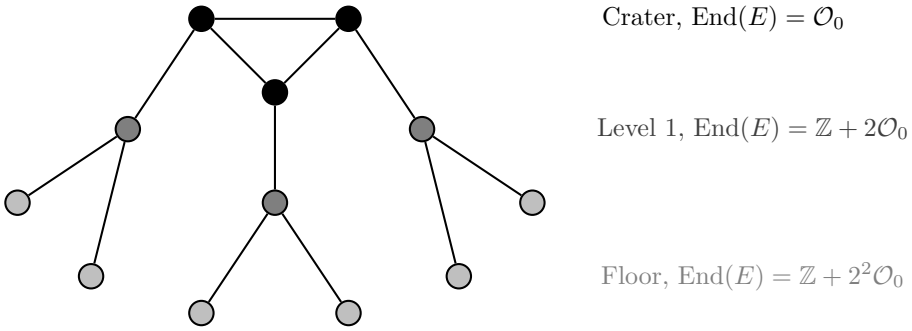



Figure 2.4: 2-isogeny volcano

2. *If* $p \equiv 3 \pmod 4$*, then from each vertex there are two horizontal* $\ell$*-isogenies for every prime* $\ell > 2$ *with* $(\frac{-p}{\ell}) = 1$*. There are two levels in the supersingular* 2*-isogeny graph.*

   (a) *If* $p \equiv 7 \pmod 8$*, then on each level there are* $h(-p)$ *vertices. The upper and lower levels are connected* $1:1$ *with* 2*-isogenies. On the upper level we also have two horizontal* 2*-isogenies from each vertex.*

   (b) *If* $p \equiv 3 \pmod 8$*, then we have* $h(-p)$ *vertices on the surface and* $3h(-p)$ *on the floor. Each vertex on the surface has three* 2*-isogenies to the lower level. There are no horizontal* 2*-isogenies.*

These graphs are illustrated in Figure 2.5 and Figure 2.6, for $p = 101$ and $\ell = 2$ and 3 respectively. Classes of elliptic curves are represented with the Montgomery $A$ coefficient of the curve. Note that a curve and its quadratic twist correspond to two different vertices since they are isomorphic over $\mathbb{F}_{p^2}$ but not over $\mathbb{F}_p$. For this reason, the twist of a curve written in black is written in grey. Since $p \equiv 1 \pmod 4$, there are $h(-4p) = 14$ supersingular elliptic curves defined over $\mathbb{F}_{101}$ up to $\mathbb{F}_p$-isomorphisms.

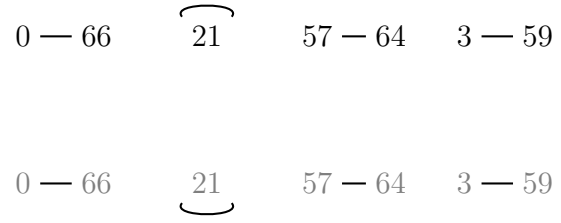



Figure 2.5: $\Gamma(\mathbb{F}_{101}, 2)$ For the 2-isogeny graph, because each vertex has one 2-neighbour, we have several connected components, each a special type of crater that is reduced to two points. Note that there is a 2-endomorphism defined over $\mathbb{F}_p$ for the curve defined by $y^2 = x^3 + 21x^2 + x$.

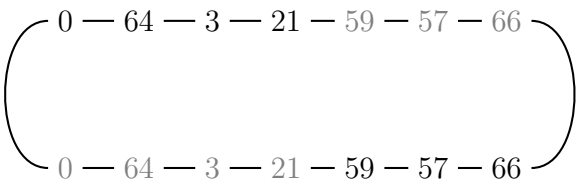



Figure 2.6: $\Gamma(\mathbb{F}_{101}, 3)$ In the 3-isogeny graph, each vertex has two neighbours. Every curve is in the same connected component, which is a large crater, because the ideal class group of $\mathbb{Q}(\sqrt{101})$ is cyclic: it is generated by a prime ideal with norm 3.

### 2.7.3 Supersingular over $\mathbb{F}_{p^2}$

In the supersingular case over $\mathbb{F}_{p^2}$ the $\ell$-isogeny graph for a prime $\ell$ is an $\ell + 1$ regular expander graph, meaning that it has good mixing properties: a sufficiently long random walk on the graph has the same probability to end up on any point of the graph. Even better, it is a Ramanujan graph

**Definition 26** (Ramanujan graph)**.** A Ramanujan graph $G$ is a regular graph of degree $k$ such that the eigenvalues $\lambda$ not equal to $\pm k$ of the adjacency matrix satisfy the bound $|\lambda| \leq 2\sqrt{k-1}$.

See e.g. [LPS88] for a survey on Ramanujan graphs.

**Proposition 23.** *Let $G$ be a Ramanujan graph. Let $S$ be any subset of the vertices of $G$, and $x$ be any vertex in $G$. Then a random walk of length at least*

$$\frac{|S|^{-1/2} \log(2|G|)}{\log \frac{k}{c}}$$

*starting from $x$ will land in $S$ with probability at least $\frac{|S|}{2|G|}$.*

*Proof.* See [LPS88]. □

**Theorem 24** ([Piz98])**.** *$\Gamma(\mathbb{F}_{p^2}, \ell)$ is a connected $k = \ell + 1$-regular multigraph satisfying the Ramanujan bound of $|\lambda| \leq 2\sqrt{\ell} = 2\sqrt{k-1}$ for the non-trivial eigenvalues of its adjacency matrix.*

These graphs are illustrated in Figure 2.7 and Figure 2.8, for $p = 101$ and $\ell = 2$ and 3 respectively.

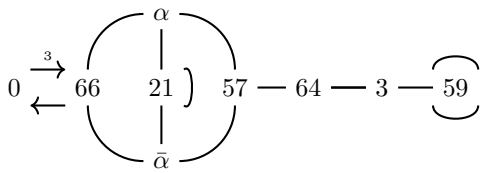



Figure 2.7: $\Gamma(\mathbb{F}_{101^2}, 2)$. Classes of elliptic curves are represented with the Montgomery $A$ coefficient of the curve. Here, $\alpha = 37 + t$ and $\bar{\alpha} = 37 - t$ are conjugate and defined over $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$ with $t^2 = 2$. There are 9 supersingular elliptic curves defined over $\mathbb{F}_{101^2}$ up to isomorphism. The 2-isogeny graph is 3-regular with edges counted with multiplicity.

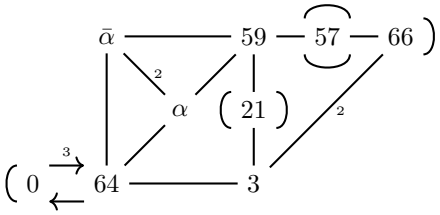



Figure 2.8: $\Gamma(\mathbb{F}_{101^2}, 3)$. Isomorphism classes of elliptic curves are represented with the Montgomery $A$ coefficient of the curve, except for the two classes of curves labelled with $\alpha = 37 + t$ and $\bar{\alpha} = 37 - t$ that are conjugated and defined over $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$ with $t^2 = 2$. There are 9 supersingular elliptic curves defined over $\mathbb{F}_{101^2}$ up to isomorphism. The 3-isogeny graph is 4-regular with edges counted with multiplicity.

# Chapter 3

# Isogeny-based key exchange protocols

In this chapter we describe three isogeny-based key exchange protocols, following the chronological order of their discovery. We start by presenting the isogeny-based key exchange protocol in the ordinary case from Couveignes and Rostovstev–Stolbunov (CRS). We then present SIDH (Supersingular Isogeny Diffie–Hellman) discovered by Jao and De Feo, on which is based SIKE, the NIST candidate key exchange protocol. Finally we describe CSIDH (Commutative Supersingular Diffie–Hellman) that uses supersingular elliptic curves defined over $\mathbb{F}_p$, developed by Castryck, Lange, Martindale, Panny and Renes. This protocol is based on the commutative action of the ideal class group on the set of elliptic curves. We compare the strength and weaknesses of these three scheme.

## 3.1 Ordinary case (CRS)

We start by describing the underlying security problems in the ordinary case. We then present the parameters and the key exchange protocol from Couveignes [Cou06], and Rostovstev–Stolbunov [RS06], before presenting the computational improvements proposed by De Feo, Kieffer and Smith in [DKS18]. We eventually discuss the feasibility of the scheme. In the following, we refer to the key exchange over ordinary curves as the CRS protocol, unless we want to highlight which version is considered, in which case we use full names.

Recall Theorem 18 from Chapter 2: If $\mathrm{Ell}_q(\mathcal{O})$ is the set of isomorphism classes over $\mathbb{F}_q$ of ordinary curves with $\mathcal{O} \cong \mathrm{End}(E)$ a maximal order, then $\mathrm{Cl}(\mathcal{O})$ acts freely and transitively on $\mathrm{Ell}_q(\mathcal{O})$. This free and transitive action is at the heart of the CRS protocol.

### 3.1.1 Security of the scheme and parameter sizes

The set $\mathrm{Ell}_q(\mathcal{O})$ of isomorphism classes over $\mathbb{F}_q = \mathbb{F}_{p^r}$ of ordinary curves with $\mathcal{O} \subset \mathrm{End}(E)$ is a conjectural hard homogeneous space for $\mathrm{Cl}(\mathcal{O})$. The security hence relies on the vectorization and parallelization problems (see Definition 3). Graphically speaking, the security of the scheme relies on the difficulty of finding a path between two given elliptic curves in an ordinary isogeny graph.

**Classical attack** The best classical attack known on vectorization is to use random walks on the graph of isogeny as in [DG16]), which gives a solution after an expected $O(p^{\frac{1}{4}})$ isogeny steps.

**Quantum attack** Since vectorization is an instance of the Abelian Hidden Shift Problem, the best quantum attack is Kuperberg's algorithm [Kup05, Reg04, Kup13] using the Childs–Jao–Soukharev quantum isogeny-evaluation algorithm as a subroutine [CJS14]. The result is a subexponential algorithm running in time $L_N[1/2, \sqrt{2}]$, with $N$ the cardinality of the ideal class group. There is some debate as to the concrete cost of this quantum algorithm, and the size of the ideal class group required to provide a cryptographically hard problem instance for common security levels [BLMP19, BS20, Pei20].

### 3.1.2 Couveignes key exchange protocol

**Public parameters** The protocol requires: a prime $p$; an order $\mathcal{O}$ in a quadratic field; and an initial ordinary elliptic curve $E_0$ defined over $\mathbb{F}_q$, where $q = p^r$, such that $\mathcal{O} \subset \mathrm{End}(E_0)$. The structure of the ideal class group $\mathrm{Cl}(\mathcal{O})$ and the lattice of relations between the ideals are necessary to compute the action of a randomly-sampled fractional ideal: without that structure, there is no way to convert a random ideal to an equivalent product of small-norm ideals. This lattice of relations as well as the class number can be computed using the Hafner–McCurley algorithm [HM89] as noted in [Cou06].

**Key generation** Alice randomly samples her private key $\mathfrak{a} \in \mathrm{Cl}(\mathcal{O})$ and computes her public key $E_A = E_0/\mathfrak{a}$. Bob proceeds similarly: he samples his private key $\mathfrak{b} \in \mathrm{Cl}(\mathcal{O})$, and computes his public key $E_B = E_0/\mathfrak{b}$.

**Key exchange** Upon receiving Bob's public key, Alice computes $E_B/\mathfrak{a}$. Bob computes $E_A/\mathfrak{b}$. From the commutativity of the group action, these two curves are isomorphic. The shared secret is their $j$-invariant.

This protocol is summarized in Figure 3.1.

We also give a graphical approach to the protocol: Alice chooses a secret walk on the isogeny graphs, starting from curve $E_0$. She arrives on a curve $E_A$ that will be her public key. Bob does the same and arrives on a curve $E_B$ which is his

**Public parameters:** $q = p^r$, an ordinary curve $E_0$ such that $\mathcal{O} \subset \mathrm{End}(E_0)$, where $\mathcal{O}$ is an order of an imaginary quadratic field.
The structure of $\mathrm{Cl}(\mathcal{O})$ and the lattice of relations between the ideals.

| $\mathcal{A}$lice | | $\mathcal{B}$ob |
|---|---|---|
| **Private key:** $\mathfrak{a} \in \mathrm{Cl}(\mathcal{O})$ | | **Private key:** $\mathfrak{b} \in \mathrm{Cl}(\mathcal{O})$ |
| **Public key computation:** Compute a smooth ideal $\mathfrak{a}'$ equivalent to $\mathfrak{a}$ using the group structure. | | **Public key computation:** Compute a smooth ideal $\mathfrak{b}'$ equivalent to $\mathfrak{b}$ using the group structure. |
| **Public key:** $E_A = E_0/\mathfrak{a}'$ | $\xrightarrow{E_A}$ | **Public key:** $E_B = E_0/\mathfrak{b}'$ |
| | $\xleftarrow{E_B}$ | |
| **Shared secret computation:** $E_{AB} = E_B/\mathfrak{a}'$ | | **Shared secret computation:** $E_{BA} = E_A/\mathfrak{b}'$ |
| **Shared secret:** $j(E_{AB})$ | | **Shared secret:** $j(E_{BA})$ |

Figure 3.1: Couveignes key exchange protocol.

public key. After this key generation step, Alice and Bob are ready to compute a shared secret. Alice simply reproduces her secret walk, but starting from Bob's curve $E_B$, and arrives on a curve $E_{AB}$. Bob does the same, reproducing his secret walk starting at Alice's curve, and arrives on a curve $E_{BA}$. Thanks to the commutativity property of the ideal class group action, the curves $E_{AB}$ and $E_{BA}$ are isomorphic and thus share the same $j$-invariant. This $j$-invariant is precisely the shared secret of Alice and Bob.

### 3.1.3 Rostovstev–Stolbunov key exchange protocol

**Public parameters** The protocol requires a prime $p$; an order $\mathcal{O}$ in a quadratic field; an initial ordinary elliptic curve $E_0$ defined over $\mathbb{F}_q$, where $q = p^r$, such that $\mathcal{O} \subset \mathrm{End}\, E_0$; a set of primes $\ell_i$ such that the Kronecker symbol $\left(\frac{t^2-4p}{\ell_i}\right) = 1$, where $t$ is the trace of Frobenius; a fractional ideal $\mathfrak{l}_i$ above each $\ell_i$; a set of possible integer exponents $\mathcal{S}$.

**Key generation** Alice samples a private exponent vector $(e_i)_{1 \le i \le n} \in \mathcal{S}$, sets $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i} \in \mathrm{Cl}(\mathcal{O})$ and computes her public key $E_0/\mathfrak{a}$. Bob pro-

ceeds similarly, he samples his private exponent vector $(e'_i)_{1\leq i\leq n} \in \mathcal{S}$, sets $\mathfrak{b} = \prod_{i=1}^{n} \mathfrak{l}_i^{e'_i} \in \mathrm{Cl}(\mathcal{O})$ and computes his public key $E_0/\mathfrak{b}$.

**Key exchange** Upon receiving Bob's public key, Alice computes $E_B/\mathfrak{a}$. Bob computes $E_A/\mathfrak{b}$. From the commutativity of the group action, these two curves are isomorphic. The shared secret is their $j$-invariant.

This protocol is summarized in Figure 3.2.

**Public parameters:** $q = p^r$, an ordinary curve $E_0$ such that $\mathcal{O} \subset \mathrm{End}(E_0)$, where $\mathcal{O}$ is an order of an imaginary quadratic field;
primes $\ell_i$ such that $\left(\frac{t^2-4p}{l_i}\right) = 1$;
a fractional ideal $\mathfrak{l}_i$ above each $\ell_i$.

| $\mathcal{A}$lice | $\mathcal{B}$ob |
|---|---|
| **Private key:** $(e_i)_{1\leq i\leq n} \in \mathcal{S}$ | **Private key:** $(e'_i)_{1\leq i\leq n} \in \mathcal{S}$ |
| **Public key:** $E_A = E/\mathfrak{a}$ such that $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ | **Public key:** $E_B = E/\mathfrak{b}$ such that $\mathfrak{b} = \prod_{i=1}^{n} \mathfrak{l}_i^{e'_i}$ |
| | $E_A \rightleftharpoons E_B$ |
| **Shared secret computation:** $E_{AB} = E_B/\mathfrak{a}$ | **Shared secret computation:** $E_{BA} = E_A/\mathfrak{b}$ |
| **Shared secret:** $j(E_{AB})$ | **Shared secret:** $j(E_{BA})$ |

Figure 3.2: Rostovstev–Stolbunov key exchange protocol.

**Differences between the Couveignes and Rostovstev–Stolbunov protocols** Using a set of given prime ideals and a set of exponents avoids the problem of needing to compute the class group structure and the $p$-smooth equivalent ideals in Couveignes' version. The work of [DKS18] and later [CLM+18] are built on the Rostovstev–Stolbunov protocol rather than on Couveignes version. However this convenience comes with a drawback: we are probably not working with the entire Hard Homogeneous Space, and the key sampling may not be uniform.

### 3.1.4 Computation

The CRS protocols are elegant but also very slow, since several minutes are needed to compute the key exchange on a laptop [DKS18]. This is due to the fact that computing the action of the ideal class group using the algorithms of Couveignes and Rostovstev–Stolbunov involves computing roots of modular polynomials over field extensions.

To address this drawback, De Feo, Kieffer and Smith used the underlying mathematical structure to accelerate the computational time. They look for a starting curve having the special property that the number of points defined over $\mathbb{F}_p$, i.e. $\#E(\mathbb{F}_p)$, is divisible by as many small $\ell_i$ as possible. This ensures that the ideal class group action is much faster to compute for these $\ell_i$. Indeed, it implies that the points defining the kernel all lie in $\mathbb{F}_p$, and are exactly the $\ell_i$-torsion subgroups. This allows us to compute the action using only Vélu's formula, and completely avoids modular polynomials and field extensions [DKS18].

However, it turns out to be difficult to find an ordinary curve with many such $\ell_i$ simultaneously, and De Feo, Kieffer and Smith only managed to apply this acceleration for seven primes after an extensive search for starting curves. However their improvement already accelerates the key exchange protocol by a factor of 4. This idea is reused in the supersingular case over $\mathbb{F}_p$ that will be described below, and hence opens a door to new efficient cryptographic protocols, and primitives.

**Timings** For 128-bits of classical security, the proof-of-concept algorithm in [DKS18] needs 520 seconds for a key generation. Although that is a factor 4 faster than the original CRS algorithm, and not optimized on the field arithmetic level, this timing keeps the CRS key exchange in the impracticable protocols league.

## 3.2 Supersingular case over $\mathbb{F}_{p^2}$ (SIDH and SIKE)

First published in 2011 by Jao and De Feo [JD11], Supersingular Isogeny Diffie–Hellman (SIDH) is the building block for Supersingular Isogeny Key Encapsulation (SIKE) [JAC+17]. The protocol SIKE is one of the round-3 alternate candidates for the NIST post-quantum contest. Both protocols use isogenies between supersingular elliptic curves over a finite field $\mathbb{F}_{p^2}$, which gives a faster scheme than CRS with more resistance to a quantum computer.

### 3.2.1 Commutative diagram

The endomorphism rings of supersingular curves over $\mathbb{F}_{p^2}$ are orders in a (non-commutative) quaternion algebra. Although it is possible to define a free and transitive group action on some specific subsets of the supersingular isogeny classes (see Section 3.3 and Chapter 6), there is no known commutative action

having these properties on the full set of supersingular elliptic curves. In order to circumvent the lack of a commutative action, Jao and De Feo proposed in [JD11] to use the commutativity of quotient isogenies, as in Figure 3.3.

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_A} & E_A = E_0/\langle G_A \rangle \\ \Big\downarrow{\scriptstyle \phi_B} & & \Big\downarrow{\scriptstyle \phi_{AB}} \\ E_B = E_0/\langle G_B \rangle & \xrightarrow{\phi_{BA}} & \begin{array}{c} E_{AB} \\ \simeq E_B/\langle \phi_B(G_A) \rangle \\ \simeq E_A/\langle \phi_A(G_B) \rangle \end{array} \end{array}$$

Figure 3.3: Commutative diagram for supersingular curves, with $E_0$ an elliptic curve and $\langle G_A \rangle$, $\langle G_B \rangle$ two subgroups of $E_0$.

This commutative diagram allows to have a key exchange protocol *à la* Diffie–Hellman, while the lack of a commutative group action protects the scheme against the subexponential attack of Childs, Jao and Soukharev [CJS14], offering a more quantum-resistant key exchange protocol than the ordinary case.

### 3.2.2 SIDH key exchange protocol

The commutative diagram in Figure 3.3 is at the heart of the key exchange protocol. Alice computes the horizontal arrows of the commutative diagram in Figure 3.3, whereas Bob computes the vertical ones.

**Public parameters** Let $p$ be a prime number such that $p = f \cdot \ell_A^{e_A} \ell_B^{e_B} \pm 1$, where $\ell_A$ and $\ell_B$ are primes, and $f$ is a cofactor making $p$ prime. Let $E$ be a supersingular elliptic curve such that $\#E(\mathbb{F}_{p^2}) = (p \mp 1)^2 = (f \cdot \ell_A^{e_A} \ell_B^{e_B})^2$. Let $(P_A, Q_A)$ and $(P_B, Q_B)$ be bases of $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$, the subgroups of $\ell_A^{e_A}$ and $\ell_B^{e_B}$-torsion respectively.

**Key generation** Alice chooses two secret integers $m_A$ and $n_A$, computes the point $G_A = [m_A]P_A + [n_A]Q_A$, the separable quotient isogeny $\phi_A$ of kernel $\langle G_A \rangle$, and the curve $E_A = E/\langle G_A \rangle$. She also computes the image of $P_B$ and $Q_B$ under her isogeny $\phi_A$. Her private key is the couple $(m_A, n_A)$, and her public key is $(E_A, \phi_A(P_B), \phi_A(Q_B))$.

Bob does the same using his own secret $(m_B, n_B)$ and the points of $\ell_B$-torsion $P_B$ and $Q_B$, instead of $P_A$ and $Q_A$, to get a public key $(E_B, \phi_B(P_A), \phi_B(Q_A))$.

**Shared secret** Alice receives Bob's public key and computes

$$E_{AB} = E_B/\langle \phi_B(G_A) \rangle = E_B/\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle \ .$$

Bob receives Alice's public key and computes

$$E_{BA} = E_A/\langle \phi_A(G_B) \rangle = E_A/\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle \ .$$

Thanks to the commutativity of quotient isogenies, these two curves are isomorphic. Their $j$-invariant is the shared secret. The protocol is summarized in Figure 3.4.

**Public parameters:** $p = f \cdot \ell_A^{e_A} \ell_B^{e_B} \pm 1$;
$E$ supersingular with cardinality $(p \mp 1)^2 = (f \cdot \ell_A^{e_A} \ell_B^{e_B})^2$;
$(P_A, Q_A)$ and $(P_B, Q_B)$ bases of $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$, respectively.

| $\mathcal{A}$lice | | $\mathcal{B}$ob |
|---|---|---|
| **Private key:** | | **Private key:** |
| $m_A, n_A \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ | | $m_B, n_B \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ |
| **Public key:** | | **Public key:** |
| $E_A = E/\langle [m_A]P_A + [n_A]Q_A \rangle$ | | $E_B = E/\langle [m_B]P_B + [n_B]Q_B \rangle$ |
| $\phi_A(P_B)$, $\phi_A(Q_B)$ | $\rightleftharpoons$ | $\phi_B(P_A)$, $\phi_B(Q_A)$ |
| **Shared secret computation:** | | **Shared secret computation:** |
| $G_{AB} =$ | | $G_{BA} =$ |
| $[m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A)$ | | $[m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B)$ |
| $E_{AB} = E_B/\langle G_{AB} \rangle$ | | $E_{BA} = E_A/\langle G_{BA} \rangle$ |
| **Shared secret:** | | **Shared secret:** |
| $j(E_{AB})$ | | $j(E_{BA})$ |

Figure 3.4: SIDH key exchange.

We also give an intuitive graphical explanation of the protocol. Alice chooses a walk on the $\ell_A$-isogeny graph to a curve $E_A$ that will be her public key. Bob does the same on the $\ell_B$-isogeny graph to a curve $E_B$. To compute the shared secret, Alice and Bob will switch places and apply their secret walk again, on the $\ell_A$ and $\ell_B$-isogeny graphs respectively. They will arrive on two curves that are isomorphic over $\mathbb{F}_{p^2}$, and that consequently share the same $j$-invariant. This $j$-invariant is their shared secret.

The absence of short cycles[1] in this isogeny graph (see [OAT20]) implies that the subgraphs reached by these isogenies look like regular trees. The shared secret is uniformly distributed on the set of curves with a cyclic $\ell_A^{e_A}\ell_B^{e_B}$-isogeny from $E_0$.

### 3.2.3 Underlying security problems

Let $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ be subgroups of $\ell_A^{e_A}$ and $\ell_B^{e_B}$ torsion respectively. Let $G_A$ and $G_B$ be the generators of subgroups in $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ respectively. In

[1] at least shorter than the number of steps in SIKE$p$434 and SIKE$p$503

the case of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, the image of $\phi_A(\langle G_B \rangle)$ and $\phi_B(\langle G_A \rangle)$ are needed to complete the commutative diagram. This constraint leads to analogues of the vectorization and parallelization problems.

**Definition 27** (Computational Supersingular Isogeny (CSSI) problem [JD11])**.** We keep the same notations as above. Additionally let $(P_B, Q_B)$ be a basis of $E_0[\ell_B^{e_B}]$ and let $\phi_A : E_0 \to E_A$ be an isogeny whose kernel is equal to $\langle G_A \rangle$. Given $E_A$, $\phi_A(P_B)$, and $\phi_A(Q_B)$, the *Computational Supersingular Isogeny problem* is to find $\phi_A$.

**Definition 28** (Supersingular Computational Diffie–Hellman (SSCDH) problem [JD11])**.** We keep the same notations as above. Additionally let $m_A$, $n_A$ (respectively $m_B$, $n_B$) be chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and not both divisible by $\ell_A$ (respectively $\ell_B$). Let $\phi_A : E_0 \to E_A$ be an isogeny whose kernel is equal to $\langle [m_A]P_A + [n_A]Q_A \rangle$, and let $\phi_B : E_0 \to E_B$ be an isogeny whose kernel is $\langle [m_B]P_B + [n_B]Q_B \rangle$. Given the curves $E_A$, $E_B$ and the points $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$, the *Supersingular Computational Diffie–Hellman problem* is to find the $j$-invariant of $E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle$.

Both problems are considered to be weaker instances of the isogeny path problem due to the additional information contained in the torsion points images revealed. However when $\ell_A^{e_A}$ and $\ell_B^{e_B}$ are balanced, we do not currently know of any attacks that exploit this information.

**Classical attack** The best classical attacks is the Van–Oorschot and Wiener collision finding algorithm, as shown in [ACC+18]. It runs in time $\mathcal{O}(\ell_A^{e_A}) = \mathcal{O}(p^{\frac{1}{4}})$ (when $\ell_A^{e_A}$ and $\ell_B^{e_B}$ are balanced, i.e. when $\ell_A^{e_A} \simeq \ell_B^{e_B}$).

**Quantum attack** As there is no commutative action involved, the Kuperberg algorithm used in the ordinary case does not apply to SIDH. The best quantum attack against vectorization uses Tani's algorithm [Tan20]. It is exponential, and runs in $\mathcal{O}(p^{\frac{1}{6}})$. However the analysis of Adj, Cervantes-Vázquez, Chi-Domínguez, Menezes, and Rodríguez-Henríquez [ACC+18] shows that this exponential attack requires a huge amount of memory, and that the classical van Oorschot and Wiener attack might turn out to be more efficient in practice.

**Timings** Isogeny-based cryptography benefits from years of research and optimization on elliptic curves protocol, which allows us to reach an acceptable running time for a widespread use. Improvements have been made on formulae for isogeny computation and on efficient arithmetic by Costello, Longa and Naehrig in [CLN16] and Costello and Hisil in [CH17]. For 128-bits of classical security, the actual SIKE implementation runs in 5.9 ms for the encapsulation and decapsulation as claimed in [JAC+17], Table 2.1 (running on a 3.4GHz Intel Core i7–6700 (Skylake) processor with the use of hand-tuned x64 assembly).

### 3.2.4 From SIDH to SIKE

In its call for quantum protocols, NIST required key encapsulation mechanisms (KEM). Key encapsulation uses asymmetric encryption to transmit a ciphertext, from which a symmetric key is derived on both sides, later used for message encryption. In the case of SIDH, going from a Diffie–Hellman protocol to a key encapsulation protocol can be done following two steps:

1. The first step derives a public key encryption (PKE) protocol from SIDH, by XORing the hash of the shared secret obtained from SIDH with the message to be encrypted.

2. The second step uses the Hofheinz, Hövelmanns and Kiltz transform [HHK17] (a derivative of the Fujisaki–Okamoto transform [FO13]) to create a key encapsulation mechanism from the PKE protocol. It uses long-term asymmetric keys for authentification, and ephemeral asymmetric keys to encrypt an ephemeral symmetric key.

## 3.3 Supersingular case over $\mathbb{F}_p$ (CSIDH)

The action of $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$ on the set of supersingular elliptic curves defined over $\mathbb{F}_p$ described in Section 2.6 can be used for key exchange and encapsulation [CLM+18], signatures [DG18, DPV19, BKV19], and other more advanced protocols. We focus on the key exchange protocol, CSIDH, in the following.

Compared to SIDH [JD11, DJP14], CSIDH is slower. On the positive side, CSIDH has smaller public keys (although it depends on security estimates for parameters), is based on a better-understood security assumption, and supports an easy key validation procedure, making it better-suited than SIDH for static key exchange.

### 3.3.1 The ideal class group action

Recall Theorem 19 from Section 2.6. Let $\mathcal{S}_p$ be the set of supersingular elliptic curves over $\mathbb{F}_p$. Let $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$. The ideal class group $\mathrm{Cl}(\mathcal{O})$ acts freely and transitively on $\mathcal{S}_p$.

Let $E$ be a supersingular elliptic curve defined over $\mathbb{F}_p$ with $\mathrm{End}(E) \cong \mathbb{Z}[\sqrt{-p}]$. For CSIDH, we are interested in computing the action of small prime ideals. Consider one of the primes $\ell_i$ dividing $p+1$; the principal ideal $(\ell_i) \subset \mathbb{Z}[\sqrt{-p}]$ splits into two primes, namely $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$ and $\bar{\mathfrak{l}}_i = (\ell_i, \pi_p + 1)$, where $\pi_p$ is the element of $\mathbb{Z}[\sqrt{-p}]$ mapping to the Frobenius endomorphism of the curves. Since $\bar{\mathfrak{l}}_i \mathfrak{l}_i = (\ell_i)$ is principal, we have $\bar{\mathfrak{l}}_i = \mathfrak{l}_i^{-1}$ in $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$, and hence

$$\bar{\mathfrak{l}}_i \cdot (\mathfrak{l}_i \cdot E) = \mathfrak{l}_i \cdot (\bar{\mathfrak{l}}_i \cdot E) = E \,.$$

**A graphical toy example** Figure 3.5 represents the graph of supersingular isogenies over $\mathbb{F}_p$ for $p = 59 = 4 \cdot 3 \cdot 5 - 1$. The circular black graph is the

graph of 3-isogenies obtained by applying ideals in the class of $[(3, \pi_p - 1)]$ that correspond to 3-isogenies having their kernel in $E(\mathbb{F}_p)$. The graph in blue is the graph of 5-isogenies obtained by applying ideals in the class of $[(5, \pi_p - 1)]$.

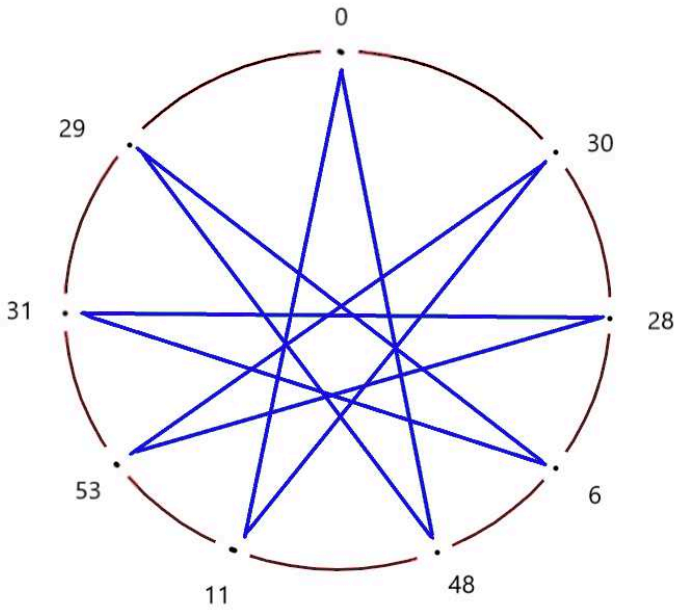



Figure 3.5: The supersingular 3-isogeny (the black circle) and 5-isogeny (the blue star) graph over $\mathbb{F}_p$ for $p = 59$. Vertices are labelled by the $A$ coefficient of the Montgomery representation of the curves.

### 3.3.2 CSIDH key exchange protocol

**Public parameters** The protocol requires a prime $p = 4 \cdot \ell_1 \cdots \ell_n \cdot f - 1$, where $\ell_i$ are primes and $f$ is a cofactor, and an exponent space $\mathcal{S} \in \mathbb{Z}^n$. We then choose an initial supersingular curve $E$ defined over $\mathbb{F}_p$ with cardinality $E(\mathbb{F}_p) = (p + 1)$. We write $\mathfrak{l}_i = [(\ell_i, \pi_p - 1)]$.

**Key generation** Alice samples a private exponent vector $(e_i)_{1 \leq i \leq n} \in \mathcal{S}$, sets $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i} \in \mathrm{Cl}(\mathbb{Z}(\sqrt{-p}))$ and computes her public key $E_0/\mathfrak{a}$ as a sequence of $e_i$ actions by each $\mathfrak{l}_i$. Bob proceeds similarly, he samples his private exponent vector $(e'_i)_{1 \leq i \leq n} \in \mathcal{S}$, sets $\mathfrak{b} = \prod_{i=1}^{n} \mathfrak{l}_i^{e'_i} \in \mathrm{Cl}(\mathbb{Z}(\sqrt{-p}))$ and computes his public key $E_0/\mathfrak{b}$.

**Key exchange** Upon receiving Bob's public key, Alice computes $E_B/\mathfrak{a}$. Bob computes $E_A/\mathfrak{b}$. From the commutativity of the group action, these two curves

are isomorphic. The shared secret is their $j$-invariant.

This protocol is summarized in Figure 3.6. Graphically speaking, we navigate on the $\ell_i$-isogeny graphs using the action of the ideal class group on curves exactly as in the ordinary case.

**Public parameters:** A prime $p = 4 \cdot \ell_1 ... \ell_n \cdot f - 1$;
A list of prime ideals $\mathfrak{l}_i$ above $\ell_i$ for each $i$;
A supersingular curve $E$ defined over $\mathbb{F}_p$ such that $\mathrm{End}(E) \simeq \mathcal{O}$,
where $\mathcal{O}$ is an imaginary quadratic order;
An exponent space $\mathcal{S}$ in $\mathbb{Z}^n$

| $\mathcal{A}$lice | $\mathcal{B}$ob |
|---|---|
| **Private key:** $(e_i)_{1 \leq i \leq n} \in \mathcal{S}$ | **Private key:** $(e'_i)_{1 \leq i \leq n} \in \mathcal{S}$ |
| **Public key:** $E_A = E/\mathfrak{a}$ such that $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$ | **Public key:** $E_B = E/\mathfrak{b}$ such that $\mathfrak{b} = \prod_{i=1}^{n} \mathfrak{l}_i^{e'_i}$ |
| $\xrightarrow{E_A}$ | |
| | $\xleftarrow{E_B}$ |
| **Shared secret computation:** $E_{AB} = E_B/\mathfrak{a}$ | **Shared secret computation:** $E_{BA} = E_A/\mathfrak{b}$ |
| **Shared secret:** $j(E_{AB})$ | **Shared secret:** $j(E_{BA})$ |

Figure 3.6: CSIDH key exchange.

### 3.3.3 Security of the scheme

The set $\mathcal{S}_p$ of supersingular elliptic curves over $\mathbb{F}_p$ is a conjectural hard homogeneous space for $\mathrm{Cl}(\mathbb{Z}(\sqrt{-p}))$, assuming that finding isogenies between supersingular curves is hard. The security of CSIDH hence relies on the vectorization and parallelization problems (see Definition 3). Graphically speaking, the security of the scheme relies on the difficulty of finding a path between two given elliptic curves in a supersingular isogeny graph over $\mathbb{F}_p$. For cryptographic purposes, the exponent vectors $(e_1, \ldots, e_n)$ must be taken from a space of size at least $2^{2\lambda}$, where $\lambda$ is the (classical) security parameter.

**Classical attack** The best classical attack known on vectorization is to use random walks on the isogeny graph as in [DG16], which gives a solution after

an expected $O(p^{\frac{1}{4}})$ isogeny steps.

**Quantum attack** As for CRS, since vectorization is an instance of the Abelian Hidden Shift Problem, the best quantum attack is Kuperberg's algorithm [Kup05, Reg04, Kup13] using the Childs–Jao–Soukharev quantum isogeny-evaluation algorithm as a subroutine [CJS14]. The result is a subexponential algorithm running in time $L_p[1/2, \sqrt{2}]$. There is some debate as to the concrete cost of this quantum algorithm, and the size of $p$ required to provide a cryptographically hard problem instance for common security levels (see [BLMP19, BS20, Pei20]).

### 3.3.4 Computation

CSIDH works over a finite field $\mathbb{F}_p$, where $p$ is a prime of the form

$$p := 4 \prod_{i=1}^{n} \ell_i - 1$$

with $\ell_1, \ldots, \ell_n$ a set of small odd primes. Concretely, the original CSIDH article [CLM+18] defined a 511-bit $p$ with $\ell_1, \ldots, \ell_{n-1}$ the first 73 odd primes, and $\ell_n = 587$.

The set of public keys in CSIDH is a subset of all supersingular elliptic curves defined over $\mathbb{F}_p$, in *Montgomery form* $y^2 = x^3 + Ax^2 + x$, where $A \in \mathbb{F}_p$ is called the *A-coefficient* of the curve. The endomorphism rings of these curves are isomorphic to orders in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$. These orders have to contain $\mathbb{Z}[\sqrt{-p}]$, because the Frobenius endomorphism is always in the endomorphism ring, which implies that there at most two possibilities: $\mathbb{Z}[\sqrt{-p}]$, and $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$ (if $p \equiv 3 \pmod 4$).

The authors of [CLM+18] choose to restrict the starting curve and public keys to curves with endomorphism rings isomorphic to $\mathbb{Z}[\sqrt{-p}]$. However, when $p \equiv 3 \pmod 4$, it is also possible to use curves with endomorphism ring isomorphic to $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$ as in CSURF [CD20].

At the heart of CSIDH is an algorithm that evaluates the class group action described in Section 3.3.1 on any supersingular curve over $\mathbb{F}_p$.

The input to the algorithm is an elliptic curve $E : y^2 = x^3 + Ax^2 + x$, represented by its $A$-coefficient, and an ideal class $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}$, represented by its list of exponents $(e_i, \ldots, e_n) \in \mathbb{Z}^n$. The output is the ($A$-coefficient of the) elliptic curve $\mathfrak{a} \cdot E = \mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n} \cdot E$.

The isogenies corresponding to $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$ can be efficiently computed using Vélu's formulæ and their generalizations: exploiting the fact that $\#E(\mathbb{F}_p) = p + 1 = 4 \prod \ell_i$, one looks for a point $R$ of order $\ell_i$ in $E(\mathbb{F}_p)$ (i.e., a point in $E[\ell_i]$, that is, in the kernel of both the multiplication-by-$\ell_i$ map and $(\pi_p - 1)$), computes the isogeny $\phi : E \to E/\langle R \rangle$ with kernel $\langle R \rangle$, and sets $\mathfrak{l}_i \cdot E = E/\langle R \rangle$. Iterating this procedure lets us compute $\mathfrak{l}_i^e \cdot E$ for any exponent $e \geq 0$.

The isogenies corresponding to $\mathfrak{l}_i^{-1}$ are computed in a similar fashion: this time one looks for a point $R$ of order $\ell_i$ in the kernel of $(\pi_p + 1)$, i.e., a point

in $E(\mathbb{F}_{p^2})$ of the form $(x, iy)$ where both $x$ and $y$ are in $\mathbb{F}_p$ (since $i = \sqrt{-1}$ is in $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$ and satisfies $i^p = -i$). Then one proceeds as before, setting $\mathfrak{l}_i^{-1} \cdot E = E/\langle R \rangle$.

In the sequel we assume that we are given an algorithm `QuotientIsogeny` which, given a curve $E/\mathbb{F}_p$ and a point $R$ in $E(\mathbb{F}_{p^2})$, computes the quotient isogeny $\varphi : E \to E' \cong E/\langle R \rangle$, and returns the pair $(\varphi, E')$. We refer to this operation as *isogeny computation*. Algorithm 1, taken from the original CSIDH article [CLM+18], computes the class group action.

**Algorithm 1:** `KeyGenCSIDH`: The original CSIDH class group action algorithm for supersingular curves over $\mathbb{F}_p$ where $p = 4\prod_{i=1}^{n} \ell_i - 1$. The choice of ideals $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$, where $\pi_p$ is the element of $\mathbb{Q}(\sqrt{-p})$ is mapped to the $p$-th power Frobenius endomorphism on each curve in the isogeny class, is a system parameter. This algorithm constructs exactly $|e_i|$ isogenies for each ideal $\mathfrak{l}_i$. In practice the $y$-coordinate of the points on the curve is not required, and the scalar multiplications can be done using Montgomery $x$-only arithmetic.

**Input:** $A \in \mathbb{F}_p$ such that $E_A \colon y^2 = x^3 + Ax^2 + x$ is supersingular, and an integer exponent vector $(e_1, \ldots, e_n)$

**Output:** $B$ such that $E_B : y^2 = x^3 + Bx^2 + x$ is $\mathfrak{l}_1^{e_1} \cdot \cdots \cdot \mathfrak{l}_n^{e_n} \cdot E_A$,

$B \leftarrow A$
**while** *some* $e_i \neq 0$ **do**
  Sample a random $x \in \mathbb{F}_p$
  $s \leftarrow +1$ if $x^3 + Bx^2 + x$ is square in $\mathbb{F}_p$, else $s \leftarrow -1$
  $S \leftarrow \{i \mid e_i \neq 0, \mathrm{sign}(e_i) = s\}$
  **if** $S \neq \emptyset$ **then**
    $k \leftarrow \prod_{i \in S} \ell_i$
    $Q \leftarrow [(p+1)/k]P$, where $P = (x, y)$ with $y^2 = x^3 + Bx^2 + 1$
    **for** $i \in S$ **do**
      $R \leftarrow [k/\ell_i]Q$ `// Point to be used as kernel generator`
      **if** $R \neq \infty$ **then**
        $(E_B, \varphi) \leftarrow$ `QuotientIsogeny`$(E_B, R)$
        $Q \leftarrow \varphi(Q)$
        $(k, e_i) \leftarrow (k/\ell_i, e_i - s)$
**return** $B$

**Timings** The current fastest constant-time implementation is the one from CTIDH [BBC+21]. It takes about 40 ms for key generation for 128 bits of classical security.

## 3.4 Key validation

During a key exchange protocol, an active attacker can transmit a flawed public key to gain knowledge on the counterpart's private key. To avoid such kinds of attacks, two solutions are possible: restricting to ephemeral key exchange protocols, or using static key exchanges with a key validation procedure.

On the one hand, ephemeral key exchange protocols require the private and public key to be unique to each key establishment to avoid adaptative attacks. This is the case in SIDH, where an attacker having a public key with dishonestly chosen torsion points may recover information about the counterpart's private key if it is reused for several key exchanges.

On the other hand, static key exchange protocols requires a key-validation procedure when receiving a public key to verify its correctness, i.e. that it has been honestly generated. The private and public keys can be reused for several key exchanges, under the condition that the public key must be validated by counterparts before using it. The public key is generated only once, but it is verified at each key exchange protocol. Concrete examples of validation will be given in Section 6.5.

## 3.5 Comparison of CRS, SIDH, SIKE and CSIDH

Table 3.1 gives a summary of the main differences between the isogeny-based key exchanges presented above. We also give a summary of the advantages and drawbacks for each scheme presented above.

**CRS** The CRS scheme launched isogeny-based post-quantum cryptography, and offers an elegant post-quantum Diffie–Hellman protocol. The improvements of [DKS18] are reused in CSIDH described below, but the implementations of Couveignes' and Rostovstev–Stolbunov protocols remain too slow for a practical and widespread use.

The existence of a subexponential quantum attack does not mean that the protocol is insecure. Nevertheless, such an attack implies using bigger primes, and thus bigger key sizes and an even slower protocol to reach an equivalent security level.

**SIDH and SIKE** Thanks to algorithmic improvements, the SIDH and SIKE protocols have reached an acceptable running time. SIKE has the shortest public key size of all the NIST candidates. Hence it has positioned itself as a promising post-quantum candidate for standardization.

However, some critiques have been made about the fact that the images of the torsion points need to be sent in order to be able to compute the commutative diagram without having Alice or Bob reveal their private key. This may make the problem an easier instance of the quantum-resistant isogeny-path problem. There have been a few attacks on the isogeny problem with torsion

point images for unbalanced parameters, i.e. when $\ell_A^{e_A}$ and $\ell_B^{e_B}$ have significantly different sizes ([Pet17], [dQKL+21]), but none of them apply to SIDH parameters. It is still unknown how to use this extra information properly in the SIDH context.

Furthermore, SIDH and SIKE also lack of efficient public key validation to verify that a public key has been honestly generated. Galbraith, Petit, Shani, and Ti have shown in [GPST16] that some public key validation algorithms, if they were efficient, would also give an attack on SIDH.

**CSIDH** CSIDH benefits from being several orders of magnitude faster than its ancestor CRS. It also allows shorter public and private keys than SIDH at the same security level: for 64 bits of quantum security, the first NIST security level, SIDH public keys have 330 bytes, whereas CSIDH-512 public keys could fit in 64 bytes. Unlike SIDH, it allows a secure non-interactive key exchange protocol as it has efficient public key validation.

However, CSIDH is vulnerable to subexponential quantum attacks, and its quantum security is the subject of intense discussion (see Section 3.3.3). Slower than SIDH, it is close to being practical, but remains an order of magnitude slower than other non-isogeny based key exchange protocols. Moreover, considering the on-going debates on the level of security offered by CSIDH, the parameters might need to be larger than what we are currently using to maintain the claimed security level.

| | CRS | CSIDH | SIDH/SIKE |
|---|---|---|---|
| Field | $\mathbb{F}_p$ or extensions | $\mathbb{F}_p$ | $\mathbb{F}_{p^2}$ |
| Curves | Ordinary | Supersingular | Supersingular |
| Prime | (Any) | $4\prod_{i=1}^{n}\ell_i \pm 1$ | $f\ell_1^{e_A}\ell_2^{e_B} \pm 1$ |
| $\ell$-isogeny graph | Volcano | Crater or depth-2-volcano | $\ell$-regular expander graph |
| Origin of commutativity | Ideal class group action | Ideal class group action | Commutative diagram |
| non-interactive key establishment | Safe | Safe | Unsafe |
| Best known classical attack (in isogeny steps) | Exponential $O(p^{\frac{1}{4}})$ | Exponential $O(p^{\frac{1}{4}})$ | Exponential $\mathcal{O}(p^{\frac{1}{4}})$ |
| Best known quantum attack (in quantum queries) | Subexponential $L_p[1/2, \sqrt{2}]$ | Subexponential $L_p[1/2, \sqrt{2}]$ | Exponential $\mathcal{O}(p^{\frac{1}{6}})^*$ or $\mathcal{O}(p^{\frac{1}{4}})$ |

Table 3.1: Comparative table of isogeny-based key exchange schemes. The complexity of the best known quantum attack for SIDH/SIKE is marked with an asterisk $*$ because, according to the analysis of [ACC+18], its memory requirements are too big to be met in practice, meaning that the best quantum attack would actually be the classical one.

# Part II

# CSIDH implementation

# Chapter 4

# Protecting CSIDH against side-channel attacks

**Abstract** Side-channel attacks monitor physical parameters during the computation of a cryptographic protocol. In particular, power consumption analysis, timing attacks, and fault injections can be used to recover the private key during a key exchange protocol. Some implementations of CSIDH have tackled the issue of protecting the scheme against timing and power consumption analysis [MCR19] [OAYT20]. However they use dummy operations to ensure that the computation can be run in constant-time, which makes the scheme vulnerable to fault injections. A dummy-free implementation of CSIDH is necessary to avoid fault injections attacks. In this chapter we propose two constant-time implementations of CSIDH which do not use dummy operations, including one without randomness.

The results of this section have been published in [CCC+19].

## 4.1 Preliminaries: side-channel attacks

Side channel attacks were introduced in 1996 by Paul Kocher [Koc96], at the time against contemporary RSA implementations. Today they are widely known and used against any cryptographic protocol.

The idea of side-channel attacks is to recover the private key by monitoring information about the execution of the protocol on a real platform, instead of attacking the underlying mathematical problem. These methods grant different powers to the attacker. In passive settings, side-channel attacks rely on monitoring several computational parameters such as timing, power consumption, sound, electromagnetic leaks, cache memory, or data remanence. In active settings, an attacker is allowed to induce a voluntary perturbation during execution, such as fault-injections or the use of a flawed random number generator. By observing the consequences of her perturbations she may deduce some information about the secret, or break the cryptosystem.

We detail four types of side-channel attacks that will be considered in the following sections: timing attacks, power consumption analysis, fault injection and flawed random number generators. We also introduce the notion of constant-time algorithms that are a countermeasure against such attacks.

### 4.1.1 Timing attacks

Timing attacks monitor the duration of the computation: some algorithms run faster on some inputs and significantly slower on others. When this input is a cryptographic secret, these variations leak information revealing part or all of it (see [Koc96], [BB03], and more recently attacks such as Lucky Thirteen [AP13], Meltdown [LSG+18] and Spectre [KHF+19]).

We give an example with the square-and-multiply algorithm 2 which is widely used. This algorithm computes exponentiation in a multiplicative group, e.g. in RSA. An analogue of this algorithm exists in the case of an additive group law, as in elliptic curves, where square-and-multiply becomes double-and-add. It relies on the following result: given an element $x$ of a group $\mathcal{G}$ written multiplicatively and an integer $n$:

$$x^n = \begin{cases} x(x^2)^{\frac{n-1}{2}} & \text{if } n \text{ is odd} \\ (x^2)^{\frac{n}{2}} & \text{if } n \text{ is even} \end{cases}$$

The square-and-multiply algorithm scans the bits of the exponent in base two from most significant to least significant. If the bit is zero, we square the previous result, but if it is one, we square the result *and* multiply by $x$. This is computationally more expensive and takes more time.

**Algorithm 2:** Square-and-multiply exponentiation

**Input:** $x$ an element of a multiplicative group $\mathcal{G}$,
$n \in \mathbb{N}, n = (n_{k-1}...n_0)_2$
**Output:** $x^n$

```
if n = 0 then
    return 1
y ← 1
for i ← (k − 1) to 0 do
    y ← y × y
    if n_i = 1 then
        y ← y × x
return y
```

Execution timings reveal the proportions of 0s and 1s in the exponent, although the exact places of the 0s and 1s is not revealed. When the exponent is a secret value (such as an RSA or Diffie–Hellman private key), this leaks information on the secret and creates security issues.

### 4.1.2 Power consumption analysis

Attacks by power consumption analysis monitor the power consumption variations of the computing device during the execution of an algorithm. These variations can reveal part or all of a private key.

We give an example of Simple Power Analysis by considering again the square-and-multiply exponentiation given in Algorithm 2. The power consumption during a square-and-multiply step is different and distinguishable from the consumption of a square step. When the exponent $n$ is the private key of one participant, the attacker could *read the entire key* from the power consumption graph, since the variations reveal each bit of the exponent one by one, as represented on Figure 4.1.

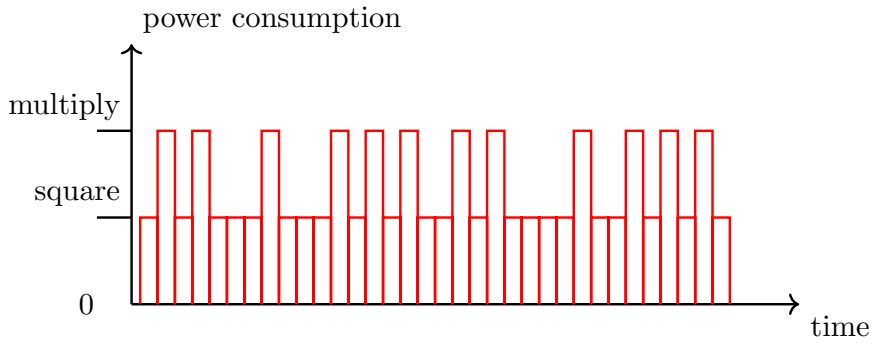



Figure 4.1: Power consumption analysis indicative example – square-and-multiply with exponent $(11001001111011000101110)_2$. For clarity we use the exaggerated ratio of 2 : 1 for the power consumed by a multiplication, relative to a square.

For details on more powerful power-analysis attacks, such as Differential Power Analysis, see [KJJ99].

### 4.1.3 Fault injection

Fault injection is a stronger attack model than timing attacks and power consumption analysis: the attacker is allowed to be active, and not simply an observer. The attacker can create one or several faults during the computation, by cutting electrical power, or using a laser to flip some bits for example. Comparing the output with and without the faults may reveal information about the private key. These attacks can be particularly useful when dummy operations have been added to an implementation to protect against timing and power consumption analysis attacks: if a fault is injected in a dummy operation, then it has no impact on the output.

Let us give an example with a modified version of square-and-multiply in Algorithm 3. In this version, timing and power consumption analysis attacks

have been prevented by adding fake and useless operations to the square-only step, to ensure that it has the same computational cost as the square-and-multiply step.

**Algorithm 3:** Modified square-and-multiply exponentiation with dummy operations

**Input:** $x$ an element of a multiplicative group $\mathcal{G}$,
$n \in \mathbb{N}, n = (n_{k-1}...n_0)_2$
**Output:** $x^n$

```
1  if n = 0 then
2      return 1
3  if n = 0 then
4      return 1
5  y ← 1
6  z ← x
7  for i ← (k − 1) to 0 do
8      if n_i = 0 then
9          y ← y × y
10         z ← y × x // dummy operation
11     else
12         y ← y × y
13         y ← y × x
14 return y
```

In the fault injection model, an attacker could inject a fault during Step $i$ (at Lines 10 and 13), and observe if the output is the same as without the fault injection. If the result is wrong, then the targeted operation was a real one and the $i^{th}$ bit was 1; if the result is correct, then the operation was a fake one and the $i^{th}$ bit was 0.

Of course, being able to insert faults at a precise moment requires more sophisticated material, and a higher cost for the attacker [TDEP21]. It is nonetheless feasible, even on everyday smart cards protected by RSA [BJL+14].

### 4.1.4 Constant-time and dummy-free algorithms

The implementation of an algorithm is said to be *constant-time* when the analysis of its execution time provides no information on secret inputs. Constant-time implementations hence ensure that no secret information visible via timing attacks leaks during the execution.[1]

Constant-time implementations can be achieved by using dummy operations, however such unnecessary steps create a vulnerability against fault injection. To

[1] Misleadingly, the execution time of a constant-time implementation does not need to be constant: it can vary due to randomness, or as a function of public inputs, but not from the secret inputs.

further protect an implementation against fault injection, every step has to be necessary to compute the correct result. A protected version of the square-and-multiply algorithm is given in Algorithm 5, where each conditional branch has a square step and a multiply step, both necessary for the completion of the algorithm [Mon87].

**Algorithm 4:** Montgomery square-and-multiply exponentiation

**Input:** $x$ an element of a multiplicative group $\mathcal{G}$,
$n \in \mathbb{N}, n = (n_{k-1}...n_0)_2$
**Output:** $x^n$

```
1 x_1 ← 1
2 x_2 ← x
3 for i = k − 1 to 0 do
4 |   if n_i = 0 then
5 |   |   (x_1, x_2) ← (x_1^2, x_1 × x_2)
6 |   else
7 |   |   (x_1, x_2) ← (x_1 × x_2, x_2^2
8 return x_1
```

Algorithm 4 has an if statement which should also be implemented in constant-time to ensure a full constant-time algorithm. To that aim we use a conditional constant-time swap between two values: $\texttt{cswap}(a, b, t)$ returns $(a, b)$ if $t = 0$ and $(b, a)$ otherwise. We obtain Algorithm 5.

**Algorithm 5:** Montgomery square-and-multiply exponentiation with conditional swap

**Input:** $x$ an element of a multiplicative group $\mathcal{G}$,
$n \in \mathbb{N}, n = (n_{k-1}...n_0)_2$
**Output:** $x^n$

```
1 x_1 ← 1
2 x_2 ← x
3 for i = k − 1 to 0 do
4 |   (x_1, x_2) ← cswap(x_1, x_2, n_i)
5 |   x_2 = x_1 × x_2
6 |   x_1 = x_1^2
7 |   (x_1, x_2) ← cswap(x_1, x_2, n_i)
8 return x_1
```

For completeness we also give in Algorithm 6 the additive version, double-and-add, of the square-and-multiply algorithm for when the group law is written additively and not multiplicatively (in elliptic curve scalar multiplication for example).

**Algorithm 6:** Montgomery double-and-add exponentiation

**Input:** $x$ an element of an additive group $\mathcal{G}$, $n \in \mathbb{N}, n = (n_{k-1}...n_0)_2$
**Output:** $n \times x$

1 $x_1 \leftarrow 0$
2 $x_2 \leftarrow x$
3 **for** $i = k-1$ *to* $0$ **do**
4  **for** $i = k-1$ *to* $0$ **do**
5  $(x_1, x_2) \leftarrow \texttt{cswap}(x_1, x_2, n_i)$
6  $x_2 = x_1 + x_2$
7  $x_1 = x_1 + x_1$
8  $(x_1, x_2) \leftarrow \texttt{cswap}(x_1, x_2, n_i)$
9 **return** $x_1$

## 4.2 Previous constant-time implementations

Prior to our study, several authors had been tackling the issue of timing and power consumption analysis in the case of CSIDH [MCR19][OAYT20]. We briefly recall their protocols before explaining why they are not fault injection resistant.

### 4.2.1 Meyer–Campos–Reith

As Meyer, Campos and Reith observe in [MCR19], the original CSIDH algorithm (Algorithm 1, Section 3.3.2) performs fewer scalar multiplications when the key has the same number of positive and negative exponents (balanced case) than it does when the exponents are all positive or all negative (unbalanced case). Indeed, when the key is balanced, the multiplication at Line 8 in Algorithm 1 has a cofactor of $\log p/2$ bits, meaning that the following multiplications at Line 10 have cofactors of decreasing size from $\log p/2$. However, when the private key has only positive or only negative exponents, then the multiplication at Line 8 has a cofactor of $\log p$, bits, hence the following multiplications at Line 10 have cofactors of decreasing size from $\log p$ only. Adding the bit length of multiplicative factors for the computation of one isogeny per degree in CSIDH-512, [MCR19] finds 9066 bits in the balanced case and 16813 bits in the unbalanced case. As the cost of a point multiplication depends on the size of the cofactor, Algorithm 1 leaks information about the distribution of positive and negative exponents under timing attacks.

The authors of [MCR19] also study power consumption attacks. They assume that by studying the variations of power consumption, an attacker can distinguish between a isogeny computation (Line 12), and a point multiplication (Line 10) in Algorithm 1. This allows the attacker to compare the degree of the isogenies computed. They further assume that an attacker can distinguish between the while-loops at Line 2 in Algorithm 1 in the computation, which allows the attacker to identify when a new point is sampled. Recall that in

CSIDH, isogenies whose corresponding private exponents share the same sign are computed together. This implies that the power consumption analysis described above can compare the degree of the isogeny computed and observe which batches of isogenies are computed together throughout the execution. Since only isogenies having exponents of same sign exponent can be computed together, this reveals information about the signs. Hence the possible key space is reduced, and the complexity of finding the correct key is reduced as well.

In view of this vulnerability, Meyer, Campos and Reith proposed a constant-time CSIDH algorithm in [MCR19] whose running time does not depend on the private key (though, unlike [JAMJ19], it still varies due to randomness). The essential differences between the algorithm of [MCR19] and classic CSIDH are as follows. First, to address the vulnerability to timing attacks, they choose to use only positive exponents in $[0, 10]$ for each $\ell_i$, instead of $[-5, 5]$ in the original version [CLM+18], while keeping the same prime $p = \prod_{i=1}^{74} \ell_i - 1$. To mitigate power consumption analysis attacks, their algorithm always computes the maximal amount of isogenies allowed by the exponent bound, using dummy isogeny computations if needed. Their algorithm is described in Algorithm 7.

### 4.2.2 Onuki–Aikawa–Yamazaki–Takagi

Still assuming that the attacker can perform only power consumption analysis and timing attacks, Onuki, Aikawa, Yamazaki and Takagi proposed a faster constant-time version of CSIDH in [OAYT20]. The key idea is to use two points to evaluate the action of an ideal, one in $\ker(\pi_p - 1)$ (i.e., in $E(\mathbb{F}_p)$) and one in $\ker(\pi_p + 1)$ (i.e., in $E(\mathbb{F}_{p^2})$ with $x$-coordinate in $\mathbb{F}_p$). This allows them to avoid timing attacks, while keeping the same primes and exponent range $[-5, 5]$ as in the original CSIDH algorithm. Their algorithm also employs dummy isogenies to mitigate some power analysis attacks, as in [MCR19]. With these improvements, they achieve a speed-up of 27.35% compared to [MCR19].

We include pseudocode for the algorithm of [OAYT20] in Algorithm 8, to serve as a departure point for our dummy-free algorithm in Section 4.3. Although not described here, the Elligator algorithm is used in this context as an algorithm that allows us to randomly generate points on a given elliptic curve (see [BHKL13]). Specifically, `Elligator(E, u)` returns $T_-$ in $E[\pi_p - 1]$ and $T_+$ in $E[\pi_p + 1]$.

*Remark* 1. Algorithms 7 and 8 can be adapted to use other curve models. The Montgomery model here is used to exploit Montgomery arithmetic for $x$-only scalar multiplication.

## 4.3 Contribution: Fault-attack resistance

This section presents the results obtained on constant-time implementation of CSIDH in the joint work [CCC+19]. The use of dummy operations in the previous constant-time algorithms implies that the attacker can obtain information on the private key by injecting faults during the computation. For example, if

**Algorithm 7:** The Meyer–Campos–Reith CSIDH algorithm for supersingular curves over $\mathbb{F}_p$, where $p = 4\prod_{i=1}^{n} \ell_i - 1$. The ideals $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$, where $\pi_p$ maps to the $p$-th power Frobenius endomorphism on each curve, and the exponent bound vector $(m_1, \ldots, m_n)$, all positive, are system parameters. This algorithm computes exactly $m_i$ isogenies for each $\ell_i$.

**Input:** $A \in \mathbb{F}_p$ such that $E_A : y^2 = x^3 + Ax^2 + x$ is supersingular, and a list of integers $e = (e_1, ..., e_n)$ with $e_i \in \{0, 1, .., m_i\}$ for all $i \leq n$.

**Output:** $A' \in \mathbb{F}_p$, the curve parameter of the resulting curve $E_{A'}$.

1 Initialize $k = 4$, $e = (e_1, ..., e_n)$ and $f = (f_1, ..., f_n)$, where $f_i = m_i - e_i$.
2 **while** *some* $e_i \neq 0$ *or* $f_i \neq 0$ **do**
3  Sample random values $x \in \mathbb{F}_p$ until $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$.
4  Set $P = (x : 1)$, $P \leftarrow [k]P$, $S = \{i \mid e_i \neq 0 \text{ or } f_i \neq 0\}$.
5  **foreach** $i \in S$ **do**
6  Let $m = \prod_{j \in S, j \geq i} \ell_i$.
7  Set $K \leftarrow [m]P$.
8  **if** $K \neq \mathcal{O}$ **then**
9  **if** $e_i \neq 0$ **then**
10  $(E_{A'}, \phi) \leftarrow \texttt{QuotientIsogeny}(E_A, \langle K \rangle)$
11  $A \leftarrow A'$, $P \leftarrow \phi(P)$, $e_i \leftarrow e_i - 1$.
12  **else**
13  $(\_, \_) \leftarrow \texttt{QuotientIsogeny}(E_A, \langle K \rangle)$ `// dummy`
14  $A \leftarrow A$, $P \leftarrow [\ell_i]P$, $f_i \leftarrow f_i - 1$.
15  **if** $e_i = 0$ *and* $f_i = 0$ **then**
16  Set $k \leftarrow k \cdot \ell_i$.
17 **return** $A'$

one of the values in Line 19 of Algorithm 8 is modified without affecting the final result, then the adversary learns whether the corresponding exponent $e_i$ was zero at that point.

We propose an approach to constant-time CSIDH without dummy computations, making every operation essential for a correct final result. This gives us some natural resistance to fault injections, at the cost of approximately a twofold slowdown. Our approach is to change the format of secret exponent vectors $(e_1, \ldots, e_n)$. In both the original CSIDH and the Onuki *et al.* variants, the exponents $e_i$ are sampled from an integer interval $[-m_i, m_i]$ centered on 0. For naive CSIDH, evaluating the action of $\mathfrak{l}_i^{e_i}$ requires evaluating between 0 and $m_i$ isogenies corresponding to either the ideal $\mathfrak{l}_i$ (for positive $e_i$) or $\mathfrak{l}_i^{-1}$ (for negative $e_i$). If we follow the approach of [OAYT20], then we must also compute $m_i - |e_i|$ dummy $\ell_i$-isogenies to ensure a constant-time behaviour.

For our new algorithm, the exponents $e_i$ are uniformly sampled from sets

$$\mathcal{S}(m_i) = \{e \mid e = m_i \bmod 2 \text{ and } |e| \leq m_i\},$$

that is, centered intervals containing only even or only odd integers. The interesting property of these sets is that a vector drawn from $\mathcal{S}(m)^n$ can always be rewritten (in a non-unique way) as a sum of $m$ vectors with entries $\{-1, +1\}$ (i.e., vectors in $\mathcal{S}(1)^n$). But the action of a vector drawn from $\mathcal{S}(1)^n$ can clearly be implemented in constant-time without dummy operations: for each coefficient $e_i$, we compute and evaluate the isogeny associated to $\mathfrak{l}_i$ if $e_i = 1$, or the one associated to $\mathfrak{l}_i^{-1}$ if $e_i = -1$. It follows that we can compute the action of vectors drawn from $\mathcal{S}(m)^n$ by repeating this step $m$ times.

More generally, we want to evaluate the action of vectors $(e_1, \ldots, e_n)$ drawn from $\mathcal{S}(m_1) \times \cdots \times \mathcal{S}(m_n)$. Algorithm 9 achieves this in constant-time, and without using dummy operations. The outer loop at Line 3 is repeated exactly $\max(m_i)$ times, but the inner "if" block at Line 5 is only executed $m_i$ times for each $i$; it is clear that this flow does not depend on secrets. Inside the "if" block, the coefficients $e_i$ are implicitly interpreted as

$$|e_i| = \underbrace{1 + 1 + \cdots + 1}_{e_i \text{ times}} + \underbrace{(1 - 1) - (1 - 1) + (1 - 1) - \cdots}_{m_i - e_i \text{ times}},$$

i.e., the algorithm starts by acting by $\mathfrak{l}_i^{\mathtt{sign}(e_i)}$ for $e_i$ iterations, then alternates between $\mathfrak{l}_i$ and $\mathfrak{l}_i^{-1}$ for $m_i - e_i$ iterations. We assume that the $\mathtt{sign} : \mathbb{Z} \to \{\pm 1\}$ operation is implemented in constant time, and that $\mathtt{sign}(0) = 1$. If one is careful to implement the isogeny evaluations in constant-time, then the full algorithm is also constant-time.

Note that Algorithm 9 is only an idealized version of the CSIDH group action algorithm. Indeed, like in [MCR19, OAYT20], it may happen in some iterations that Elligator outputs points of order not divisible by $\ell_i$, and thus the action of $\mathfrak{l}_i$ or $\mathfrak{l}_i^{-1}$ cannot be computed in that iteration. In this case, we simply skip the loop and retry later: this translates into the variable $z_i$ not being decremented, so the total number of iterations may end up being larger than $\max(m_i)$. If the input value $u$ fed to Elligator is random, its output is uncorrelated to secret values[2], and thus the fact that an iteration is skipped does not leak information on the secret. The resulting algorithm is summarized in Algorithm 10.

To maintain $\lambda$ bits of classical security, the bounds $m_i$ must be chosen so that the key space is at least as large as $2^\lambda$. For example, the original implementation [CLM+18] samples secrets in $[-5, 5]^{74}$, which gives a key space of size $11^{74}$; hence, to get the same security we would need to sample secrets in $\mathcal{S}(10)^{74}$. But a constant-time version of CSIDH-512 *à la* Onuki *et al.* only needs to evaluate five isogeny steps per prime $\ell_i$, whereas Algorithm 10 would need to evaluate ten isogeny steps, leading to an approximately twofold slowdown for this variant compared to [OAYT20]. The field operation counts and clock

[2] Assuming the usual heuristic assumptions on the distribution of the output of Elligator, see [BBC+21].

cycle counts for the constant-time CSIDH-512 implementations of [CCC+19][3] is given in Table 4.1 and Table 4.2 below.

## 4.4 Contribution: Derandomized CSIDH

### 4.4.1 Flawed pseudorandom number generators

Constant-time algorithms are usually allowed to depend on randomness, meaning that the computation time variations cannot depend on secret parameters, but may depend on random elements. Pseudorandom Number Generators (PRNG) return a sequence of numbers that is computationally undistinguishable from a real random sequence, from a seed generated via a source of entropy such as keyboard strokes, or nuclear disintegration). The quality of the PRNG output depends on the quality of entropy available.

Flawed PRNGs can have disastrous consequences. If the source of randomness is not sufficiently strong, an attacker might predict a supposed random number and gain precious information on the computation. For example, [ABC+19] describes a voting machine system in Brazil with a PRNG seeded with switch-on time, but most of the machines were turned on at 8:00 exactly [ABC+19], creating a breach in their security. Even a bias in the PRNG can be sufficient, as the example of Taiwanese digital signatures showed [BCC+13]: several RSA public keys had primes in common, making them insecure.

The algorithms presented in the previous section depend on the availability of high-quality randomness for their security: the input to Elligator must be randomly chosen to ensure that the total running time is uncorrelated to the private key. Typically, this would imply the use of a PRNG seeded with high quality true randomness that must be kept secret. An attack scenario where the attacker may know the output of the PRNG, or where the quality of PRNG output is less than ideal, therefore degrades the security of all algorithms. This is true even when the secret was generated with a high-quality PRNG if the keypair is static, and the private key is then used by an algorithm with low-quality randomness.

### 4.4.2 Derandomized CSIDH with dummies

We can avoid this issue completely if points of order $\prod \ell_i^{|m_i|}$, where $|m_i|$ is the maximum possible exponent (in absolute value) for $\ell_i$, are available from the start. Unfortunately this is not possible with standard CSIDH, because such points are defined over field extensions of exponential degree.

Instead, we suggest modifying CSIDH as follows. First, we take a prime $p = 4\prod_{i=1}^{n} \ell_i - 1$ such that $\lceil n \log(3) \rceil = 2\lambda$, where $\lambda$ is a security parameter, and we restrict to exponents of the private key sampled from $\{-1, 0, 1\}$. Then we compute two points of order $(p+1)/4$ on the starting public curve, one in $\ker(\pi_p - 1)$ and the other in $\ker(\pi_p + 1)$, where $\pi_p$ is the Frobenius endomorphism.

[3] available at `https://github.com/JJChiDguez/csidh`.

This computation involves no secret information, and can be implemented in variable-time; furthermore, if the starting curve is the initial curve with $A = 0$, or a public curve corresponding to a long term private key, these points can be precomputed offline and attached to the system parameters or the public key. We also remark that for static public keys, this would additionally speed-up the key validation process since a point of order $p + 1$ would be immediately accessible.

Since we have restricted exponents to $\{-1, 0, 1\}$, every $\ell_i$-isogeny in Algorithm 8 can be computed using only (the images of) the two precomputed points. There is no possibility of failure in the test of Line 13, and no need to sample any other point. We note that this algorithm still uses dummy operations.

### 4.4.3 Derandomized dummy-free CSIDH

If fault-injection attacks are a concern, the exponents can be further restricted to $\{-1, 1\}$, and the group action evaluated as in (a stripped down form of) Algorithm 10. However this further increases the size of $p$, as $n$ must now be equal to $2\lambda$.

This protection comes at a steep price: at the 128 bits security level, the prime $p$ goes from 511 bits to almost 1500. The resulting field arithmetic would be considerably slower, although the global running time would be slightly offset by the smaller number of isogenies to evaluate. Besides, the computation of large degree isogenies would benefit from the latter published work of [BDLS20], which shows that they can be computed in $\mathcal{O}(\sqrt{\ell})$ instead of $\mathcal{O}(\ell)$.

On the positive side, the resulting system would have much stronger quantum security. Indeed, the best known quantum attacks are exponential in the size of the key space ($\approx 2^{2\lambda}$ here), but only subexponential in $p$ (see [CJS14, DKS18, CLM+18]). Since our modification more than doubles the size of $p$ without changing the size of the key space, quantum security is automatically increased. For this same reason, for security levels beyond NIST–1 (64 quantum bits of security), the size of $p$ increases more than linearly in $\lambda$, and the variant proposed here becomes natural. Finally, parameter sets with a similar imbalance between the size of $p$ and the security parameter $\lambda$ have already been considered in the context of isogeny based signatures [DG18], where they provide tight security proofs in the quantum random oracle model.

## 4.5 Following constant-time implementations

We explored ways to protect CSIDH implementations against fault injection and flawed PRNG. Several other studies have been published afterwards, and we briefly summarize their content for completeness.

- Published in 2020, the work of [BDLS20] drastically reduces the computational effort to compute large degree isogenies from $\mathcal{O}(\ell)$ to $\mathcal{O}(\sqrt{\ell})$. Although this makes original CSIDH and other variants more practical, it

is even more beneficial for the derandomized version of Section 4.4 which involves larger prime-degree isogenies.

- The works of [CKM+20], [LH21], [TDEP21] show that fault injection is not that easy to implement in practice, and rather propose to detect fault injection during the computation while keeping dummy operations, instead of avoiding them. They propose mechanisms to detect such intrusions, which make the scheme faster than the dummy-free version proposed above.
- The work of [BBC+21], named CTIDH, shows a faster way for constant-time implementations of CSIDH (with dummies) by carefully choosing the exponent sets using batches of primes.

**Algorithm 8:** The Onuki–Aikawa–Yamazaki–Takagi CSIDH algorithm for supersingular curves over $\mathbb{F}_p$, where $p = 4\prod_{i=1}^{n} \ell_i - 1$. The ideals $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$, where $\pi_p$ maps to the $p$-th power Frobenius endomorphism on each curve, and the exponent bound vector $(m_1, \ldots, m_n)$, are system parameters. This algorithm computes exactly $m_i$ isogenies for each $\ell_i$.

**Input:** $A \in \mathbb{F}_p$ such that $E_A : y^2 = x^3 + ax^2 + x$ is supersingular, and an integer exponent vector $(e_1, \ldots, e_n)$ with each $e_i \in [-m_i, m_i]$.

**Output:** $B$ the curve parameter of $E_B \colon y^2 = x^3 + Bx^2 + x$ such that $E_B = \mathfrak{l}_1^{e_1} \cdot \cdots \cdot \mathfrak{l}_n^{e_n} \cdot E_A$.

$(e'_1, \ldots, e'_n) \leftarrow (m_i - |e_1|, \ldots, m_i - |e_n|)$ `// Number of dummy computations`
$E_B \leftarrow E_A$
**while** *some* $e_i \neq 0$ *or* $e'_i \neq 0$ **do**
  $S \leftarrow \{i \mid e_i \neq 0 \text{ or } e'_i \neq 0\}$
  $k \leftarrow \prod_{i \in S} \ell_i$
  $u \leftarrow \texttt{Random}\big(\{2, \ldots, \frac{p-1}{2}\}\big)$
  $(T_-, T_+) \leftarrow \texttt{Elligator}(E_B, u)$ `//` $T_- \in E_B[\pi_p - 1]$ `and` $T_+ \in E_B[\pi_p + 1]$
  $(P_0, P_1) \leftarrow \big([(p+1)/k]T_+, [(p+1)/k]T_-\big)$
  **for** $i \in S$ **do**
    $s \leftarrow \operatorname{sign}(e_i)$ `// Ideal` $\mathfrak{l}_i^s$ `to be used`
    $Q \leftarrow [k/\ell_i]P_{\frac{1-s}{2}}$ `// Secret kernel point generator`
    $P_{\frac{1+s}{2}} \leftarrow [\ell_i]P_{\frac{1+s}{2}}$ `// Secret point to be multiplied`
    **if** $Q \neq \infty$ **then**
      **if** $e_i \neq 0$ **then**
        $(E_B, \varphi) \leftarrow \texttt{QuotientIsogeny}(E_B, Q)$
        $(P_0, P_1) \leftarrow \big(\varphi(P_0), \varphi(P_1)\big)$
        $e_i \leftarrow e_i - s$
      **else**
        $E_B \leftarrow E_B$; $P_{\frac{1-s}{2}} \leftarrow [\ell_i]P_{\frac{1-s}{2}}$; $e'_i \leftarrow e'_i - 1$ `// Dummies`
    $k \leftarrow k/\ell_i$
**return** $B$

**Algorithm 9:** An idealized dummy-free constant-time evaluation of the CSIDH group action.

**Input:** $E$ an elliptic curve and $(e_1, \ldots, e_n) \in \mathcal{S}(m_1) \times \cdots \times \mathcal{S}(m_n)$ secret vector

**Output:** $(\prod_{i=0}^{n} \mathfrak{l}_i^{e_i}) \cdot E$

$(t_1, \ldots, t_n) \leftarrow (\texttt{sign}(e_1), \ldots, \texttt{sign}(e_n))$ `// Secret`

$(z_1, \ldots, z_n) \leftarrow (m_1, \ldots, m_n)$ `// Not secret`

**while** *some* $z_i \neq 0$ **do**

  **for** $i \in \{1, \ldots, n\}$ **do**

    **if** $z_i > 0$ **then**

      $E \leftarrow \mathfrak{l}_i^{t_i} \cdot E$

      $b = \texttt{isequal}(e_i, 0)$

      $e_i \leftarrow e_i - t_i$

      $t_i \leftarrow (-1)^b \cdot t_i$ `// Swap sign when` $e_i$ `has gone past 0`

      $z_i \leftarrow z_i - 1$

**return** $E$

| Implementation | Constant-time | Dummy-free | M | S | A | Ratio |
|---|---|---|---|---|---|---|
| [CLM+18] | No | No | 0.252 | 0.130 | 0.348 | 0.26 |
| [MCR19] | Yes | No | 1.054 | 0.410 | 1.053 | 1.00 |
| [OAYT20] | Yes | No | 0.733 | 0.244 | 0.681 | 0.67 |
| **Algo. 10**, [CCC+19] | **Yes** | **Yes** | **1.319** | **0.423** | **1.389** | **1.19** |

Table 4.1: Field operation counts for constant-time CSIDH-512. Counts are given in millions of operations, averaged over 1024 random experiments. The performance ratio uses [MCR19] as a baseline, considers only multiplication and squaring operations, and assumes $M = S$.

| Implementation | Constant-time | Dummy-free | Mcycles | Ratio |
|---|---|---|---|---|
| [CLM+18] | No | No | 155 | 0.39 |
| [MCR19] | Yes | No | 395 | 1.00 |
| **Algo. 10**, [CCC+19] | **Yes** | **Yes** | **481** | **1.22** |

Table 4.2: Clock cycle counts for constant-time CSIDH-512 implementations, averaged over 1024 experiments. The ratio is computed using [MCR19] as baseline implementation.

**Algorithm 10:** Dummy-free randomized constant-time CSIDH class group action for supersingular curves over $\mathbb{F}_p$, where $p = 4\prod_{i=1}^{n} \ell_i - 1$. The ideals $\mathfrak{l}_i = (\ell_i, \pi_p - 1)$, where $\pi_p$ maps to the $p$-th power Frobenius endomorphism on each curve, and the vector $(m_1, \ldots, m_n)$ of exponent bounds, are system parameters. This algorithm computes exactly $m_i$ isogenies for each ideal $\mathfrak{l}_i$.

**Input:** A supersingular curve $E_A$ over $\mathbb{F}_p$, and an exponent vector $(e_1, \ldots, e_n)$ with each $e_i \in [-m_i, m_i]$ and $e_i \equiv m_i \pmod 2$.

**Output:** $E_B = \mathfrak{l}_1^{e_1} \cdot \cdots \cdot \mathfrak{l}_n^{e_n} \cdot E_A$.

$(t_1, \ldots, t_n) \leftarrow \left(\frac{\mathtt{sign}(e_1)+1}{2}, \ldots, \frac{\mathtt{sign}(e_n)+1}{2}\right)$ `// Secret`

$(z_1, \ldots, z_n) \leftarrow (m_1, \ldots, m_n)$ `// Not secret`

$E_B \leftarrow E_A$

**while** *some* $z_i \neq 0$ **do**

  $u \leftarrow \mathtt{Random}\left(\{2, \ldots, \frac{p-1}{2}\}\right)$

  $(T_-, T_+) \leftarrow \mathtt{Elligator}(E_B, u)$ `//` $T_- \in E_B[\pi_p - 1]$ `and` $T_+ \in E_B[\pi_p + 1]$

  $(T_+, T_-) \leftarrow ([4]T_+, [4]T_-)$ `//` `Now` $T_+, T_- \in E_B\left[\prod_i \ell_i\right]$

  **for** $i \in \{1, \ldots, n\}$ **do**

    **if** $z_i \neq 0$ **then**

      $(G_+, G_-) \leftarrow (T_+, T_-)$

      $\mathtt{cswap}(G_+, G_-, t_i)$ `// Secret kernel generator:` $G_+$

      $\mathtt{cswap}(T_+, T_-, t_i)$ `// Secret point to be multiply:` $T_-$

      **for** $j \in \{i+1, \ldots, n\}$ **do**

        $G_+ \leftarrow [\ell_j]G_+$

      **if** $G_+ \neq \infty$ **then**

        $(E_B, \phi) \leftarrow \mathtt{QuotientIsogeny}(E_B, G_0)$

        $(T_+, T_-) \leftarrow \big(\phi(T_+), \phi(T_-)\big)$

        $b \leftarrow \mathtt{isequal}(e_i, 0)$

        $e_i \leftarrow e_i + (-1)^{t_i}$

        $t_i \leftarrow t_i \oplus b$

        $z_i \leftarrow z_i - 1$

      $T_1 \leftarrow [\ell_i]T_1$

      $\mathtt{cswap}(T_+, T_-, t_i)$

**return** $B$

# Part III

# CSIDH generalization: higher-degree supersingular group actions

# Chapter 5

# $(d, \epsilon)$-structures

**Abstract** A supersingular curve is defined over $\mathbb{F}_p$ if its defining equation coefficients are elements of $\mathbb{F}_p$. This property is equivalent to having a degree one isogeny (i.e. an isomorphism) from the curve to its Galois conjugate. Due to the existence of a free and transitive group action of the ideal class group on the set of supersingular curve defined over $\mathbb{F}_p$ (see Section 2.6), they have been used to build efficient cryptosystems like CSIDH [CLM+18] or CSURF [CD20]. While the isogeny graph for curves over $\mathbb{F}_p$ is well known ([DG16]), the isogeny graph for curves having a degree $d$ isogeny to their Galois conjugate has not been studied. In this chapter, we prove the existence of a free and transitive class group action on the set of curves having a $d$-isogeny to their conjugate. We use this action to study the isogeny graph of such curves.

The results of this section have been published in [CS21].

## 5.1 Curves with a $d$-isogeny to their conjugate

In this section we define and study the properties of elliptic curves with a $d$-isogeny to their conjugate.

### 5.1.1 Galois conjugates

The Galois conjugate of an elliptic curve over $\mathbb{F}_{p^2}$ is its image under the $p$-power Frobenius. Let $E$ be an elliptic curve. If $E/\mathbb{F}_{p^2}$ is an elliptic curve, then its *Galois-conjugate* curve $E^{(p)}$ is defined by $p$-th powering all of the coefficients in the defining equation of $E$. The curve $E$ and its conjugate $E^{(p)}$ are connected by inseparable "Frobenius" $p$-isogenies $\pi_p : E \to E^{(p)}$ and $\pi_p : E^{(p)} \to E$, defined by $p$-th powering the coordinates (abusing notation, all inseparable $p$-isogenies will be denoted by $\pi_p$). Observe that $(E^{(p)})^{(p)} = E$, and the composition of $\pi_p : E \to E^{(p)}$ and $\pi_p : E^{(p)} \to E$ is the $p^2$-power Frobenius endomorphism $\pi_E$ of $E$.

Conjugation also operates on isogenies: each isogeny $\phi : E \to E'$ defined over $\mathbb{F}_{p^2}$ has a Galois conjugate isogeny $\phi^{(p)} : E^{(p)} \to E'^{(p)}$, defined by $p$-th powering all of the coefficients in a rational map defining $\phi$. We always have

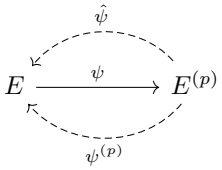
$$(\phi^{(p)})^{(p)} = \phi \qquad \text{and} \qquad \pi_p \circ \phi = \phi^{(p)} \circ \pi_p \,.$$

In particular, conjugation gives an isomorphism of rings between $\mathrm{End}(E)$ and $\mathrm{End}(E^{(p)})$, because $(\phi_1 + \phi_2)^{(p)} = {\phi_1}^{(p)} + {\phi_2}^{(p)}$ and $(\phi_1\phi_2)^{(p)} = {\phi_1}^{(p)}{\phi_2}^{(p)}$.

### 5.1.2 $(d, \epsilon)$-structures

Let $p > 3$ be a prime, and $d$ a squarefree integer prime to $p$.[1] We are interested in elliptic curves $E/\mathbb{F}_{p^2}$ equipped with a $d$-isogeny $\psi : E \to E^{(p)}$. Given any such $d$-isogeny $\psi$, we have two returning $d$-isogenies:

$$\psi^{(p)} : E^{(p)} \to E \qquad \text{and} \qquad \widehat{\psi} : E^{(p)} \to E \,.$$

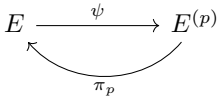



**Definition 29.** Let $E/\mathbb{F}_{p^2}$ be an elliptic curve equipped with a $d$-isogeny $\psi : E \to E^{(p)}$ to its conjugate. We say that $(E, \psi)$ is a $(d, \epsilon)$-*structure* if

$$\widehat{\psi} = \epsilon\psi^{(p)} \quad \text{with} \quad \epsilon \in \{-1, 1\} \,.$$

Each $(d, \epsilon)$-structure $(E, \psi)$ has an *associated endomorphism*

$$\mu := \pi_p \circ \psi \in \mathrm{End}(E) \,.$$




We say that $(E, \psi)$ is ordinary resp. supersingular if $E$ is ordinary resp. supersingular.

The following lemma gives a useful criterion for identifying a $(d, \epsilon)$-structure, though it does not distinguish between $\epsilon = 1$ and $\epsilon = -1$.

**Lemma 25.** *If $E/\mathbb{F}_{p^2}$ is an elliptic curve with $j(E) \notin \{0, 1728\}$ and $\psi : E \to E^{(p)}$ is a $d$-isogeny, then $(E, \psi)$ is a $(d, \pm 1)$-structure if and only if* $\ker \psi^{(p)} = \ker \widehat{\psi}$.

[1] Typically, $p$ is very large and $d$ is very small.

*Proof.* We have $\ker \psi^{(p)} = \ker \widehat{\psi}$ if and only if $\psi^{(p)} = \sigma\widehat{\psi}$ for some $\sigma$ in $\mathrm{Aut}(E)$, and then $\psi^{(p)}\psi = \sigma\widehat{\psi}\psi = \sigma[d]$. If $j(E) \notin \{0, 1728\}$, then $\mathrm{Aut}(E) = \{\pm 1\}$. If $\sigma = 1$ then $\psi$ is a $(d, 1)$-structure; if $\sigma = -1$ then $\psi$ is a $(d, -1)$-structure. □

**Proposition 26.** *If $(E, \psi)$ is a $(d, \epsilon)$-structure and $\mu$ is its associated endomorphism, then*

$$\mu^2 = [\epsilon d]\pi_E \,.$$

*If $\pi_E$ is the Frobenius endomorphism of $E$ and $t_E$ is its trace, then there exists an integer $r$ such that $[r]\mu = [p] + \epsilon\pi_E$ in $\mathrm{End}(E)$, $dr^2 = 2p + \epsilon t_E$ in $\mathbb{Z}$, and the characteristic polynomial of $\mu$ is*

$$P_\mu(T) = T^2 - rdT + dp \,.$$

*Proof.* We have $\psi\pi_p = \pi_p\psi^{(p)}$, so $\mu^2 = \pi_p\psi\pi_p\psi = \pi_p(\pi_p\psi^{(p)})\psi = \pi_E(\psi^{(p)}\psi)$. Now $\psi^{(p)} = \epsilon\widehat{\psi}$ (because $(E, \psi)$ is a $(d, \epsilon)$-structure), so $\psi^{(p)}\psi = [\epsilon d]$, and therefore $\mu^2 = [\epsilon d]\pi_E$. For the rest: $\mu$ has degree $dp$, so it satisfies a quadratic polynomial $P_\mu(T) = T^2 - aT + dp$ for some integer $a$. The first assertion then implies $[a]\mu = \mu^2 + [dp] = [\epsilon d]\pi_E + [dp]$. Squaring, we obtain

$$\begin{aligned}([a]\mu)^2 &= [d]^2(\pi_E^2 + p^2) + 2[dp][\epsilon d]\pi_E \\ &= [d]^2(t_E\pi_E) + 2[dp][\epsilon d]\pi_E \\ &= [\epsilon d]\pi_E([\epsilon d]t_E + 2dp)\,,\end{aligned}$$

so $a^2 = \epsilon d t_E + 2dp$, hence $d \mid a^2$. But $d$ is squarefree, so $d \mid a$, and then $r = a/d$ satisfies the given conditions.

□

*Remark* 2. In the situation of Proposition 26: If $E$ is ordinary, then $\mathbb{Z}[\mu]$ and $\mathbb{Z}[\pi_E]$ are orders in $\mathbb{Q}(\pi_E)$ of discriminant $d^2r^2 - 4dp$ and $t_E^2 - 4p^2 = r^2(d^2r^2 - 4dp)$, respectively, so $|r|$ is the conductor of $\mathbb{Z}[\pi_E]$ in $\mathbb{Z}[\mu]$.

### 5.1.3 Isogenies of $(d, \epsilon)$-structures

The notions of isogenies, quadratic twists and supersingularity can be extended from elliptic curves to these $(d, \epsilon)$-structures with minor adaptations.

**Definition 30.** Let $(E, \psi)$ and $(E', \psi')$ be two $(d, \epsilon)$-structures. We say that an isogeny (resp. isomorphism) $\phi : E \to E'$ is an *isogeny (resp. isomorphism) of $(d, \epsilon)$-structures* if $\psi'\phi = \phi^{(p)}\psi$, that is, if the following diagram commutes:

$$\begin{array}{ccc} E & \xrightarrow{\psi} & E^{(p)} \\ \downarrow{\scriptstyle \phi} & & \downarrow{\scriptstyle \phi^{(p)}} \\ E' & \xrightarrow{\psi'} & (E')^{(p)} \end{array}$$

It is easily verified that isogenies of $(d, \epsilon)$-structures follow the usual rules obeyed by isogenies: the composition of two isogenies of $(d, \epsilon)$-structures is an isogeny of $(d, \epsilon)$-structures, the dual of an isogeny of $(d, \epsilon)$-structures is an isogeny of $(d, \epsilon)$-structures, and every $(d, \epsilon)$-structure has an isogeny to itself (the identity map, for example). Isogeny therefore forms an equivalence relation on $(d, \epsilon)$-structures.

### 5.1.4 Twisting

**Lemma 27.** *Let $(E, \psi)$ be a $(d, \epsilon)$-structure. Let $\delta$ be a nonsquare in $\mathbb{F}_{p^2}$, and let $\sqrt{\delta}$ be a square root of $\delta$ in $\mathbb{F}_{p^4}$. Let $\tau$ be as defined in Section 2.4*

1. *If $(E, \psi)$ is a $(d, \epsilon)$-structure then*

$$(E, \psi)^{\sqrt{\delta}} := (E^{\sqrt{\delta}}, \tau_{(\sqrt{\delta})^{(p-1)}} \circ \psi^{\sqrt{\delta}})$$

   *is a $(d, -\epsilon)$-structure.*

2. *In particular, the isomorphism class of $(E, \psi)^{\sqrt{\delta}}$ is independent of the choice of nonsquare $\delta$.*

3. *$((E, \psi)^{\sqrt{\delta}})^{\sqrt{\delta}} \cong (E, \psi)$.*

4. *If $\phi : (E, \psi) \to (E', \psi')$ is an isogeny of $(d, \epsilon)$-structures, then $\phi^{\sqrt{\delta}}$ induces an isogeny of $(d, -\epsilon)$-structures*

$$\phi^{\sqrt{\delta}} : (E, \psi)^{\sqrt{\delta}} \to (E', \psi')^{\sqrt{\delta}}.$$

*Proof.* To ease notation, write $\alpha$ for $\sqrt{\delta}$. Let $\tilde{\psi} := \tau_{\alpha^{(p-1)}} \psi^{\alpha} = \tau_{\alpha^p} \psi \tau_{\alpha^{-1}} : E^{\alpha} \to E^{\alpha^p} = (E^{\alpha})^{(p)}$. Now $\widehat{\tilde{\psi}} = \tau_{\alpha} \widehat{\psi} \tau_{\alpha^{-p}}$ and $(\tilde{\psi})^{(p)} = \tau_{\alpha^{p^2}} \psi^{(p)} \tau_{\alpha^{-p}} = \tau_{\alpha^{p^2}} \epsilon \widehat{\psi} \tau_{\alpha^{-p}}$ because $\psi^{(p)} = \epsilon \widehat{\psi}$, so $(\tilde{\psi})^{(p)} = \tau_{\alpha^{(p^2-1)}} \epsilon \widehat{(\tilde{\psi})}$. But $\alpha^{(p^2-1)} = -\alpha/\alpha = -1$, because $\alpha$ is the square root in $\mathbb{F}_{p^4}$ of a nonsquare in $\mathbb{F}_{p^2}$; thus $\tau_{\alpha^{p^2-1}} = \tau_{-1} = [-1]$, which proves the first claim. The second and third claims are then straightforward, as is the fourth: if $\psi' \phi = \phi^{(p)} \psi$, then $\tau_{\alpha^{(p-1)}} (\psi')^{\alpha} \phi^{\alpha} = (\phi^{\alpha})^{(p)} \tau_{\alpha^{(p-1)}} \psi^{\alpha}$. □

We call $(E, \psi)^{\sqrt{\delta}}$ the *quadratic twist* of $(E, \psi)$.

*Remark* 3. Twisting takes us from the category of $(d, \epsilon)$-structures into the category of $(d, -\epsilon)$-structures and back again.

**Example 8.** Consider the case $d = 1$. Each $(1, 1)$-structure is $\mathbb{F}_{p^2}$-isomorphic to the base-extension to $\mathbb{F}_{p^2}$ of a curve defined over $\mathbb{F}_p$ (with the 1-isogeny being $[\pm 1]$); the associated endomorphism is the $p$-power Frobenius endomorphism on the base-extended curve, and the integer $r$ of Proposition 26 is the trace of the $p$-power Frobenius. Each $(1, -1)$-structure is the quadratic twist of a $(1, 1)$-structure: essentially, an ordinary $(1, -1)$-structure is isomorphic to a GLS curve [GLS11]. of [Smi16, §3].

**Definition 31.** We write $\mathcal{D}_{d,\epsilon}(p)$ for the set of supersingular $(d,\epsilon)$-structures over $\mathbb{F}_{p^2}$ up to $\mathbb{F}_{p^2}$-isomorphism, and $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ for the graph on $\mathcal{D}_{d,\epsilon}(p)$ whose edges are ($\mathbb{F}_{p^2}$-isomorphism classes of) isogenies of $(d,\epsilon)$-structures. For each prime $\ell \neq p$, we write $\Gamma_\ell(\mathcal{D}_{d,\epsilon}(p))$ for the subgraph of $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ where the edges are $\ell$-isogenies.

Since twisting takes isogenies of $(d,\epsilon)$-structures to isogenies of $(d,-\epsilon)$-structures, in general $(E^{\sqrt{\delta}}, \psi^{\sqrt{\delta}})$ is *not* a $(d,\pm 1)$-structure because conjugation and twisting generally do not commute. Observe that the quadratic twist gives an isomorphism of graphs $\Gamma(\mathcal{D}_{d,\epsilon}(p)) \cong \Gamma(\mathcal{D}_{d,-\epsilon}(p))$.

### 5.1.5 Involutions

If $(E,\psi)$ is a $(d,\epsilon)$-structure with associated endomorphism $\mu$, then

$$-(E,\psi) := (E,-\psi) \qquad \text{and} \qquad (E,\psi)^{(p)} := (E^{(p)}, \psi^{(p)})$$

are $(d,\epsilon)$-structures with associated endomorphisms $-\mu$ and $\mu^{(p)}$, respectively. If $\phi : (E,\psi) \to (E',\psi')$ is an isogeny of $(d,\epsilon)$-structures, then $\phi : -(E,\psi) \to -(E',\psi')$ and $\phi^{(p)} : (E,\psi)^{(p)} \to (E',\psi')^{(p)}$ are also isogenies of $(d,\epsilon)$-structures. We thus have two involutions, **negation** and **conjugation**, on the category of $(d,\epsilon)$-structures and their isogenies.

*Remark* 4. The isogenies $\psi$ and $\pi_p : E \to E^{(p)}$ are both in fact isogenies of $(d,\epsilon)$-structures $(E,\psi) \to (E,\psi)^{(p)}$.

### 5.1.6 Supersingular $(d,\epsilon)$-structures

**Proposition 28.** *Let $(E,\psi)$ be a $(d,\epsilon)$-structure with associated endomorphism $\mu$. If $E$ is supersingular, then*

1. $\mu^2 = [-dp]$.
2. *The trace of Frobenius satisfies $t_E = -2\epsilon p$, and in particular $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p+\epsilon)\mathbb{Z})^2$.*

*Proof.* With the notation of Proposition 26: The curve $E$ is supersingular if and only if $p \mid t_E$. Now $p \nmid d$, so $p \mid r$ by Proposition 26. The characteristic polynomial $P_\mu(T)$ of $\mu$ has discriminant $(rd)^2 - 4dp$; this discriminant cannot be positive, so $|r| \leq 2\sqrt{p/d}$. Since $p \mid r$, we have $r = 0$, so $\mu^2 = [-dp]$, and $t_E = \frac{-2p}{\epsilon} = -2\epsilon p$. □

Proposition 28 tells us that if $(E,\psi)$ is a supersingular $(d,\epsilon)$-structure, then $\epsilon$ is completely determined by the $\mathbb{F}_{p^2}$-isogeny class of $E$. Further, $t_E$ can only be $\pm 2p$: the special supersingular traces $-p$, 0, and $p$ (corresponding to non-quadratic twists of curves of $j$-invariant 0 and 1728, if these are supersingular) cannot occur.

### 5.1.7 Curves with non-integer $d^2$-endomorphisms

There exist rare cases of elliptic curve having a non-integer $d^2$-endomorphism, which are not $(d, \epsilon)$-structures but studied nonetheless for completeness. If $\psi : E \to E^{(p)}$ is a $d$-isogeny but $(E, \psi)$ is *not* a $(d, \pm 1)$-structure, then $\psi^{(p)} \circ \psi$ is a $d^2$-endomorphism that is not $\pm[d]_E$. There are two ways that this can happen. First, if $\mathrm{Aut}(E) \neq \{\pm 1\}$ then we could have $\psi^{(p)} \circ \psi = \sigma[d]$ for some $\sigma$ in $\mathrm{Aut}(E) \setminus \{\pm 1\}$. Second, $\psi^{(p)} \circ \psi$ could be a $d^2$-endomorphism of $E$ with cyclic kernel.

We describe a technique that can be used to determine all of the $j$-invariants of curves with cyclic $d^2$-endomorphisms. These, together with $j = 0$ and 1728, are the curves that we need to be careful with. Recall that if $\Phi_n(X, Y)$ is the level-$n$ classical modular polynomial, then $\Phi_d(j(E_1), j(E_2)) = 0$ if and only if there is a cyclic $n$-isogeny from $E_1$ to $E_2$ (possibly defined over some extension field). Now if $\Phi_n(j(E), j(E)) = 0$ then $E$ has a cyclic $n$-endomorphism, so we just need to be careful with the roots of $\Phi_d(X, X)$ and $\Phi_{d^2}(X, X)$.

**Example 9.** Consider $d = 2$: we want to find all the curves that might have cyclic 2- and 4-endomorphisms. This means we need to be careful with curves whose $j$-invariants are roots (in $\mathbb{F}_{p^2}$) of

$$\Phi_2(X, X) = -(X - 1728)(X - 8000)(X + 3375)^2$$

or

$$\begin{aligned}\Phi_4(X, X) = -2(X - 287496)(X - 54000)^2(X + 3375)^2 \\ \cdot (X^2 + 191025X - 121287375)^2 .\end{aligned}$$

Other curves do not have cyclic 2- and 4-endomorphisms.

**Example 10.** Now consider $d = 3$: in this case, we need to be careful with curves whose $j$-invariants are roots (in $\mathbb{F}_{p^2}$) of

$$\Phi_3(X, X) = -X(X - 54000)(X - 8000)^2(X + 32768)^2$$

or

$$\begin{aligned}\Phi_9(X, X) = -3(X - 8000)^2(X + 32768)^2(X + 12288000)^2 \\ \cdot (X^2 - 153542016X - 1790957481984) \\ \cdot (X^2 - 52250000X + 12167000000)^2 \\ \cdot (X^2 - 1264000X - 681472000)^2 \\ \cdot (X^2 + 117964800X - 134217728000)^2 .\end{aligned}$$

Other curves do not have cyclic 3- or 9-endomorphisms.

## 5.2 Action on supersingular $(d, \epsilon)$-structures

### 5.2.1 Preliminaries on orientations

Proposition 28 tells us that the associated endomorphism of each supersingular $(d, \epsilon)$-structure acts like a square root of $-dp$ in the endomorphism ring.

We can make this notion more precise using *orientations*, as described by Coló and Kohel in [CK20] and Onuki in [Onu21]. Before going further, we recall some generalities. We start by introducing the notions of orientations, primitive orientations and induced orientations from [CK20]. We write $\mathrm{End}^0(E)$ for $\mathrm{End}(E) \otimes \mathbb{Q}$.

**Definition 32** (Orientations)**.** Let $k$ be an imaginary quadratic field, $\mathcal{O}_k$ its ring of integers, and $\mathcal{O}$ an order in $k$.

- A $k$-orientation on an elliptic curve $E/\mathbb{F}_{p^2}$ is a homomorphism $\iota : k \to \mathrm{End}^0(E)$; we call the pair $(E, \iota)$ a *$k$-oriented elliptic curve*.
- We say $\iota$ is an $\mathcal{O}$-orientation, and $(E, \iota)$ is an $\mathcal{O}$-oriented elliptic curve, if $\iota(\mathcal{O}) \subseteq \mathrm{End}(E)$.
- An $\mathcal{O}$-orientation $\iota : k \to \mathrm{End}^0(E)$ is *primitive* if $\iota(\mathcal{O}) = \mathrm{End}(E) \cap \iota(k)$: that is, if $\iota$ is "full" in the sense that it does not extend to an $\mathcal{O}'$-orientation for any strict super-order $\mathcal{O}' \supset \mathcal{O}$.

**Example 11.** Let $E$ be a supersingular elliptic curve defined over $\mathbb{F}_p$. Then the homomorphism $\iota : \mathbb{Q}(\sqrt{-p}) \longrightarrow End^0(E)$ that maps $\sqrt{-p}$ to the $p$-power Frobenius $\pi_p$ is a $\mathbb{Q}(\sqrt{-p})$-orientation, and a $\mathbb{Z}[\sqrt{-p}]$-orientation. If the endomorphism ring of the curve over $\mathbb{F}_p$ is $\mathbb{Z}[\sqrt{-p}]$ then $\iota$ is also a primitive $\mathbb{Z}[\sqrt{-p}]$-orientation.

**Definition 33** (Induced orientation)**.** Let $(E, \iota)$ be a $k$-oriented elliptic curve. If $\phi : E \to E'$ is an isogeny, then there is an *induced $k$-orientation* $\phi_*(\iota)$ on $E'$ defined by

$$\phi_*(\iota) : \alpha \longmapsto \frac{1}{\deg(\phi)} \phi \circ \iota(\alpha) \circ \widehat{\phi}\,.$$

We now describe isogenies and isomorphisms that preserve the orientation.

**Definition 34** ($k$-oriented isogenies and isomorphisms)**.** Given two oriented curves $(E, \iota)$ and $(E', \iota')$, an isogeny $\phi : E \to E'$ is said to be $k$-oriented, or an isogeny of $k$-oriented elliptic curves, if $\iota' = \phi_*(\iota)$. In this case we write $\phi : (E, \iota) \to (E', \iota')$. If there exists a $k$-oriented isogeny $\tilde{\phi} : (E', \iota') \to (E, \iota)$ such that $\tilde{\phi} \circ \phi = [1]_E$ and $\phi \circ \tilde{\phi} = [1]_{E'}$, then we say that $\phi$ is a $k$-oriented isomorphism, and we write $(E, \iota) \cong (E', \iota')$.

Note that $\phi : (E, \iota) \to (E', \iota')$ is an oriented isomorphism if and only if the underlying isomorphism of curves $\phi$ satisfies $\phi \circ \iota(\alpha) = \iota'(\alpha) \circ \phi$ for all $\alpha$ in $k$.

From $k$-oriented isogenies we can define the notion of horizontal, ascending and descending isogenies similarly to isogenies of ordinary elliptic curves. Let $\phi : (E, \iota) \to (E', \iota')$ be a $k$-oriented isogeny, with $\deg \phi = \ell$ be a prime (not equal to $p$), Then $\iota$ is a primitive $\mathcal{O}$-orientation and $\iota'$ is a primitive $\mathcal{O}'$-orientation for some orders $\mathcal{O}$ and $\mathcal{O}'$ in $k$.

Moreover one of the following cases is true:

- $\mathcal{O} = \mathcal{O}'$, and $\phi$ is said to be *horizontal*; or

- $\mathcal{O} \subset \mathcal{O}'$ with $[\mathcal{O}' : \mathcal{O}] = \ell$, and $\phi$ is said to be *ascending*; or
- $\mathcal{O} \supset \mathcal{O}'$ with $[\mathcal{O} : \mathcal{O}'] = \ell$, and $\phi$ is said to be *descending*.

### 5.2.2 Action on primitive $\mathcal{O}$-oriented curves

With the definitions above, we can describe the properties of action of the ideal class group on the set of $\mathcal{O}$-oriented elliptic curves. Onuki [Onu21] shows that if we restrict to a certain subset of the *primitive* $\mathcal{O}$-oriented curves, then this action is transitive and free.

**Definition 35.** Let $\mathcal{O}$ be an order in a quadratic field $k$ such that $p$ does not split in $k$ or divide the conductor of $\mathcal{O}$. Following [CK20], we let $SS_{\mathcal{O}}(p)$ denote the set of $\mathcal{O}$-oriented supersingular elliptic curves over $\overline{\mathbb{F}}_p$ up to $k$-oriented isomorphism. The subset of primitive $\mathcal{O}$-oriented curves (up to $k$-oriented isomorphism) is denoted by $SS^{\mathrm{pr}}_{\mathcal{O}}(p)$.

**Proposition 29** (Transitive action)**.** *There is a transitive group action*

$$\mathrm{Cl}(\mathcal{O}) \times SS_{\mathcal{O}}(p) \longrightarrow SS_{\mathcal{O}}(p)\,.$$

*Proof.* For any integral invertible ideal $\mathfrak{a}$ in $\mathcal{O}$ and any $\mathcal{O}$-oriented curve $(E, \iota)$, we have a finite subgroup

$$E[\mathfrak{a}] := \{P \in E \mid \iota(\alpha)(P) = 0 \quad \forall \alpha \in \mathfrak{a}\}\,.$$

Now suppose $\mathfrak{a}$ is prime to the conductor of $\mathcal{O}$ in $\mathcal{O}_k$.[2] If $\phi_{\mathfrak{a}} : E \to E/E[\mathfrak{a}]$ is the quotient isogeny, then $(\phi_{\mathfrak{a}})_*(\iota)$ is an $\mathcal{O}$-orientation on $E/E[\mathfrak{a}]$, and $\phi_{\mathfrak{a}}$ is a horizontal isogeny of $\mathcal{O}$-oriented curves. If $\mathfrak{a}$ is principal then $(E/E[\mathfrak{a}], (\phi_{\mathfrak{a}})_*(\iota)) \cong (E, \iota)$, so the map

$$(\mathfrak{a}, (E, \iota)) \mapsto (E/E[\mathfrak{a}], (\phi_{\mathfrak{a}})_*(\iota))$$

extends to fractional ideals and factors through the class group, and as in [CK20] we get a transitive group action $\mathrm{Cl}(\mathcal{O}) \times SS_{\mathcal{O}}(p) \longrightarrow SS_{\mathcal{O}}(p)$. □

We now introduce the main theorem of [Onu21]. Let $\mathcal{O}$ be an order in a quadratic field $k$ such that $p$ does not split in $k$ or divide the conductor of $\mathcal{O}$. Let $\mathcal{J}_{\mathcal{O}}$ denote the set of $j$-invariants of elliptic curves $E$ over $\mathbb{C}$ (not $\overline{\mathbb{F}}_p$) with $\mathcal{O} \subset \mathrm{End}(E)$. All elements in $\mathcal{J}_{\mathcal{O}}$ are algebraic integers, so an elliptic curve whose $j$-invariant is in $\mathcal{J}_{\mathcal{O}}$ has potential good reduction at any prime ideal. Since $\mathcal{J}_{\mathcal{O}}$ is finite, we can take a number field $L$ and a prime ideal $\mathfrak{p}$ of $L$ above $p$ such that for all $j \in \mathcal{J}_{\mathcal{O}}$, there exists an elliptic curve over $L$ with good reduction at $\mathfrak{p}$ and $j$-invariant $j$. Fix an injection of the residue field of $L$ modulo $\mathfrak{p}$ into $\overline{\mathbb{F}}_p$. Let $\mathrm{Ell}(\mathcal{O})$ be the set of isomorphism classes of elliptic curves $E$ over $L$ with good reduction at $p$ and $j$-invariants in $\mathcal{J}_{\mathcal{O}}$. For every such $E$, we let $[\cdot]_E$ be the *normalized* $\mathcal{O}$-orientation: that is, such that for any invariant

[2] Working with the class group, we can always replace ideals that are not prime to the conductor with equivalent integral ideals that are.

differential $\omega$ on $E$, $([\alpha]_E)^*\omega = \alpha\omega$ for all $\alpha$ in $\mathcal{O}$. Then reduction mod $\mathfrak{p}$ defines a map

$$\begin{aligned} \rho : \mathrm{Ell}(\mathcal{O}) &\longrightarrow SS^{\mathrm{pr}}_{\mathcal{O}}(p) \\ E &\longmapsto (\widetilde{E}, [.]_{\widetilde{E}}) , \end{aligned}$$

where $\widetilde{E}$ is the reduction of $E/L$ at $\mathfrak{p}$ and $[\cdot]_{\widetilde{E}}$ is the orientation such that $[\alpha]_{\widetilde{E}} = [\alpha]_E \pmod{\mathfrak{p}}$ for all $\alpha$ in $\mathcal{O}$. The map $\rho$ is surjective up to $p$-conjugation: for all $(E, \iota)$ in $SS^{\mathrm{pr}}_{\mathcal{O}}(p)$, at least one of $(E, \iota)$ and $(E^{(p)}, \iota^{(p)}))$ is in $\rho(\mathrm{Ell}(\mathcal{O}))$ (see [Onu21, Proposition 3.3]).

**Theorem 30** (Onuki [Onu21, Theorem 3.4])**.** *With the notation above:* $\mathrm{Cl}(\mathcal{O})$ *acts freely and transitively on* $\rho(\mathrm{Ell}(\mathcal{O}))$*.*

### 5.2.3 Natural orientation for supersingular $(d, \epsilon)$-structures

We aim to make Theorem 30 more precise and manageable by focusing on the case $k = \mathbb{Q}(\sqrt{-dp})$. We prove that a *natural orientation* arises in this case, and study some of its properties. From now on we let $k = \mathbb{Q}(\sqrt{-dp})$, and let $\mathcal{O}_k$ be the maximal order of $k$.

**Definition 36** (Natural orientation)**.** If $(E, \psi)$ is a supersingular $(d, \epsilon)$-structure and $\mu$ is the associated endomorphism, then

$$\begin{aligned} \iota_\psi : \mathbb{Q}(\sqrt{-dp}) &\longrightarrow \mathrm{End}^0(E) \\ \sqrt{-dp} &\longmapsto \mu \end{aligned}$$

is a $\mathbb{Z}[\sqrt{-dp}]$-orientation by Proposition 28. We call this the *natural* orientation.

**Lemma 31.** *If $E/\mathbb{F}_{p^2}$ is a supersingular elliptic curve with $\#E(\mathbb{F}_{p^2}) = (p+\epsilon)^2$ and $\iota$ is a $\mathbb{Z}[\sqrt{-dp}]$-orientation on $E$, then $\iota$ is the natural orientation for some $(d, \epsilon)$-structure $(E, \psi)$.*

*Proof.* Let $\mu := \iota(\sqrt{-dp})$ in $\mathrm{End}(E)$. We have $\deg(\mu) = dp$ and $p \nmid d$, so $\mu$ factors over $\mathbb{F}_{p^2}$ into the composition of a $d$-isogeny and a $p$-isogeny. Since $E$ is supersingular, the $p$-isogeny is isomorphic to $\pi_p$, and so $\mu = \pi_p\psi$ for some $d$-isogeny $\psi : E \to E^{(p)}$. It remains to show that $\widehat{\psi} = \epsilon\psi^{(p)}$. Now $[-dp] = \mu^2 = \pi_p\psi\pi_p\psi = \psi^{(p)}\pi_p^2\psi = \psi^{(p)}\psi\pi_p^2$, and $\pi_p^2 = [-\epsilon p]$ because $E$ is supersingular with $\#E(\mathbb{F}_{p^2}) = (p+\epsilon)^2$, so $[d] = \epsilon\psi^{(p)}\psi$, and therefore $\widehat{\psi} = \epsilon\psi^{(p)}$. □

**Lemma 32.** *Let $(E, \psi)$ and $(E', \psi')$ be $(d, \epsilon)$-structures with natural orientations $\iota_\psi$ and $\iota_{\psi'}$, respectively. If $\phi : E \to E'$ is an isogeny, then*

$$\phi_*(\iota_\psi) = \iota_{\psi'} \iff \phi^{(p)} \circ \psi = \psi' \circ \phi ;$$

*that is, $\phi$ is an isogeny (resp. isomorphism) of $\mathbb{Z}[\sqrt{-dp}]$-oriented elliptic curves $(E, \iota) \to (E', \iota')$ if and only if it is an isogeny (resp. isomorphism) of $(d, \epsilon)$-structures $(E, \psi) \to (E', \psi')$.*

*Proof.* Let $\mu$ resp. $\mu'$ be the associated endomorphisms of $(E,\psi)$ resp. $(E',\psi')$; then

$$\begin{aligned}
\phi_*(\iota_\psi) = \iota_{\psi'} &\iff \phi_*(\iota_\psi)(\sqrt{-dp}) = \iota_{\psi'}(\sqrt{-dp}) && (\sqrt{-dp} \text{ generates } \mathbb{Q}(\sqrt{-dp})\\
&\iff \phi \circ \mu \circ \widehat{\phi} = \mu'[\deg \phi] && (\text{multiplying by } \deg\phi)\\
&\iff \phi \circ \mu = \mu' \circ \phi && (\text{cancelling } \widehat{\phi})\\
&\iff \phi \circ \pi_p \circ \psi = \pi_p \circ \psi' \circ \phi && (\text{by definition})\\
&\iff \pi_p \circ \phi^{(p)} \circ \psi = \pi_p \circ \psi' \circ \phi && (\pi_p \circ \phi = \phi^{(p)} \circ \pi_p)\\
&\iff \phi^{(p)} \circ \psi = \psi' \circ \phi && (\text{cancelling } \pi_p)
\end{aligned}$$

and the result follows on comparing definitions. □

Coló and Kohel [CK20] and Onuki [Onu21] use class-group actions to study the isogeny graphs $\Gamma(SS_{\mathcal{O}}(p))$ with vertex set $SS_{\mathcal{O}}(p)$ for different orders $\mathcal{O}$. Proposition 33 allows us to transfer their results to our setting of $(d,\epsilon)$-structures.

**Proposition 33.** *The graphs $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ and $\Gamma(SS_{\mathbb{Z}[\sqrt{-dp}]}(p))$ are explicitly isomorphic for $\epsilon = 1$ and $\epsilon = -1$.*

*Proof.* This follows from Lemmas 31 and 32, once we can show that the isomorphism class of any $\mathbb{Z}[\sqrt{-dp}]$-oriented supersingular curve $(E,\iota)$ over $\overline{\mathbb{F}}_p$ contains a representative over $\mathbb{F}_{p^2}$ of order $(p+\epsilon)^2$. Since $j(E)$ is in $\mathbb{F}_{p^2}$, after a suitable $\overline{\mathbb{F}}_p$-isomorphism we may suppose that $E$ is defined over $\mathbb{F}_{p^2}$ and $\#E(\mathbb{F}_{p^2}) = (p+\epsilon)^2$; and then $\iota$ is defined over $\mathbb{F}_{p^2}$ because for a supersingular elliptic curve over $\mathbb{F}_{p^2}$ all of the endomorphisms are defined over $\mathbb{F}_{p^2}$. □

### 5.2.4 Link between natural and induced orientation

Let $k = \mathbb{Q}(\sqrt{-dp})$. The order $\mathbb{Z}[\sqrt{-dp}]$ has index 2 in $\mathcal{O}_k$ if $-dp \equiv 1 \pmod 4$, and is equal to $\mathcal{O}_k$ otherwise. If $-dp \not\equiv 1 \pmod 4$, then, every natural orientation is a primitive $\mathcal{O}_k$-orientation; if $-dp \equiv 1 \pmod 4$, each natural orientation is either a primitive $\mathbb{Z}[\sqrt{-dp}]$ orientation or a primitive $\mathcal{O}_k$-orientation.

**Proposition 34.** *Let $(E,\psi)$ be a supersingular $(d,\epsilon)$-structure with natural orientation $\iota_\psi$.*

1. *If $-dp \not\equiv 1 \pmod 4$, then $\iota_\psi$ is a primitive $\mathcal{O}_k$-orientation.*
2. *If $-dp \equiv 1 \pmod 4$, then $\iota_\psi$ is a primitive $\mathcal{O}_k$-orientation if the associated endomorphism $\mu$ fixes $E[2]$* pointwise*, and a primitive $\mathbb{Z}[\sqrt{-dp}]$-orientation otherwise.*

*Proof.* By definition, $\iota_\psi$ is a $\mathbb{Z}[\sqrt{-dp}]$-orientation. To complete Case (2), it suffices to check whether the element $\iota_\psi(\frac{1}{2}(-1+\sqrt{-dp})) = \frac{1}{2}(\mu - [1])$ of $\mathrm{End}^0(E) \cap \iota_\psi(k)$ is in $\mathrm{End}(E)$ (because $\frac{1}{2}(-1+\sqrt{-dp})$ generates $\mathcal{O}_k$, but is not in $\mathbb{Z}[\sqrt{-dp}]$). This is the case if and only if $\mu - [1]$ factors over $[2]$, if and only if $\mu$ fixes $E[2]$ pointwise. □

In the light of Propositions 33 and 34, we partition $\mathcal{D}_{d,\epsilon}(p)$ into two subsets:

**Definition 37.**

$$\mathcal{D}_{d,\epsilon}(p) = \mathcal{D}^{\max}_{d,\epsilon}(p) \sqcup \mathcal{D}^{\text{sub}}_{d,\epsilon}(p)\,,$$

where $\mathcal{D}^{\max}_{d,\epsilon}(p)$ contains the classes whose natural orientations are primitive $\mathcal{O}_k$-orientations, and $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ contains the classes whose natural orientations are primitive orientations by the order of conductor 2 in $\mathcal{O}_k$.

**Proposition 35.** *If $-dp \not\equiv 1 \pmod 4$, then $\mathcal{D}^{\max}_{d,\epsilon}(p) = \mathcal{D}_{d,\epsilon}(p)$ and $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p) = \emptyset$. If $-dp \equiv 1 \pmod 4$, then $[\mathcal{O}_k : \mathbb{Z}[\sqrt{-dp}]] = 2$, so $\mathcal{D}^{\max}_{d,\epsilon}(p)$ resp. $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ consists of the $(d,\epsilon)$-structures where $\mu$ acts trivially resp. non-trivially on the 2-torsion.*

*Proof.* If $-dp \not\equiv 1 \pmod 4$, then the maximal order of $\mathbb{Q}(\sqrt{-dp})$ is $\mathbb{Z}[\sqrt{-dp}]$, hence all orientations are primitive.

If $-dp \equiv 1 \pmod 4$, then $\mathcal{O}_k = \mathbb{Z}[\frac{1+\sqrt{-dp}}{2}]$, and $[\mathcal{O}_k : \mathbb{Z}[\sqrt{-dp}]] = 2$. Let $(E,\psi)$ be a $(d,\epsilon)$-structure, with associated endomorphism $\mu$. When $\mu$ acts trivially on the 2-torsion, $\frac{1+\mu}{2}$ is the identity and hence belongs to the endomorphism ring. We obtain that $(E,\psi)$ belongs to $\mathcal{D}^{\max}_{d,\epsilon}(p)$. Otherwise, $\frac{1+\mu}{2}$ is not an endomorphism and $(E,\psi)$ belongs to $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$. □

Given Lemma 32, $\ell$-isogenies of $(d,\epsilon)$-structures are "ascending", "descending", and "horizontal" with respect to the natural orientations: we have horizontal $\ell$-isogenies between vertices in $\mathcal{D}^{\max}_{d,\epsilon}(p)$ and between vertices in $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$, while $\mathcal{D}^{\max}_{d,\epsilon}(p)$ and $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ are connected by ascending and descending 2-isogenies. In the language of isogeny volcanoes, vertices in $\mathcal{D}^{\max}_{d,\epsilon}(p)$ form the "crater", and vertices in $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ the "floor".

### 5.2.5 Free and transitive class group action

Having defined the natural orientation in the case $k = \mathbb{Q}(\sqrt{-dp})$, we are now able to prove on which subsets the action is free and transitive, paving the way for a new conjectural hard homogeneous space.

Proposition 33 translates the action of $\text{Cl}(\mathbb{Q}(\sqrt{-dp}))$ on $SS_{\mathbb{Z}[\sqrt{-dp}]}(p)$ defined above into an action on $\mathcal{D}_{d,\epsilon}(p)$. Theorem 36 makes this precise, showing that $\mathcal{D}^{\max}_{d,\epsilon}(p)$ is a principal homogeneous space under $\text{Cl}(\mathcal{O}_k)$, and that if $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ is not empty then it is a principal homogeneous space under $\text{Cl}(\mathbb{Z}(\sqrt{-dp}))$.

**Theorem 36.** *Let $k = \mathbb{Q}(\sqrt{-dp})$, let $\mathcal{O}_k$ be its maximal order, let $p$ be a prime that does not split in $k$, and let $\epsilon = \pm 1$.*

- *The class group $\text{Cl}(\mathcal{O}_k)$ acts freely and transitively on $\mathcal{D}^{\max}_{d,\epsilon}(p)$.*
- *If $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p) \neq 0$, then $\text{Cl}(\mathbb{Z}(\sqrt{-dp}))$ acts freely and transitively on $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$.*

*Proof.* Let $\mathcal{O} = \mathcal{O}_k$ or $\mathbb{Z}[\sqrt{-dp}]$. Theorem 30 tells us that $\text{Cl}(\mathcal{O})$ acts freely and transitively on $\rho(\text{Ell}(\mathcal{O})) \subseteq SS^{\text{pr}}_{\mathcal{O}}(p)$. Given the isomorphism of Proposition 33,

it only remains to prove that $\rho(\mathrm{Ell}(\mathcal{O})) = SS^{\mathrm{pr}}_{\mathcal{O}}(p)$. For any $(E,\iota)$ in $SS^{\mathrm{pr}}_{\mathcal{O}}(p)$, Proposition 3.3 of [Onu21] tells us that $(E,\iota)$ or $(E,\iota)^{(p)}$ is in $\rho(\mathrm{Ell}(\mathcal{O}))$. In our case, *both* are in $\rho(\mathrm{Ell}(\mathcal{O}))$, so the action on $SS^{\mathrm{pr}}_{\mathcal{O}}(p)$ is free: the action of $\mathfrak{d} = (d, \sqrt{-dp})$ on $SS^{\mathrm{pr}}_{\mathcal{O}}(p)$ maps $(E,\iota)$ to $(E,\iota)^{(p)}$, because it maps $(E,\psi)$ to $(E,\psi)^{(p)}$, because $E[\mathfrak{d}] = E[d] \cap \ker\mu = \ker\psi$. □

**Corollary 37.** *Let $k = \mathbb{Q}(\sqrt{-dp})$, with maximal order $\mathcal{O}_k$. If we write $h_k = \#\operatorname{Cl}(\mathcal{O}_k)$, then*

$$\#\mathcal{D}^{\max}_{d,\epsilon}(p) = h_k \qquad \textit{and} \qquad \#\mathcal{D}^{\mathrm{sub}}_{d,\epsilon}(p) = \begin{cases} h_k & \textit{if } -dp \equiv 1 \pmod 8, \\ 3h_k & \textit{if } -dp \equiv 5 \pmod 8, \\ 0 & \textit{otherwise}. \end{cases}$$

*Proof.* By Theorem 36, we have $\#\mathcal{D}^{\max}_{d,\epsilon}(p) = \#\operatorname{Cl}(\mathcal{O}_k)$ and either $\#\mathcal{D}^{\mathrm{sub}}_{d,\epsilon}(p) = 0$ (if $-dp \not\equiv 1 \pmod 4$) or $\#\mathcal{D}^{\mathrm{sub}}_{d,\epsilon}(p) = \#\operatorname{Cl}(\mathbb{Z}[\sqrt{-dp}])$ (if $-dp \equiv 1 \pmod 4$). It remains to compute $\#\operatorname{Cl}(\mathbb{Z}[\sqrt{-dp}])$ in the case $-dp \equiv 1 \pmod 4$, where $\mathbb{Z}[\sqrt{-dp}]$ has conductor 2. In this case, the formula of [Cox13, Theorem 7.24]

$$\#\operatorname{Cl}(\mathcal{O}) = \frac{h_k \cdot f}{[\mathcal{O}_k^\times : \mathcal{O}^\times]} \prod_{\ell \mid f} \left(1 - \left(\frac{\Delta}{\ell}\right)\frac{1}{\ell}\right)$$

simplifies to

$$\#\operatorname{Cl}(\mathbb{Z}[\sqrt{-dp}]) = \frac{\#\operatorname{Cl}(\mathcal{O}_k)}{[\mathcal{O}_k^\times : \mathbb{Z}[\sqrt{-dp}]^\times]} \left(2 - \left(\frac{-dp}{2}\right)\right),$$

where $(-dp/2)$ is the Legendre symbol. The result follows on noting that $[\mathcal{O}_k^\times : \mathbb{Z}[\sqrt{-dp}]^\times] = 1$, because $-dp$ is never $-3$ or $-4$. □

## 5.3 The $(d,\epsilon)$-supersingular isogeny graph

### 5.3.1 General structure

We can now describe the structure of the isogeny graph $\Gamma(\mathcal{D}_{d,\epsilon}(p))$. Factoring isogenies, it suffices to describe $\Gamma_\ell(\mathcal{D}_{d,\epsilon}(p))$ for each prime $\ell \neq p$. The class group actions of Theorem 36 imply the isogeny counts in the theorem illustrated in Table 5.1.

**Proposition 38.** *For $\ell > 2$, each vertex in $\Gamma_\ell(\mathcal{D}_{d,\epsilon}(p))$ has $1 + (\frac{-dp}{\ell})$ horizontal $\ell$-isogenies, and no ascending or descending $\ell$-isogenies. For $\ell = 2$,*

1. *If $-dp \equiv 1 \pmod 8$ then $\mathcal{D}^{\mathrm{sub}}_{d,\epsilon}(p) \neq \emptyset$.*
   - *Each vertex in $\mathcal{D}^{\max}_{d,\epsilon}(p)$ has two horizontal 2-isogenies to vertices in $\mathcal{D}^{\max}_{d,\epsilon}(p)$, no ascending 2-isogenies, and one descending 2-isogeny to a vertex in $\mathcal{D}^{\mathrm{sub}}_{d,\epsilon}(p)$.*

- *Each vertex in* $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p)$ *has no horizontal* 2*-isogenies, one ascending* 2*-isogeny to a vertex in* $\mathcal{D}_{d,\epsilon}^{\max}(p)$*, and no descending* 2*-isogenies.*

2. *If* $-dp \equiv 5 \pmod 8$ *then* $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p) \neq \emptyset$.

   - *Each vertex in* $\mathcal{D}_{d,\epsilon}^{\max}(p)$ *has no horizontal or ascending* 2*-isogenies, and three descending* 2*-isogenies to vertices in* $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p)$.
   - *Each vertex in* $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p)$ *has no horizontal* 2*-isogenies, one ascending* 2*-isogeny to a vertex in* $\mathcal{D}_{d,\epsilon}^{\max}(p)$*, and no descending* 2*-isogenies.*

3. *Otherwise,* $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p) = \emptyset$ *and* $\mathcal{D}_{d,\epsilon}(p) = \mathcal{D}_{d,\epsilon}^{\max}(p)$*. Each vertex in* $\mathcal{D}_{d,\epsilon}(p)$ *has one horizontal* 2*-isogeny, and no ascending or descending* 2*-isogenies.*

*Proof.* Let $\ell > 2$. Let $V$ be a vertex in the graph. From the free and transitive group action given in Theorem 36 there are no descending or ascending $\ell$-isogenies from $V$. If $\ell$ is split, i.e. if $(\frac{-dp}{\ell}) = 1$, then $(\ell)$ is a product of two distinct prime ideals of $\mathcal{O}_k$, and there are two horizontal isogenies from $V$. If $\ell$ is inert, i.e. if $(\frac{-dp}{\ell}) = -1$, then $(\ell)$ is a prime ideal, and there is no horizontal isogeny from $V$. If $\ell$ is ramified, i.e. if $(\frac{-dp}{\ell}) = 0$, then $(\ell)$ is the square of a prime ideal of $\mathcal{O}_k$, and there is one horizontal isogeny from $V$.

Now we consider the case $\ell = 2$. If $-dp \equiv 1 \pmod 4$, then the maximal order is $\mathbb{Z}[\frac{1+\sqrt{-dp}}{2}]$, and $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p) \neq \emptyset$.

- If $-dp \equiv 1 \pmod 8$, then $(\frac{-dp}{2}) = 1$, the ideal $(2)$ is split in $\mathcal{O}_k$, and there are two horizontal 2-isogenies and $\ell - 1 = 1$ descending 2-isogeny from vertices in $\mathcal{D}_{d,\epsilon}^{\max}(p)$.

- If $-dp \equiv 5 \pmod 8$, then $(\frac{-dp}{2}) = -1$, the ideal $(2)$ is inert in $\mathcal{O}_k$, and there are $\ell + 1 = 3$ descending 2-isogenies, and no ascending or horizontal 2-isogenies from vertices in $\mathcal{D}_{d,\epsilon}^{\max}(p)$.

In both cases, from the volcano structure, each vertex in $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p)$ has one ascending 2-isogeny to a vertex in $\mathcal{D}_{d,\epsilon}^{\max}(p)$. □

| Prime $\ell$ | Conditions on $(d,p)$ | Vertex (sub)set | $\rightarrow$ | $\nearrow$ | $\searrow$ |
|---|---|---|---|---|---|
| $\ell = 2$ | $-dp \equiv 1 \pmod 8$ | $\mathcal{D}_{d,\epsilon}^{\max}(p)$ | 2 | 0 | 1 |
| | | $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p)$ | 0 | 1 | 0 |
| | $-dp \equiv 5 \pmod 8$ | $\mathcal{D}_{d,\epsilon}^{\max}(p)$ | 0 | 0 | 3 |
| | | $\mathcal{D}_{d,\epsilon}^{\text{sub}}(p)$ | 0 | 1 | 0 |
| | $-dp \not\equiv 1,5 \pmod 8$ | $\mathcal{D}_{d,\epsilon}(p)$ | 1 | 0 | 0 |
| $\ell > 2$ | – | $\mathcal{D}_{d,\epsilon}(p)$ | $1 + (\frac{-dp}{\ell})$ | 0 | 0 |

Table 5.1: The number of horizontal, ascending, and descending $\ell$-isogenies from each vertex in the $\ell$-isogeny graph $\Gamma_\ell(\mathcal{D}_{d,\epsilon}(p))$.

### 5.3.2 Examples

Figures 5.1, 5.2, and 5.3 display $\ell$-isogeny graphs on $\mathcal{D}_{3,1}(83)$, $\mathcal{D}_{3,1}(101)$, and $\mathcal{D}_{3,-1}(97)$, for various $\ell$ generating the class groups. These figures also form examples of the various 2-isogeny structures listed in Table 5.1. Vertices are encoded using the Hasegawa parameters for $d = 3$ that we will introduce in Section 5.6.2.

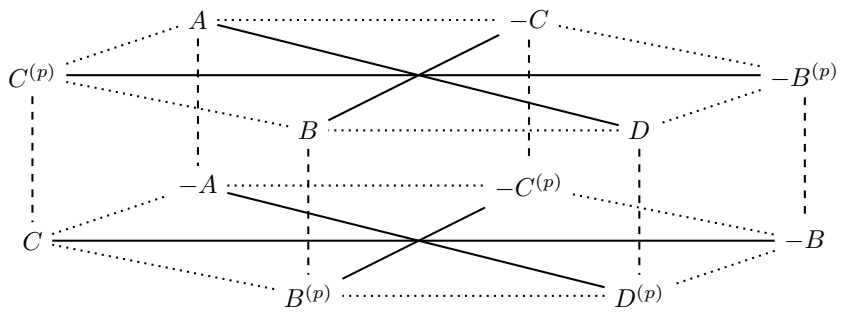



Figure 5.1: $\Gamma_\ell(\mathcal{D}_{3,1}(83))$ for $\ell = 2$ (solid), $\ell = 3$ (dashed) and $\ell = 5$ (dotted). All isogenies are horizontal. We have $\mathrm{Cl}(\mathbb{Q}(\sqrt{-3 \cdot 83})) \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}$, with the $\mathbb{Z}/2\mathbb{Z}$-factor generated by the ideal above 3, and the $\mathbb{Z}/6\mathbb{Z}$-factor generated by an ideal above 5 (we see this in the length-6 cycles). The ideal above 2 is the cube of an ideal above 5. The correspondence between vertex labels and parameters for the degree-3 Hasegawa family of §5.6.2 (with $\Delta = 2$) is $A \leftrightarrow 0$, $B \leftrightarrow 32$, $C \leftrightarrow 40$; the vertex $D$, which corresponds to the Hasegawa parameter $\infty$, is $(E : y^2 = x^3 + 1, \psi)$ where $\psi$ maps $(x, y)$ to $(((72\sqrt{2} + 14)x^3 + (39\sqrt{2} + 56))/x^2, \sqrt{2}(35x^3 + 52)y/x^3)$. Note that $-A = A^{(p)}$ and $-D = D^{(p)}$.

### 5.3.3 Involutions

There are two obvious involutions on $\Gamma(\mathcal{D}_{d,\epsilon}(p))$, negation and conjugation. These are generally not the only involutions. Every prime $\ell$ dividing the discriminant ramifies in $\mathcal{O}_k$ (and $\mathbb{Z}[\sqrt{-dp}]$); the prime $\mathfrak{l}$ over $\ell$ gives an element of order 2 in $\mathrm{Cl}(\mathcal{O}_k)$ (and $\mathrm{Cl}(\mathbb{Z}[\sqrt{-dp}])$), and thus an involution on $\Gamma(\mathcal{D}_{d,\epsilon}(p))$.

Let $\mathfrak{d}_1, \ldots, \mathfrak{d}_n$ be the primes above the prime factors of $d$, and $\mathfrak{p}$ the prime above $p$; note that $[\mathfrak{d}_1] \cdots [\mathfrak{d}_n] = [\mathfrak{p}]$, because $\mathfrak{d}_1 \cdots \mathfrak{d}_n \mathfrak{p} = (\mu)$.

- If $-dp \equiv 1$ or $2 \pmod 4$ then $\mathrm{Cl}(\mathcal{O}_k)[2] = \langle [\mathfrak{d}_1], \ldots, [\mathfrak{d}_n], [\mathfrak{p}] \rangle$, so $\mathrm{Cl}(\mathcal{O}_k)[2] \cong (\mathbb{Z}/2\mathbb{Z})^n$.
- If $-dp \equiv 3 \pmod 4$, then $\mathrm{Cl}(\mathcal{O}_k)[2] = \langle [\mathfrak{a}], [\mathfrak{d}_1], \ldots, [\mathfrak{d}_n], [\mathfrak{p}] \rangle$ where $\mathfrak{a}$ is the ideal above 2, and $\mathrm{Cl}(\mathcal{O}_k)[2] \cong (\mathbb{Z}/2\mathbb{Z})^{n+1}$.

In each case, the action of the ideal class $\prod_i [\mathfrak{d}_i] = [\mathfrak{p}]$ on any $(d, \epsilon)$-structure $(E, \psi)$ is realised by the isogeny $\psi : (E, \psi) \to (E^{(p)}, \psi^{(p)})$, and is therefore equal to the conjugation involution.

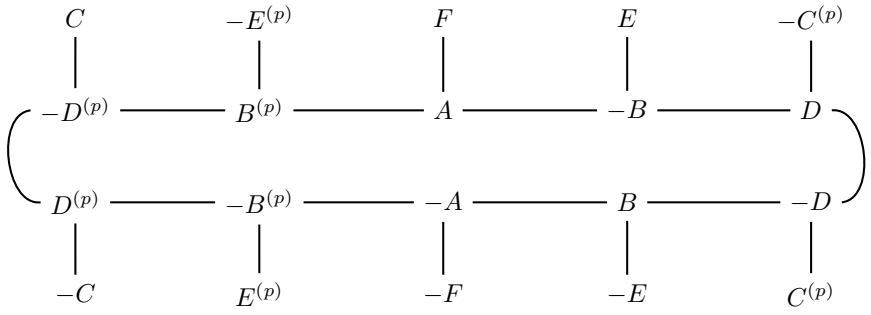



Figure 5.2: $\Gamma_2(\mathcal{D}_{3,1}(101))$ for $\ell = 2$. The class group of $\mathbb{Q}(\sqrt{-303})$ is isomorphic to $\mathbb{Z}/10\mathbb{Z}$, and generated by an ideal over 2 (we see this in the length-10 cycle). The correspondence between vertex labels and parameters for the degree-3 Hasegawa family of §5.6.2 (with $\Delta = 2$) is $A \leftrightarrow 0$, $B \leftrightarrow 6$, $C \leftrightarrow 24$, $D \leftrightarrow 25$, and $E \leftrightarrow 42$; the vertex $F$, which corresponds to the Hasegawa parameter $\infty$, is $(E, \psi)$ with $E : y^2 = x^3+1$ and $\psi : (x, y) \mapsto ((67x^3+66)/x^2, (89x^3+96)\sqrt{2}y/x^3)$. Note that $A^{(p)} = -A$ and $F^{(p)} = -F$. The underlying curves of $B$ and $C$ are isomorphic.

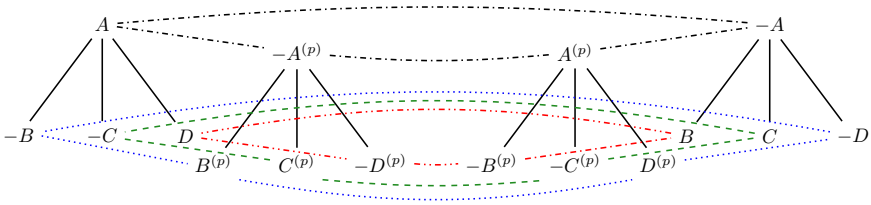



Figure 5.3: The isogeny graphs $\Gamma_2(\mathcal{D}_{3,-1}(97))$ (solid) and $\Gamma_5(\mathcal{D}_{3,-1}(97))$ (dotted). We have $\mathrm{Cl}(\mathbb{Q}(\sqrt{-3 \cdot 97})) \cong \mathbb{Z}/4\mathbb{Z}$, generated by an ideal over 5. The 2-isogenies are ascending/descending up/down the page; the 5-isogenies are horizontal. The correspondence between vertex labels and parameters for the degree-3 Hasegawa family of §5.6.2 (with $\Delta = 5$) is $A \leftrightarrow 47$, $B \leftrightarrow 1$, $C \leftrightarrow 14$, and $D \leftrightarrow 22$. The underlying curves of $A$ and $C$ are isomorphic.

Since the group actions are free, each of the involutions that come from non-trivial two-torsion elements in the class groups – including conjugation – has no fixed points. Negation, on the other hand, can have fixed points: for example, if $p \equiv 3 \pmod 4$ and $E$ is the curve with $j$-invariant 1728, and $i$ is an automorphism of degree 4, then $(E, i)$ is a $(1, 1)$-structure, and $-(E, i) \cong (E, -i)$. This is the only fixed point among $(1, 1)$-structures, and its existence is implied by the fact that the class number of $\mathrm{Cl}(\sqrt{-p})$ is odd when $p \equiv 3 \pmod 4$.

### 5.3.4 Automorphism of order 3

If $-dp \equiv 5 \pmod 8$, then there is an order-3 automorphism $T$ of $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ cycling the triplets of vertices with ascending 2-isogenies to the same vertex in $\mathcal{D}^{\max}_{d,\epsilon}(p)$. In fact $T$ is induced by the action of an ideal class in $\mathrm{Cl}(\mathbb{Z}[\sqrt{-dp}])$. The ideal $\mathfrak{t} = (4, \sqrt{-dp} - 1)\mathbb{Z}[\sqrt{-dp}]$ has order 3 in $\mathrm{Cl}(\mathbb{Z}[\sqrt{-dp}])$, but capitulates to become the principal ideal (2) in $\mathcal{O}_k$ (where $\sqrt{-dp}-1 = 2\omega$, where $\omega$ is the unit $\frac{1}{2}(\sqrt{-dp} - 1)$). Indeed, $\mathfrak{t}$ generates the kernel of the canonical homomorphism $\mathrm{Cl}(\mathbb{Z}[\sqrt{-dp}]) \to \mathrm{Cl}(\mathcal{O}_k)$. Since $\mathfrak{t}$ intersects non-trivially with the conductor, its action on $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ is not well-defined, but we can consider the action of an equivalent ideal in the class group. Let $\prod_i \ell_i^{e_i}$ be the prime factorization of $(dp + 1)/4$ (and note that each $\ell_i$ is odd); then $(\sqrt{-dp} - 1) = \mathfrak{t} \cdot \prod_i \mathfrak{l}_i^{e_i}$ where $\mathfrak{l}_i := (\ell_i, \sqrt{-dp}-1)$. The product $\prod_i \mathfrak{l}_i^{e_i}$ is equivalent to $\mathfrak{t}$ in $\mathrm{Cl}(\mathbb{Z}[\sqrt{-dp}])$, prime to the conductor, and its action on $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ induces the automorphism $T$. In the case where $d = 1$ (CSIDH), this is explained at length in [OT20].

## 5.4 Crossroads: curves with multiple $(d, \epsilon)$-structures

The map $(E, \psi) \mapsto E$ defines a covering from $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ onto a subgraph of the isogeny graph of all supersingular curves over $\mathbb{F}_{p^2}$. For $d_1 \neq d_2$ the images of $\Gamma(\mathcal{D}_{d_1,\epsilon}(p))$ and $\Gamma(\mathcal{D}_{d_2,\epsilon}(p))$ can intersect, forming "crossroads" where we can switch from walking in $\Gamma(\mathcal{D}_{d_1,\epsilon}(p))$ into $\Gamma(\mathcal{D}_{d_2,\epsilon}(p))$, and vice versa.

**Definition 38.** Let $d_1$ and $d_2$ be integers such that $d_1 d_2 > 1$ is squarefree. We say that a supersingular curve $E/\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = (p + \epsilon)^2$ is a $(d_1, d_2)$-*crossroad* if there exist isogenies $\psi_1 : E \to E_1$ and $\psi_2 : E \to E_2$ such that $(E, \psi_1)$ is a $(d_1, \epsilon)$-structure and $(E, \psi_2)$ is a $(d_2, \epsilon)$-structure.

If $(E, \psi)$ is a $(d_1, \epsilon)$-structure, then we can easily check whether $E$ is a $(d_1, d_2)$-crossroad by evaluating the classical modular polynomial $\Phi_{d_2}$ at $(j(E,), j(E)^p)$. However, $(d_1, d_2)$-crossroads are generally very rare. Indeed, if $E$ is a $(d_1, d_2)$-crossroad, then it has an endomorphism of degree $d_1 d_2$ with cyclic kernel. We can therefore enumerate the entire set of $(d_1, d_2)$-crossroads over a given $\mathbb{F}_{p^2}$ by computing the set of roots $j$ of $\Phi_{d_1 d_2}(x, x)$ in $\mathbb{F}_{p^2}$, and then checking for which $j$ we have $\Phi_{d_1}(j, j^p) = 0$. The polynomial $\Phi_{d_1 d_2}(x, x)$ has degree $\prod_\ell(\ell + 1)$ where $\ell$ ranges over the prime factors of $d_1 d_2$, so there are only $O(d_1 d_2)$ $(d_1, d_2)$-crossroads (up to isomorphism) among the $\mathcal{O}(\sqrt{dp})$ vertices in $\Gamma(\mathcal{D}_{d_1,\epsilon}(p))$.

But while crossroads are rare, computing the few examples is relatively easy, and computing $(d_1, d_2)$-crossroads gives us a useful way of quickly constructing some vertices in $\Gamma(\mathcal{D}_{d_1,\epsilon}(p))$ (and in $\Gamma(\mathcal{D}_{d_2,\epsilon}(p))$) at least when the $d_i$ are small. Suppose we want to construct a vertex in $\Gamma(\mathcal{D}_{d,\epsilon}(p))$. Choose the smallest squarefree $d'$ such that $p$ does not split in the maximal order $\mathcal{O}_k$ of $k = \mathbb{Q}(\sqrt{-dd'})$, and then construct a curve $E/\mathbb{F}_{p^2}$ from a root $j$ in $\mathbb{F}_{p^2}$ of the Hilbert class polynomial for $\mathcal{O}_k$. After a suitable twist, $E$ is a supersingular $(d, d')$-crossroad with $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p + \epsilon)\mathbb{Z})^2$. All other vertices in $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ can then be reached through the class group action.

Another way of enumerating crossroads: Suppose that $E$ is a $(d_1, d_2)$ crossroad, i.e. $E$ has a $(d_1, \epsilon)$-structure *and* a $(d_2, \epsilon)$-structure. Composing the $d$-isogeny with the dual of the $d_2$-isogeny $E$ has an endomorphism $\alpha$ of degree $d_1 d_2$. Its characteristic polynomial is $X^2 - t_\alpha X + d_1 d_2$, which has discriminant $\Delta_\alpha = t_\alpha^2 - 4 d_1 d_2$. Because $\Delta_\alpha$ is less than zero, there is only a small number of possible $t_\alpha$.

We can use this necessary condition to find all $(d_1, d_2)$ crossroads : for each $t_\alpha$ such that $t_\alpha^2 < 4 d_1 d_2$, compute the roots of the Hilbert class polynomial of the discriminant $\Delta_\alpha = t_\alpha^2 - 4 d_1 d_2$. For each root $r$, test if the elliptic curves with $j$-invariant $r$ are supersingular, and have a $d_1$-isogeny and $d_2$-isogeny to their conjugate. Algorithm 11 computes a crossroad, if it exists, for two squarefree integers $d_1$ and $d_2$ coprime to $p$, and such that $d_1 d_2$ is squarefree.

**Algorithm 11:** Find $(d_1, d_2)$-crossroad

**Input:** $p$ prime, $d_1$, $d_2$ squarefree integers coprime with $p$, and such that $d_1 d_2$ is squarefree
**Output:** A crossroad in $\mathcal{D}_{d_1,\epsilon}(p)$ and $\mathcal{D}_{d_2,\epsilon}(p)$

```
Φ_{d1 d2} ← ModularPolynomial(d1 d2)                    // precomputation
for j in Roots(Φ_{d1 d2}(x, x), F_{p^2}) do
    if Φ_{d1}(j, j^p) = 0 then
        return j
return None
```

*Remark* 5. A formula for $\Phi_{d_1 d_2}$ in term of $\Phi_{d_1}$ and $\Phi_{d_2}$ is given in [Cox13] Proposition 13.14.

## 5.5 Map from $(d, \epsilon)$-structures to modular curves

In this section we write $S_{d,\epsilon}$ for the set of isomorphism classes of $(d, \epsilon)$-structures over $\mathbb{F}_{p^2}$.

**Negation** We consider the quotient of $S_{d,\epsilon}$ by negation, which identifies $(E, \psi)$ with $-(E, \psi)$. Taking elements of $S_{d,\epsilon}$ up to negation allows to identify them with their kernel, a cyclic subgroup of order $d$. An exception has to be made for vertices with automorphism group different than $\{\pm 1\}$, i.e. $j = 0$ and 1728, but these isolated cases are easy to handle separately. Hence the quotient by the negation maps the set $S_{d,\epsilon}$ into $X_0(d)(\mathbb{F}_{p^2})$, mapping

$$S_{d,\epsilon} \ni (E, \psi) \longrightarrow (E, \ker(\psi)) \in X_0(d)(\mathbb{F}_{p^2})$$

(see Section 2.4.4 for an introduction to modular curves). It identifies $(E, \psi)$ and $-(E, \psi)$.

**Conjugation** We now consider the quotient by conjugation. The Atkin–Lehner involution $\omega_d$ defined in Section 2.4.4, which maps a modular point onto

its "dual", acts as conjugation on the image of $S_{d,\epsilon}$. It follows that the quotient by the Atkin–Lehner involution $X_0^+(d) = X_0(d)/\langle\omega_d\rangle$ identifies $\pm(E,\psi)$ and $\pm(E,\psi)^{(p)}$.

**Map from $S_{d,\epsilon}$ onto $X_0^+(d)(\mathbb{F}_p)$** We consequently obtain a four-to-one map from $S_{d,\epsilon}$ into $X_0^+(d)$, identifying the isomorphism class of $(E,\psi)$ with $-(E,\psi)$, $(E,\psi)^{(p)}$, and $-(E,\psi)^{(p)}$. Since $\{\pm(E,\psi),\pm(E,\psi)^{(p)}\}$ is stable by conjugation, it is a point in $X_0(d)(\mathbb{F}_p)$. We can therefore represent an element of $S_{d,\epsilon}$ as a point in $X_0^+(d)(\mathbb{F}_p)$ plus two bits: one to determine the sign of the isogeny, and one to set which of the two conjugate structures is encoded.

**Example 12.** To illustrate the technique above in more detail, suppose we want to compress $(5,\epsilon)$-structures over $\mathbb{F}_{p^2}$ to elements of $\mathbb{F}_p$. The classical modular polynomial of level 5 is a polynomial $\Phi_5(J_0,J_1)$ with integer coefficients, of degree 6 in $J_0$ and $J_1$. It is symmetric in $J_0$ and $J_1$, so we can write

$$\Phi_5(J_0,J_1) = -F_5(J_0+J_1, J_0J_1)$$

where

$$\begin{aligned}F_5(T,N) = {} & N^5 + 40(93T + 41650211662)N^4 \\ & + 36(126415T^2 - 2996636724991200T + 12277031464804661791632)N^3 \\ & + \cdots\end{aligned}$$

is an integer polynomial of degree 6 in $T$ and 5 in $N$. In terms of modular curves: $\Phi_5$ defines an affine model of $X_0(5)$, and the Atkin–Lehner involution on $X_0(5)$ exchanges the variables $J_0$ and $J_1$ in this model, so $F_5$ defines an affine model of $X_0^+(5)$, with the quotient map $X_0(5) \to X_0^+(5)$ defined by $(J_0,J_1) \mapsto (T,N) = (J_0+J_1, J_0J_1)$.

Now suppose we are given a $(5,\epsilon)$-structure $(E,\psi)$ over $\mathbb{F}_{p^2}$; we want to compress $(E,\psi)$ down to a single element of $\mathbb{F}_p$ plus a few bits. For simplicity, we will assume that $E$ has no extra automorphisms. First, there is an element $\gamma$ of $\mathbb{F}_{p^2}$ such that $\psi^*(\omega_{E^{(p)}}) = \gamma\omega_E$, where $\omega_E$ and $\omega_{E^{(p)}}$ are the invariant differentials on $E$ and $E^{(p)}$, respectively. Fixing a sign function on $\mathbb{F}_{p^2}$, we can encode the sign of the isogeny $\psi$ as a bit $\epsilon_1$ determining the sign of $\gamma$. Now $(E,\psi)$ is determined by $(E, \ker\psi, \epsilon_1)$.

The pair $(E,\ker\psi)$ corresponds to the point $(j(E), j(E^{(p)})) = (j(E), j(E)^p)$ on $X_0(5)$. Set $t = j(E)+j(E)^p)$ and $n = j(E)j(E)^p$, both in $\mathbb{F}_p$, and let $\epsilon_2$ be a bit determining $j(E)$ as one of the roots in $\mathbb{F}_{p^2}$ of the quadratic $X^2 - tX + n$; then $(E,\psi)$ corresponds to $(\epsilon_1,\epsilon_2,t,n)$. Now let $1 \le i \le 5$ determine the position of $n$ (in lexicographic order, say) among the (at most) 5 roots in $\mathbb{F}_p$ of the quintic $F_5(t,X)$; then $(E,\psi)$ corresponds to $(\epsilon_1,\epsilon_2,i,t)$.

Working in the other direction: given $(\epsilon_1,\epsilon_2,i,t)$, we compute the roots of $F_5(t,X)$ in $\mathbb{F}_p$, sort them, and let $n$ be the $i$-th one; then we use $\epsilon_2$ to choose a root $\alpha$ of $X^2 - tX + n$; then we construct a curve $\tilde{E}$ with $j(\tilde{E}) = \alpha$, and recover a 5-isogeny $\tilde{\psi} : \tilde{E} \to \tilde{E}^{(p)}$ using Elkies' algorithm, for example (see [Sch95, §7

and §8]). We use $\epsilon_1$ to correct the sign of $\tilde{\psi}$ if required by looking at the action on invariant differentials.

## 5.6 Parametrization

We can also find a compact representation of the modular curve $X_0^+(d)(\mathbb{F}_p)$. If $X_0^+(d)$ is a curve, it can be viewed as a cover of $\mathbb{P}^1$, allowing us to further compress the representative point in $X_0^+(d)(\mathbb{F}_p)$ to one element of $\mathbb{F}_p$ plus a few bits. This step depends strongly on the geometry of $X_0^+(d)(\mathbb{F}_p)$: for example, if $X_0^+(d)$ has genus 0 then we can rationally parametrize it, giving a simple compression of points in $X_0^+(d)(\mathbb{F}_p)$ to single elements of $\mathbb{F}_p$; if $X_0^+(d)$ is hyperelliptic, then we can compress points in $X_0^+(d)(\mathbb{F}_p)$ to a single element of $\mathbb{F}_p$ plus a "sign" bit in the usual way. As the gonality of $X_0^+(d)$ increases, so does the number of auxiliary bits required.

Useful explicit constructions for $d = 2, 3, 5$ and 7 appear in [Smi16], derived from explicit parametrizations of $\mathbb{Q}$-curves due to Hasegawa [Has97]. We reproduce these parametrizations below.

Let $E$ be an elliptic curve over a quadratic field $k$. We say $E$ is a *quadratic $\mathbb{Q}$-curve* of degree $d$ if $E$ is $d$-isogenous to its Galois conjugate with respect to $k/\mathbb{Q}$. Note that if $p$ is an inert prime in $k$, then the good reduction of $E$ modulo $p$ has a $d$-isogeny to its $p$-conjugate, and hence we get a $(d, \epsilon)$-structure.

It is known that any $\mathbb{Q}$-curve of degree $d$ without complex multiplication defined over a quadratic field corresponds to a point of $X_0^+(d)(\mathbb{Q})$ (see [GLQ04] Section 2). Having a map from $S_{d,\epsilon}$ onto $X_0^+(d)(\mathbb{F}_p)$, and a parametrization of $\mathbb{Q}$-curves due to Hasegawa, we obtain a parametrization of elements in $S_{d,\epsilon}$ up to negation and conjugation. We have already used these parametrizations in our examples in the previous chapter, namely in Figures 5.1, 5.2 and 5.3.

### 5.6.1 Representing $(2, \epsilon)$-structures

**Proposition 39.** *Let $\Delta$ be a nonsquare in $\mathbb{F}_p$, and fix a square root $\sqrt{\Delta}$ in $\mathbb{F}_{p^2}$. Let $\epsilon = -(-2/p)$. The following map gives a parametrization of $(2, \epsilon)$-structures:*

$$\begin{array}{ccc} \mathbb{F}_p & \longrightarrow & \mathcal{D}_{2,\epsilon}(p) \\ u & \longrightarrow & (E_{2,u}, \psi_{2,u}) \end{array}$$

*with*

$$E_{2,u}/\mathbb{F}_{p^2} : y^2 = x^3 - 6(5 - 3u\sqrt{\Delta})x + 8(7 - 9u\sqrt{\Delta})$$

$$\psi_{2,u} : (x, y) \longmapsto \left( \frac{-x}{2} - \frac{9(1 + u\sqrt{\Delta})}{x - 4}, \frac{y}{\sqrt{-2}} \left( \frac{-1}{2} + \frac{9(1 + u\sqrt{\Delta})}{(x-4)^2} \right) \right).$$

*Proof.* [Smi16] Let $\Delta$ be a nonsquare in $\mathbb{F}_p$, and fix a square root $\sqrt{\Delta}$ in $\mathbb{F}_{p^2}$. For each $u$ in $\mathbb{F}_p$, the curve

$$E_{2,u}/\mathbb{F}_{p^2} : y^2 = x^3 - 6(5 - 3u\sqrt{\Delta})x + 8(7 - 9u\sqrt{\Delta})$$

has a rational 2-torsion point $(4,0)$, which generates the kernel of a 2-isogeny $\psi_{2,u} : E_{2,u} \to {E_{2,u}}^{(p)}$ defined over $\mathbb{F}_{p^2}$. If we use Vélu's formulae to compute the (normalized) quotient isogeny $E_{2,u} \to E_{2,u}/\langle(4,0)\rangle$, then the isomorphism $E_{2,u}/\langle(4,0)\rangle \to {E_{2,u}}^{(p)}$ is $\tau_{1/\sqrt{-2}}$. Composing, we obtain an expression for $\psi_{2,u}$ as a rational map:

$$\psi_{2,u} : (x,y) \longmapsto \left( \frac{-x}{2} - \frac{9(1+u\sqrt{\Delta})}{x-4}, \frac{y}{\sqrt{-2}} \left( \frac{-1}{2} + \frac{9(1+u\sqrt{\Delta})}{(x-4)^2} \right) \right).$$

At this point we can either directly compute the dual isogeny $\widehat{\psi_{2,u}}$ and compare it with ${\psi_{2,u}}^{(p)}$, or we can compose ${\psi_{2,u}}^{(p)}$ with $\psi_{2,u}$ and compare the result with $[2]_{E_{2,u}}$. Either way, we find that $(E_{2,u}, \psi_{2,u})$ is a $(2, -(-2/p))$-structure, that is, $(E_{2,u}, \psi_{2,u})$ is a $(2,1)$-structure if $p \equiv 5,7 \pmod 8$, or a $(2,-1)$-structure if $p \equiv 1,3 \pmod 8$. (To obtain a family of $(2,-1)$-structures when $p \equiv 5,7 \pmod 8$ or $(2,1)$-structures if $p \equiv 1,3 \pmod 8$, it suffices to take the quadratic twist.) □

When $\epsilon$ is the opposite sign of the one wanted, one can simply take the quadratic twist of the curve given by the parametrization. Similarly, when the isogeny wanted is the one with opposite sign, then one takes $(E_{2,u}, -\psi_{2,u})$.

### 5.6.2 Representing $(3,\epsilon)$-structures

**Proposition 40.** *Let $\Delta$ be a nonsquare in $\mathbb{F}_p$, and fix a square root $\sqrt{\Delta}$ in $\mathbb{F}_{p^2}$. Let $\epsilon = -(-3/p)$. The following map gives a parametrization of $(3,\epsilon)$-structures:*

$$\begin{array}{ccc} \mathbb{F}_p & \longrightarrow & \mathcal{D}_{3,\epsilon}(p) \\ u & \longrightarrow & (E_{3,u}, \psi_{3,u}) \end{array}$$

*with*

$$E_{3,u}/\mathbb{F}_{p^2} : y^2 = x^3 - 3\big(5 + 4u\sqrt{\Delta}\big)x + 2\big(2u^2\Delta + 14u\sqrt{\Delta} + 11\big)$$

*the kernel polynomial of $\psi_{3,u}$ being*

$$\chi(\psi_{3,u}) = x - 3$$

*Proof.* [Smi16] Let $\Delta$ be a nonsquare in $\mathbb{F}_p$, and fix a square root $\sqrt{\Delta}$ in $\mathbb{F}_{p^2}$. For each $u$ in $\mathbb{F}_p$, the elliptic curve

$$E_{3,u}/\mathbb{F}_{p^2} : y^2 = x^3 - 3\big(5 + 4u\sqrt{\Delta}\big)x + 2\big(2u^2\Delta + 14u\sqrt{\Delta} + 11\big)$$

has a subgroup of order 3 defined by the polynomial $x-3$, consisting of the points $\mathcal{O}$ and $(3, \pm 2(1-u\sqrt{\Delta}))$. Taking the quotient with Vélu's formulae and composing with $\tau_{1/\sqrt{-3}}$ yields an explicit 3-isogeny $\psi_{3,u} : E_{3,u} \to {E_{3,u}}^{(p)}$, and we find that $(E_{3,u}, \psi_{3,u})$ is a $(3, -(-3/p))$-structure, that is, $(E_{3,u}, \psi_{3,u})$ is a $(3,1)$-structure if $p \equiv 2 \pmod 3$, or a $(3,-1)$-structure if $p \equiv 1 \pmod 3$. (To obtain a family of $(3,-1)$-structures when $p \equiv 2 \pmod 3$ or $(3,1)$-structures if $p \equiv 1 \pmod 3$, take the quadratic twist.) □

When $\epsilon$ is the opposite sign of the one wanted, one can simply take the quadratic twist of the curve given by the parametrization. Similarly, when the isogeny wanted is the one with opposite sign, then one takes $(E_{3,u}, -\psi_{3,u})$.

### 5.6.3 Representing $(5, \epsilon)$-structures

For $d = 5$, there exists a family of $\mathbb{Q}$-curves of degree 5 for every prime $p \equiv 3 \pmod 4$.

**Proposition 41.** *Let $p$ be a prime such that $p \equiv 3 \pmod 4$. The following map gives a parametrization of $(5, 1)$-structures:*

$$\begin{array}{ccc} \mathbb{F}_p & \longrightarrow & \mathcal{D}_{5,1}(p) \\ u & \longrightarrow & (E_{5,u}, \psi_{5,u}) \end{array}$$

*with*

$$E_{5,u}/\mathbb{F}_{p^2} : y^2 = x^3 + 3A(u)x + B(u)$$

*where*

$$A(u) = -27u(11u-2)(3(6u^2+6u-1) - 20u(u-1)\sqrt{-1}),$$

$$B(u) = 54u^2(11u-2)^2((13u^2+59u-9) - 2(u-1)(20u+9)\sqrt{-1})$$

*and the kernel polynomial of $\psi_{5,u}$ being*

$$\chi(\psi_{5,u}) = (1+2\sqrt{-1})(x - 3u(11u-2)(2-\sqrt{-1}))^2 + 81u(11u-2)(1+u\sqrt{-1})^2$$

*Proof.* The proof is similar to that of Proposition 39 and Proposition 40. See [Smi16]. □

For primes $p$ such that $p \equiv 1 \pmod 4$, Hasegawa gives in Proposition 2.3 of [Has97] a methodology to find a quadratic $\mathbb{Q}$-curve of degree 5, provided we have an element $\Delta$ such that $(5/p_i) = 1$ for every prime $p_i \neq 5$ dividing $\Delta$.

### 5.6.4 Representing $(7, \epsilon)$-structures

For $d = 7$, there exists a family of $\mathbb{Q}$-curves of degree 7 for every prime $p \equiv 3 \pmod 4$.

**Proposition 42.** *Let $\Delta$ be a nonsquare in $\mathbb{F}_p$, and fix a square root $\sqrt{\Delta}$ in $\mathbb{F}_{p^2}$. Let $\epsilon = -(-7/p)$. The following map gives a parametrization of $(7, \epsilon)$-structures:*

$$\begin{array}{ccc} \mathbb{F}_p & \longrightarrow & \mathcal{D}_{7,\epsilon}(p) \\ u & \longrightarrow & (E_{7,u}, \psi_{7,u}) \end{array}$$

*with*

$$E_{7,u}/\mathbb{F}_{p^2} : y^2 = x^3 + 3A(u)x + B(u)$$

*where*

$$A(u) = -3C(u)(85 + 96u\sqrt{\Delta} + 15u^2\Delta),$$

$$B(u) = 14C(u)(9(3u^4\Delta^2 + 130u^2\Delta + 171) + 16(9u^2\Delta + 163)u\sqrt{\Delta})$$
$$C(u) = 7(27 + u^2\Delta)$$

*and*

$$\chi(\psi_{7,u}) = (x - C(u))^3 - 42(1 - u\sqrt{\Delta})^2 C(u)[3(x - C(u)) + 4(1 - u\sqrt{\Delta})(27 + u\sqrt{\Delta})]$$

*Proof.* The proof is similar to that of Proposition 39 and Proposition 40. See [Smi16]. □

# Chapter 6

# HD CSIDH: Higher degree commutative supersingular Diffie–Hellman

**Abstract** In Chapter 5 we introduced a generalization of the CSIDH group action. We extended it from elliptic curves defined over $\mathbb{F}_p$ to any elliptic curve having a degree $d$-isogeny to their conjugate, named as $(d, \epsilon)$-structures, CSIDH being the case $d = 1$. Having a generalization of the group action, we want to build a key exchange protocol using $(d, \epsilon)$-structures. In this chapter, we detail the derived key exchange protocol HD CSIDH, study the underlying security problems and the practical computation, as well as key compression and key validation procedures.

The results of this section have been published in [CS21].

## 6.1 HD CSIDH: Higher degree CSIDH

We use the same notations as in Chapter 5: let $k = \mathbb{Q}(\sqrt{-dp})$ with $\mathcal{O}_k$ the maximal order of $k$, let $\mathcal{D}^{\max}_{d,\epsilon}(p)$ be the set of isomorphism classes of supersingular $(d, \epsilon)$-structure over $\mathbb{F}_{p^2}$ whose natural orientation are primitive $\mathcal{O}_k$-orientations, and $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ the set of classes whose natural orientations are primitive orientations by the order of conductor 2 in $\mathcal{O}_k$.

The action of $\mathrm{Cl}(\mathcal{O}_k)$ on $\mathcal{D}^{\max}_{d,\epsilon}(p)$ and $\mathrm{Cl}(\mathbb{Z}[\sqrt{-dp}])$ on $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ makes the graph $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ a natural candidate for HHS-based post-quantum cryptosystems following Stolbunov [RS06, Sto09, Sto10] and Couveignes [Cou06]. For each $d > 1$, we can define a key exchange algorithm on $\mathcal{D}_{d,\epsilon}(p)$ generalizing CSIDH [CLM+18] and CSURF [CD20], which use respectively the action of $\mathrm{Cl}(\mathbb{Z}[\sqrt{-p}])$ on $\mathcal{D}^{\text{sub}}_{1,1}(p)$ and $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$ on $\mathcal{D}^{\max}_{1,1}(p)$.

Despite the prominence of orientations, the relationship between key exchange in $\mathcal{D}_{d,\epsilon}(p)$ and the "oriented-SIDH" OSIDH protocol [CK20] is distant.

The $\mathcal{O}$-orientations in OSIDH involve orders $\mathcal{O}$ with massive conductors in $\mathcal{O}_k$ where $\mathcal{O}_k$ has tiny class number; here, $\mathcal{O}$ has tiny conductor and $\mathcal{O}_k$ has massive class number. In OSIDH, the path is a *descending path* within the graph starting from a curve with an $\mathcal{O}$-orientation where $\mathcal{O}$ is an order of large conductor in $\mathcal{O}_k$, and $\#\operatorname{Cl}(\mathcal{O}_k) = 1$. In our case, we use curves with an $\mathcal{O}$-orientation where $\mathcal{O}$ has tiny conductor in $\mathcal{O}_k$, and $\#\operatorname{Cl}(\mathcal{O}_k)$ is huge.

### 6.1.1 Hard problems

The conjectural hard problems for the action of $\operatorname{Cl}(\mathcal{O}_k)$ on $\mathcal{D}_{d,\epsilon}(p)$ are vectorization and parallelization from Couveignes' *Hard Homogenous Spaces* framework [Cou06]. We describe the instances of Vectorization and Parallelization from Definition 3 in this special context:

**Vectorization** Given $(E, \psi)$ and $(E', \psi')$ in $\mathcal{D}_{d,\epsilon}(p)$, find $\mathfrak{a} \in \operatorname{Cl}(\mathcal{O}_k)$ such that $\mathfrak{a} \cdot (E, \psi) = (E', \psi')$.

**Parallelization** Given $(E_0, \psi_0)$, $(E_1, \psi_1)$, and $(E_2, \psi_2)$ in $\mathcal{D}_{d,\epsilon}(p)$, compute the unique $(E_3, \psi_3)$ in $\mathcal{D}_{d,\epsilon}(p)$ such that $(E_3, \psi_3) = (\mathfrak{a}_1\mathfrak{a}_2) \cdot (E_0, \psi_0)$ where $(E_i, \psi_i) = \mathfrak{a}_i \cdot (E_0, \psi_0)$ for $i = 1$ and 2.

Solving Vectorization immediately solves Parallelization. In the opposite direction, no classical reduction is known in the abstract HHS framework or in the concrete world of isogenies. The quantum equivalence of these two problems is shown in [GPSV18].

An extensive study of the possible classical and quantum attacks on Vectorization for $d = 1$ can be found in [CLM+18]. All of these attacks extend to $d > 1$ with a slowdown at most polynomial in $d$ for class groups of the same size, with that slowdown due to the more complicated isogeny evaluation and comparison algorithms involved in working with $(d, \epsilon)$-structures instead of plain elliptic curves.

The best classical attack known on Vectorization is to use random walks in $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ (exactly as in the $d = 1$ case in [DG16]), which gives a solution after an expected $O((dp)^{1/4})$ isogeny steps.

Since vectorization is an instance of the Abelian Hidden Shift Problem, the best quantum attack is Kuperberg's algorithm [Kup05, Reg04, Kup13] using the Childs–Jao–Soukharev quantum isogeny-evaluation algorithm as a subroutine [CJS14], adapted to push the $d$-isogeny $\psi$ to the conjugate through the $\ell$-isogenies. This adaptation may incur a practically significant but asymptotically negligible cost; the result is a subexponential algorithm running in time $L_{dp}[1/2, \sqrt{2}]$. Even for $d = 1$, there is some debate as to the concrete cost of this quantum algorithm, and the size of $p$ required to provide a cryptographically hard problem instance for common security levels [BLMP19, BS20, Pei20]. (If and) when some consensus forms on secure parameter sizes for CSIDH, the

same parameter sizes should make Vectorization and Parallelization in $\mathcal{D}_{d,\epsilon}(p)$ cryptographically hard, too.

We argue that using $d > 1$ instead of $d = 1$ does not dramatically affect the security, but that it rather increase it asymptotically. When we compare $\mathbb{Q}(\sqrt{-dp})$ with $\mathbb{Q}(\sqrt{-p})$, the class number grows by $\frac{1}{2}\log(d)$ bits asymptotically as mentioned in Proposition 3. Using $d > 1$ instead of $d = 1$ hence leads asymptotically to using a larger class group, and to more security. However we need small $d$ ($< \log(p)$) for efficiency reasons, so if there is a net gain in security it is only modest.

However these are asymptotics that do not apply for the sizes of $p$ and $d$ that would be used in cryptographic practice. For these kinds of small $d$, and concrete $p$, computing the two class numbers $\#\,\mathrm{Cl}(\sqrt{-dp})$ and $\#\,\mathrm{Cl}(\sqrt{-p})$ and comparing them is the only way to tell if there is more or less security between CSIDH and HD CSIDH. This idea is illustrated in Example 13 in Section 6.3.

#### 6.1.1.1 Impact of involutions

We consider the impact of the various involutions existing in $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ exhibited in Section 5.3.3 on the security analysis, again comparing with CSIDH, which is the case $d = 1$.

The negation involution already exists for $d = 1$, where it essentially flips between a curve and its quadratic twist over $\mathbb{F}_p$. This involution has not yet been exploited to give an interesting speed-up in solving vectorization or parallelization in the case $d = 1$; a speed-up for any $d$ would be an interesting result.

For $d > 1$, however, there is at least one new involution: namely, conjugation. We note that solving vectorization modulo conjugation solves vectorization, because a vertex and its conjugate are always connected by the action of an ideal of norm $d$. Working modulo conjugation allows us to shrink search spaces by a factor of 2, yielding a speed-up by a factor of up to $\sqrt{2}$ analogous to working modulo negation when solving the classical ECDLP (as in [BLS11]). When $d$ has $n$ prime factors, we get more involutions that would allow us to work with equivalence classes of $2^n$ vertices, shrinking the search spaces by a factor of $2^n$. Prime $d$ therefore seems the simplest and strongest case to us.

#### 6.1.1.2 Impact of crossroads

Finally, we note that if a random walk should wander into a crossroad, then we have found an isogeny to a supersingular curve with much known on its endomorphism ring. In this case, attacks analogous to that of [GPST16] should apply. But as we have seen, crossroads are vanishingly rare, the chance of randomly wandering onto one is negligible for cryptographic size $p$. Their existence should not create any weakness for schemes based on $\Gamma(\mathcal{D}_{d,\epsilon}(p))$, no more than they do for CSIDH.

### 6.1.2 HD CSIDH

We now describe the non-interactive key exchange protocol based on the class group action on $\Gamma(\mathcal{D}_{d,\epsilon}(p))$, generalizing CSIDH (the case $d = 1$). We believe that the flexibility of the class group action allows many more cryptographic applications as for CSIDH.

The public parameters are

- a prime $p$;
- a prime $d$;
- an $\epsilon$ in $\{1, -1\}$;
- a set of primes $\{\ell_i\}_{i=1}^n$ coprime to $dp$ and splitting in $\mathbb{Q}(\sqrt{-dp})$, together with a prime ideal $\mathfrak{l}_i$ above each $\ell_i$;
- a starting vertex $(E_0, \psi_0)$ in $\mathcal{D}_{d,\epsilon}(p)$ (constructed using the crossroad technique from Section 5.4, for example).

We also fix a private keyspace $\mathbb{K} \subset \mathbb{Z}^n$ of exponent vectors such that $\#\mathbb{K} \geq 2^{2\lambda}$ to provide $\lambda$ bits of security against meet-in-the-middle attacks (though smaller $\mathbb{K}$ may suffice: see [CSCDJRH20]). The prime $p$ must be large enough that vectorization and parallelization cannot be solved in fewer than $2^\lambda$ classical operations, or a comparable quantum effort.

For key generation, each user randomly samples their private key as a vector $(e_i)_{1 \leq i \leq n}$ from $\mathbb{K}$, representing the ideal class $[\mathfrak{a}] = [\prod_{i=1}^n \mathfrak{l}_i^{e_i}]$ in $\mathrm{Cl}(\mathcal{O}_k)$. Their public key is a vertex $(E, \psi)$ representing $[\mathfrak{a}] \cdot (E_0, \psi_0)$, which we can compute using the methods of Section 6.2.

For key exchange, suppose Alice and Bob have key pairs $([\mathfrak{a}], (E_A, \psi_A))$ and $([\mathfrak{b}], (E_B, \psi_B))$, respectively. Alice receives and validates $(E_B, \psi_B)$, and computes $S_{AB} = (E_{AB}, \psi_{AB}) = [\mathfrak{a}] \cdot (E_B, \psi_B)$; Bob receives and validates $(E_A, \psi_A)$, and computes $S_{BA} = (E_{BA}, \psi_{BA}) = [\mathfrak{b}] \cdot (E_A, \psi_A)$. The commutativity of the group action implies that $S_{AB} \cong S_{BA}$, so Alice and Bob have a shared secret *up to isomorphism*.

To obtain a unique shared value for cryptographic key derivation, they can take $j(E_{AB}) = j(E_{BA})$. Although this deletes the isogeny and its sign, for a general vertex the curve only has one isogeny to its conjugate (up to sign). Hence for a general vertex $(E, \psi)$, using $j(E)$ instead of the modular invariant representing the $(d, \epsilon)$-structure only loses one bit of information (the sign of the isogeny). Even if the vertex curve had two different kernels of isogenies to its conjugate, we would only be losing two bits of information by using the $j$-invariant. The advantage of this approach is that we avoid computing the possibly complicated isomorphism invariants of the isogeny $\psi$. This protocol is described in Figure 6.1.

*Remark* 6. When ideal classes represent cryptographic secrets, it is important to compute their actions in constant time. A number of techniques have been proposed for this in the context of CSIDH [MCR19, OAYT20, CCC+19, CKM+20,

**Public parameters:** A prime $p$;
A squarefree $d \in \mathbb{Z}$, $d < p$;
An $\epsilon$ in $\{1, -1\}$;
A set of primes $\{\ell_i\}_{i=1}^n$ coprimes to $dp$ and splitting in $k = \mathbb{Q}(\sqrt{-dp})$;
A prime ideal $\mathfrak{l}_i$ above each $\ell_i$;
A starting vertex $(E_0, \psi_0)$ in $\mathcal{D}_{d,\epsilon}(p)$;
A private keyspace $\mathbb{K} \subset \mathbb{Z}^n$.

| $\mathcal{A}$lice | | $\mathcal{B}$ob |
|---|---|---|
| **Private key:** $(e_i)_{i=1}^n \in \mathbb{K}$ | | **Private key:** $(e'_i)_{i=1}^n \in \mathbb{K}$ |
| **Public key:** $(E_A, \psi_A) = [\mathfrak{a}] \cdot (E_0, \psi_0)$ with $\mathfrak{a} = \prod_{i=1}^n \mathfrak{l}_i^{e_i} \in \mathrm{Cl}(\mathcal{O}_k)$ | $\xrightarrow{(E_A, \psi_A)}$ | **Public key:** $(E_B, \psi_B) = [\mathfrak{b}] \cdot (E_0, \psi_0)$ with $\mathfrak{b} = \prod_{i=1}^n \mathfrak{l}_i^{e'_i} \in \mathrm{Cl}(\mathcal{O}_k)$ |
| | $\xleftarrow{(E_B, \psi_B)}$ | |
| **Shared secret computation:** $(E_{AB}, \psi_{AB}) =$ $[\mathfrak{a}] \cdot (E_B, \psi_B)$ | | **Shared secret computation:** $(E_{BA}, \psi_{BA}) =$ $[\mathfrak{b}] \cdot (E_A, \psi_A)$ |
| **Shared secret:** $j(E_{AB})$ | | **Shared secret:** $j(E_{BA})$ |

Figure 6.1: HD CSIDH key exchange protocol.

BBC+21]. Each of these methods generalizes in a straightforward way to compute class-group actions on $(d, \epsilon)$-structures. The only real algorithmic difference when evaluating an isogeny $\phi : (E, \psi) \to (E', \psi')$ is that the isogeny $\psi$ must be pushed through $\phi$ in constant-time as well. For $d = 2$ and 3, this amounts to pushing the $x$-coordinate of a single point through the isogeny, something that is already part of constant-time CSIDH implementations. For $d > 3$ the kernel polynomial of $\psi$ can be pushed through $\phi$ using the theory of elementary symmetric functions (see Section 6.2.1).

## 6.2 Practical computation

For our computations, we can represent a $(d, \epsilon)$-structure $(E, \psi)$ as $(E, f_\psi, \alpha)$, where $f_\psi$ is the kernel polynomial of $\psi$ (that is, the monic polynomial whose roots are the $x$-coordinates of the nonzero points in $\ker \psi$) and $\alpha$ is the element such that $\psi = \tau_\alpha \circ \tilde{\psi}$, where $\tilde{\psi} : E \to E/\ker \psi$ is the normalized "Vélu" isogeny. Note that for $j \neq 0, 1728$, $\alpha$ is determined by $(E, f_\psi)$ up to sign, so we can just store a single bit to encode $\alpha$ if desired, though this complicates the resulting algorithms.

We now detail how the ideal class group action can be computed. We start with a theoretical result, before presenting two possible approaches for the computation.

**Lemma 43.** *Let $k = \mathbb{Q}(\sqrt{-dp})$ and $\mathcal{O}_k = \mathbb{Z}[\omega]$ its maximal order. Let $I$ be an (integral) ideal of $\mathcal{O}_k$ of norm $\ell$. If there exists $b \in \mathbb{Z}$ such that $b < \ell$ and*

- *$\ell \mid (dp + b^2)$ if $-dp \not\equiv 1 \pmod 4$, or*
- *$\ell \mid (b(b+1) + \frac{dp+1}{4})$ if $-dp \equiv 1 \pmod 4$*

*then $I = (\ell, b \pm \omega)$.*

*Proof.* From Proposition 2, taking into account that the norm of the ideal is prime, which implies $(a, c) = (\ell, 1)$ or $(1, \ell)$. □

*Remark* 7. Note that $b$ depends only on $p$, $d$ and $\ell$, and can be included in public parameters.

We now want to compute $(E', \psi')$, the image of $(E, \psi)$ under the action of the ideal $\mathfrak{l} = (\ell, b + \omega)$ with $\omega$ such that $\mathcal{O}_k = \mathbb{Z}[\omega]$. Following [DKS18], we consider two approaches: "Vélu" and "modular". The first one keeps track of $\psi$, while the second tracks $\ker \psi$ and is oblivious of the sign, which occasionally makes two possible $\psi$ collide for curves that have two $d$-isogenies to their conjugates.

### 6.2.1 Vélu approach

In the **"Vélu"** approach, we compute a generator $K_\ell$ of the kernel $E[\ell]$ of $\phi$. This point may only be defined over an extension $\mathbb{F}_{p^{2r}}$ of $\mathbb{F}_{p^2}$. We then compute the quotient isogeny $\phi : E \to E' := E/\langle K_\ell \rangle$ using Vélu's formulae, at a cost

of $O(\ell)$ $\mathbb{F}_{p^{2r}}$-operations, or using the algorithm of [BDLS20] in $\widetilde{O}(\sqrt{\ell})$ $\mathbb{F}_{p^{2r}}$-operations.

Finally, we push $\psi$ through $\phi$ by computing the image of its kernel subgroup and choosing the correct sign for the $d$-isogeny. If we are given an $\mathbb{F}_{p^2}$-rational generator $G$ for $\ker\psi$, then pushing $\psi$ through $\phi$ essentially costs one isogeny evaluation; otherwise, this amounts to computing symmetric functions (see paragraph below), with a cost on the order of $O(d)$ isogeny evaluations. Each evaluation costs $O(\ell)$ or $\widetilde{O}(\sqrt{\ell})$ $\mathbb{F}_{p^{2r}}$-operations. The total cost is dominated by the cost of the multiplication by the cofactor $\#E(\mathbb{F}_{p^{2r}})/\ell$ needed to find $K_\ell$: we have $\log(\#E(\mathbb{F}_{p^{2r}})/\ell) = 2r\log p$, so constructing $K_\ell$ requires $O(r^2\log p)$ operations in $\mathbb{F}_{p^2}$. The algorithm to compute a single $\ell$-isogeny step using this approach is presented in Algorithm 12.

**Algorithm 12:** ComputeOrientedNeighborVélu: For a vertex $V$ on the crater, computes its neighbour $V'$ in the direction given by $b$, i.e. the action by the ideal $(\ell, \mu + [b])$.

**Input:** $V$, $\ell$, $b$
**Output:** $V'$

$(F_d, \phi_d) \leftarrow$ `IsogenyFromKernel` $(E, \chi)$
$\mu \leftarrow \pi_p \circ \phi_d$
`// Computing the image curve by the action`
Compute $K_\ell$ in $E[\ell]$ such that $\mu(K_\ell) + [b]K_\ell = \mathcal{O}$
$\chi_\ell \leftarrow$ `KernelPolynomial(`$K_\ell$`)`
$(F_\ell, \phi_\ell) \leftarrow$ `IsogenyFromKernel(`$E$, $\chi_\ell$`)`
Assert $\Phi_d(j(F_\ell), j(F_\ell)^p) = 0$
`// Finding the` $d$`-isogeny to its conjugate`
**for** $(F_d, \tau_d)$ *in* `ComputeKernelPolynomials` *(*$F$, $d$*)* **do**
    $V' = (F, \tau_d, 1)$
    $V'' = (F, \tau_d, -1)$
    **if** `IsOrientedEllNeighbor(`$V, V', \ell$, $\chi_\ell$`)` **then**
        **return** $V'$
    **else if** `IsOrientedEllNeighbor(`$V, V'', \ell$, $\chi_\ell$`)` **then**
        **return** $V''$

**Symmetric functions** Let $S_{n,k}(X_1, ..., X_n) = \sum_{1\le j_1<j_2<...<j_k\le n} X_{j_1}...X_{j_n}$ be the elementary symmetric polynomial with $n$ variables and degree $k$. For example the elementary symmetric polynomials with 3 variables are

$$S_{3,1} = X_1 + X_2 + X_3$$
$$S_{3,2} = X_1X_2 + X_2X_3 + X_1X_3$$
$$S_{3,3} = X_1X_2X_3.$$

**Theorem 44** (Fundamental theorem of symmetric polynomials)**.** *Let $f$ be a rational function symmetric in $n$ variables on a field $\mathcal{K}$. Then there exist a*

*unique rational function $g$ on $\mathcal{K}$ such that*

$$f(X_1, ..., X_n) = g(S_{n,1}, ..., S_{n,n})$$

*with* $S_{n,k}(X_1, ..., X_n) = \sum_{1 \leq j_1 < j_2 < ... < j_k \leq n} X_{j_1} ... X_{j_n}$

*Proof.* See Chapter I, Section 2, paragraph Elementary symmetric functions in [Mac98]. □

Let $\chi_d$ be the kernel polynomial of a degree $d$-isogeny. Then

$$\chi_d(X) = \sum_{i=0}^{n} (-1)^{n-i} c_i X^i$$

with the $c_i$ satisfying $c_i = S_{n,n-i}(\alpha_1, ..., \alpha_n)$ where the $\alpha_i$ are the roots of $\chi_d$. Now let $\tau_d$ be the kernel polynomial of the image of the degree $d$-isogeny through a degree $\ell$-isogeny $\phi_\ell$. Then

$$\tau_d(X) = \sum_{i=0}^{n} (-1)^{n-i} c'_i X^i$$

with the $c_i$ satisfying $c_i = S_{n,n-i}(\phi_\ell(\alpha_1), ..., \phi_\ell(\alpha_n))$ where the $\phi_\ell(\alpha_i)$ are the roots of $\tau_d$. For every $i \in \{1, ..., n\}$, $c'_i$ is symmetric in $\alpha_1, ..., \alpha_n$. Hence there exist a function $g_i$ such that $c'_i = g_i(S_{n,1}(\alpha_1, ..., \alpha_n), ..., S_{n,n}(\alpha_1, ..., \alpha_n)) = g_i(c_1, ..., c_n)$. By computing the $g_i$, we can compute $\tau'_d$ from the coefficients of $\chi_d$ without having to find the roots of the polynomial.

### 6.2.2 Modular approach

The **"modular"** approach uses modular polynomials. It requires the preceding neighbour $E_0$ to be given to indicate the direction to be taken, by avoiding backtracking.

The motivation for this second approach is that cases where there is more than one $d$-isogeny to the conjugate of a curve are extremely rare. More precisely, when it occurs, the curve has an endomorphism of degree $d^2$. Hence its $j$-invariant is a root of the modular polynomials $\Phi_d(X, X^{(p)})$ *and* $\Phi_{d^2}(X, X)$. Because $\Phi_{d^2}(X, X)$ has degree $d^2 + 1$, there are at most $d^2 + 1$ $j$-invariants concerned, compared to $\mathcal{O}(\frac{1}{2}(\log d)\sqrt{p})$ curves in $\mathcal{D}_{d,\epsilon}(p)$. Hence, in the vast majority of cases, the $d$-isogeny to the conjugate is unique and can be recovered easily, although up to sign only. Note that in this case the class group action is not strictly free and transitive any more. In practice, if using this method leads to a curve with two $d$-isogenies to its conjugate, we can always stop and switch to the Vélu method to disambiguate.

To compute the action of an ideal $\mathfrak{l}$ on $(E, \psi)$, we compute $G = \gcd(\Phi_d(X, X^p), \Phi_\ell(j(E), X))$ (if $d = 1$, then we take $\Phi_1(X, X^p) = X^p - X$). In general $G$ has only two roots in $\mathbb{F}_{p^2}$, corresponding to the two $\ell$-neighbours. In a non-backtracking walk we can divide by $X - j(E')$, where $(E', \psi')$ is the preceding vertex, to find the next

step. Otherwise, we can distinguish between the two neighbours by examining the action of $\mu$ on the $\ell$-torsion. However, note that $\psi'$ cannot be recovered with certainty. This method is presented in Algorithm 13.

To compute $\gcd(\Phi_d(X, X^p), \Phi_\ell(j(E), X))$, we compute $F(X) := \Phi_\ell(j(E), X)$ in $O(\ell)$ $\mathbb{F}_{p^2}$-operations, and then $Y := X^p \bmod F(X)$ using the square-and-multiply algorithm in $O(\ell \log p)$ $\mathbb{F}_{p^2}$-operations. We then compute $Z := \Phi_d(X, Y) \bmod F$, and then $\gcd(Z, F)$, in $O(d^2\ell^2)$ $\mathbb{F}_{p^2}$-operations. Generally $\ell$ is polynomial in $\log p$, but typically it is even smaller, and then the dominating step is the computation of $Y$. Note that depending on the size of $d$ with respect to $p$, the dominating cost might switch between the first step and the second one.

**Algorithm 13:** ComputeOrientedNeighborModular: For a vertex $V$ on the crater, computes its neighbour $V$ in the direction given by $b$, i.e. the action by the ideal $(\ell, \mu + [b])$.

**Input:** $V$, $\ell$, $b$
**Output:** $V'$

$(F_d, \phi_d) \leftarrow$ `IsogenyFromKernel` $(E, \chi)$
$\mu \leftarrow \pi_p \circ \phi_d$
`// Computing the image curve by the action`
Compute $K_\ell$ in $E[\ell]$ such that $\mu(K_\ell) + [b]K_\ell = \mathcal{O}$
$\chi_\ell \leftarrow$ `KernelPolynomial(`$K_\ell$`)`
$(F_\ell, \phi_\ell) \leftarrow$ `IsogenyFromKernel(`$E$, $\chi_\ell$`)`
Assert $\Phi_d(j(F_\ell), j(F_\ell)^p) = 0$
`// Finding the `$d$`-isogeny to its conjugate`
**for** $(F_d, \chi'_d)$ *in* `ComputeKernelPolynomials` $(j(F), d)$ **do**
    $V' = (j(F), F, \chi'_d, 1)$
    $V'' = (j(F), F, \chi'_d, -1)$
    **if** `IsOrientedEllNeighbor(`$V, V', \ell$, $\chi_\ell$`)` **then**
        **return** $V'$
    **else if** `IsOrientedEllNeighbor(`$V, V'', \ell$, $\chi_\ell$`)` **then**
        **return** $V''$

As in the ordinary case described in [DKS18], for $\mathbb{F}_q = \mathbb{F}_{p^r}$ the Vélu approach is more efficient when $r^2 < \ell$; in particular, when $K_\ell$ is defined over $\mathbb{F}_{p^2}$. If we are free to choose $p$, then we can optimize systems that use the action of a series of small primes $\ell_i$. This can be achieved by taking $p$ such that the $\ell_i$ split in $\mathbb{Z}[\sqrt{-dp}]$ i.e. $(\Delta/\ell_i) = 1$ where $\Delta$ is $-dp$ or $-4dp$, and $p = c \cdot \prod_{i=1}^{n} \ell_i - \epsilon$ with $c$ a cofactor making $p$ prime. In the case $d = 1$, this is exactly the optimization that is key to making CSIDH practical. Choosing $d$ in a similar way allows to have $r = r' = 1$, reducing the complexities.

## 6.3 Example

We present a toy example of the HD CSIDH key exchange protocol for

$$p = 35419 = 4 \times 5 \times 7 \times 11 \times 23 - 1 \ ,$$

with $\mathbb{F}_{p^2}$ defined as $\mathbb{F}_p(t)$ with $t^2 = 2$. We consider the supersingular elliptic curves having an 11-isogeny to their conjugate, and the $\ell$-isogenies between them for $\ell \in \{5, 7\}$ (the primes 5 and 7 are both split in $\mathbb{Q}(\sqrt{-11p})$). The starting vertex is given by the curve $E_0 : y^2 = x^3 + 24260x + 22318$ and the 11-isogeny generated by the point $(15782t + 184, 13566t + 24868, 1)$ on $E_0$ which lands on ${E_0}^{(p)}$.

We follow the steps described in Figure 6.1. Alice starts by generating her public key $PKA$ from her private key $SKA = [2, 1]$. She computes two 5-isogenies and one 7-isogeny from $E_0$ using Algorithm 12 or Algorithm 13. She obtains her public key

$$PKA : (y^2 = x^3 + (26533t + 34484)x + (18638t + 9766);$$
$$\ker \varphi_{11} = \langle (16432t + 22256, 27739t + 28012) \rangle)$$

composed of the supersingular elliptic curve and the 11-isogeny with kernel $\ker \varphi_{11}$ to its conjugate. The curve has $j$-invariant $j_A = 26208t + 11691$.

Bob proceeds similarly with his private key $SKB = [0, 3]$ and obtains his public key

$$PKB : (y^2 = x^3 + (30329t + 18059)x + (22203t + 34829);$$
$$\ker \varphi_{11} = \langle (11315t + 28673, 19838t + 20559) \rangle)$$

composed of the supersingular elliptic curve and the 11-isogeny with kernel $\ker \varphi_{11}$ to its conjugate. The curve has $j$-invariant $j_B = 6864t + 31835$.

Alice then applies her private key to Bob's public key and Bob his private key to Alice's public key. They both land on the same vertex which has a representative

$$(y^2 = x^3 + (34232t + 7209)x + (3505t + 15937);$$
$$\ker \varphi_{11} = (7122t + 21835, 22925t + 30171)) \, .$$

Their shared secret is the $j$-invariant of the curve, i.e. $28267t + 8980$.

## 6.4 Public key compression

### 6.4.1 Key compression with modular curves

Suppose we are given a $(5, \epsilon)$-structure $(E, \psi)$ over $\mathbb{F}_{p^2}$; we want to compress $(E, \psi)$ down to a single element of $\mathbb{F}_p$ plus a few bits, using the ideas of Section 5.5. For simplicity, we will assume that $E$ has no extra automorphisms.

**Sign** First, there is an element $\gamma$ of $\mathbb{F}_{p^2}$ such that $\psi^*(\omega_{E^{(p)}}) = \gamma\omega_E$, where $\omega_E$ and $\omega_{E^{(p)}}$ are the invariant differentials on $E$ and $E^{(p)}$, respectively. Fixing a sign function on $\mathbb{F}_{p^2}$, we can encode the sign of the isogeny $\psi$ as a bit $\epsilon_1$ determining the sign of $\gamma$. Now $(E, \psi)$ is determined by $(E, \ker \psi, \epsilon_1)$.

**From $(d, \epsilon)$-structures to modular curves** The pair $(E, \ker \psi)$ corresponds to the point $(j(E), j(E^{(p)})) = (j(E), j(E)^p)$ on $X_0(5)$. Set $t = j(E) + j(E)^p$ and $n = j(E)j(E)^p$, both in $\mathbb{F}_p$, and let $\epsilon_2$ be a bit determining $j(E)$ as one of the roots in $\mathbb{F}_{p^2}$ of the quadratic $X^2 - tX + n$; then $(E, \psi)$ corresponds to $(\epsilon_1, \epsilon_2, t, n)$.

**Compression** Now let $1 \leq i \leq 5$ determine the position of $n$ (in lexicographic order, say) among the (at most) 5 roots in $\mathbb{F}_p$ of the quintic $F_5(t, X)$; then $(E, \psi)$ corresponds to $(\epsilon_1, \epsilon_2, i, t)$.

**Decompression** Working in the other direction: given $(\epsilon_1, \epsilon_2, i, t)$, we compute the roots of $F_5(t, X)$ in $\mathbb{F}_p$, sort them, and let $n$ be the $i$-th one; then we use $\epsilon_2$ to choose a root $\alpha$ of $X^2 - tX + n$; then we construct a curve $\tilde{E}$ with $j(\tilde{E}) = \alpha$, and recover a 5-isogeny $\tilde{\psi} : \tilde{E} \to \tilde{E}^{(p)}$ using Elkies' algorithm, for example (see [Sch95, §7 and §8]). We use $\epsilon_1$ to correct the sign of $\tilde{\psi}$ if required by looking at the action on invariant differentials.

**Size** The encoding $(E, \psi) \mapsto (\epsilon_1, \epsilon_2, i, t)$ requires $\lceil \log(p) \rceil + 5$ bits, since $1 \leq i \leq 5$ can be encoded in three bits. We see that for general $d$, the number of extra bits depends (logarithmically) on the gonality of the modular curve $X_0^+(d)$ (i.e., its degree in $N$ above). Using alternate models of modular curves may reduce this to some extent.

### 6.4.2 Key compression with parametrization

When the modular curve has genus 0, it can be rationally parametrized over $\mathbb{F}_{p^2}$ as described in Section 5.6. This lets us get down to a single element of $\mathbb{F}_p$ plus a choice of sign, as in [Smi16, §5].

**Compression** The parameters $\epsilon_1$ and $\epsilon_2$ are computed as in Section 6.4.1. Then we use the parametrization of the modular curve to encode the parameters $n$ and $t$ from Section 6.4.1 as a single element $u$ in $\mathbb{F}_p$. The parametrization encodes elements of $(\mathcal{D}_{d,1}(p))$ or $(\mathcal{D}_{d,1}(p))$. We add a single bit $\epsilon_0$ to indicate whether the twist of the curve needs to be taken or not.

**Decompression** Working in the other direction: given $(\epsilon_0, \epsilon_1, \epsilon_2, u)$, we compute the corresponding curve and isogeny using the parametrization. Then we use $\epsilon_2$ to choose a root $\alpha$ of $X^2 - tX + n$; then we construct a curve $\tilde{E}$ with $j(\tilde{E}) = \alpha$, and recover a 5-isogeny $\tilde{\psi} : \tilde{E} \to \tilde{E}^{(p)}$ using Elkies' algorithm, for example (see [Sch95, §7 and §8]). We use $\epsilon_1$ to correct the sign of $\tilde{\psi}$ if required by looking at the action on invariant differentials. We use $\epsilon_0$ to know if the computation of the quadratic twist is needed.

**Size** The encoding $(E, \psi) \mapsto (\epsilon_0, \epsilon_1, \epsilon_2, u)$ requires $\lceil \log(p) \rceil + 3$ bits.

## 6.5 Public key validation

The public key validation procedure in CSIDH allows a secure static key exchange protocol, meaning that public keys can be reused across multiple runs of the key exchange protocol (see Section 3.4). The generalization of CSIDH described in Chapter 5 would benefit from a public key validation algorithm for the same reasons. Recall that in this case, public keys are pairs $(E, \psi)$. A public key is valid if and only if it is a $(d, \epsilon)$-structure and if $E$ is supersingular.

### 6.5.1 CSIDH versus HD CSIDH

The validation process for CSIDH from [CLM+18] turns out to be irrelevant for HD CSIDH. In CSIDH, i.e. for the case $d = 1$ and $p \equiv 3 \pmod 8$ with $p > 5$, an element $A \in \mathbb{F}_p$ is a valid public key if the Montgomery curve defined by $y^2 = x^3 + Ax^2 + x$ is supersingular and if its endomorphism ring over $\mathbb{F}_p$ $\mathrm{End}_p(E)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$. The validation process in CSIDH uses Proposition 45 and Proposition 46 below.

**Proposition 45.** *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 $\mathrm{End}_p(E) = \mathbb{Z}[\sqrt{-p}]$ 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.*

*Proof.* See Proposition 7 in [CLM+18]. □

Hence when the curve $y^2 = x^3 + Ax^2 + x$ is proven to be supersingular, the form of the endomorphism ring over $\mathbb{F}_p$ immediately follows.

**Proposition 46** ([CLM+18])**.** *Let $p = 4\prod_{i=0}^{n} \ell_i - 1$, $p \geq 5$ and let $E$ be an elliptic curve defined over $\mathbb{F}_p$. If there exists a point of order $\prod_{j\in\mathcal{J}} \ell_j$ greater than $4\sqrt{p}$ for $\mathcal{J}$ a subset of $\{1, ..., n\}$ then $E$ is supersingular.*

*Proof.* As $p \geq 5$, an elliptic curve $E$ defined over $\mathbb{F}_p$ is supersingular if and only if $\#E(\mathbb{F}_p) = p + 1$. The existence of a point of order $d > 4\sqrt{p}$ implies that there exists only one multiple of $d$ in the Hasse interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$. Besides, since $d = \prod_{j\in\mathcal{J}} \ell_j$, this multiple must be $p + 1$ by Lagrange's theorem. □

To validate a public key in CSIDH, following the propositions above, a point on the curve is sampled, and its order is computed. If it is greater than $4\sqrt{p}$, then the curve is supersingular (Proposition 46) and the key is valid (Proposition 45). Otherwise the process is repeated until a suitable point is found. In the case a repeated failure, the curve is ordinary.

However this technique for supersingularity proving does not extend to the case $d$ greater than 1, because the curves are defined over $\mathbb{F}_{p^2}$. Indeed in this case an adaptation of the proof from Proposition 46 would require to check if $E/\mathbb{F}_{p^2}$ has a point of order at least $4p$ to be able to use the result on the Hasse interval. But our valid curves have $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p + \epsilon)\mathbb{Z})^2$, and therefore no

points with order greater than $4p$. Hence the same methodology cannot be applied.

Instead we proceed in two steps: we first check that the HD CSIDH public key $(E, \psi)$ is a $(d, \epsilon)$-structure, then we prove the supersingularity of $E$ using an adapted and faster version of Sutherland's supersingularity test tailored for HD CSIDH specific context [Sut12]. We now describe these two steps in detail.

### 6.5.2 Checking $(d, \epsilon)$-structures

To check that $(E, \psi)$ is a $(d, \epsilon)$-structure, we first verify that $\psi$ is indeed an isogeny from $E$ to $E^{(p)}$ and then that $\widehat{\psi} = \epsilon\psi^{(p)}$. If this is the case then the key is valid, otherwise it is to be rejected. This can be checked with two $d$-isogeny computations, one for the conjugate and one for the dual. Using Vélu's formulæ, it costs $\mathcal{O}(d)$ curve operations when $d$ is small, and $\mathcal{O}(\sqrt{d})$ curve operations for larger $d$ (see [BDLS20]).

In the case of Montgomery curves, there exist explicit formulae to compute the dual of a degree $d$-isogeny for small $d$ in [NR19]; those formulae could be generalized to other curve forms. When a key is encoded using Hasegawa parameters as in Section 6.4, there is no need to check if it is a $(d, \epsilon)$-structure, because every such parameter specifies a curve and an isogeny from a family of $(d, \epsilon)$-structures. It only remains to verify the supersingularity of the curve.

### 6.5.3 Checking supersingularity: Sutherland's algorithm

In order to verifying supersingularity for $d > 1$, we specialize the deterministic supersingularity test of Sutherland [Sut12], which we recall below in Algorithm 14. It relies on the following result.

**Proposition 47.** *Let $\pi_{p^2}$ be the Frobenius endomorphism of $E/\mathbb{F}_{p^2}$. If $E$ is ordinary, then the maximal height of the 2-isogeny volcano containing $E$ is* $\log(p) + 1$.

*Proof.* Let $\pi_E$ be the Frobenius endomorphism of $E/\mathbb{F}_{p^2}$. The discriminant of $\mathbb{Z}[\pi_E]$ is bounded by $4p^2$, so the conductor of $\mathbb{Z}[\pi_E]$ in the maximal order $\mathcal{O}_k$ is bounded by $2p$; hence, if $E$ is ordinary, then the maximal height of the 2-isogeny volcano containing $E$ is $\log(p) + 1$. □

Since the height of an ordinary volcano is bounded, Sutherland's supersingularity test takes random non-backtracking 2-isogeny walks starting from each of the three 2-isogeny neighbours of $E$. If $E$ is ordinary, then at least one of these walks will descend the 2-isogeny volcano, and will therefore terminate (with no non-backtracking step defined over $\mathbb{F}_{p^2}$) after at most $\log(p) + 1$ steps. Conversely, if no walk terminates after $\log(p) + 1$ steps, then $E$ must be supersingular.

**Algorithm 14:** IsSupersingular

**Input:** $(E, \psi)$
**Output:** True if $E$ is supersingular, False otherwise

```
Compute the set 𝒯 of 2-neighbours.
if #𝒯 ≠ 3 then
    return False
for E in 𝒯 do
    Take a 2-isogeny step to the neighbour E' of E.
    i ← 1
    while i < log(p) do
        Compute the set 𝒯' of 2-neighbours of E'.
        if 𝒯' = {E'} then
            return False
        E' ← random element in 𝒯' \ E'
        i ← i + 1
return True
```

### 6.5.4 Adaptation of Sutherland algorithm

The supersingularity testing algorithm can be optimized for $(d, \epsilon)$-structures $(E, \psi)$ by taking advantage of the information contained in $\psi$.

**Walk length** First, the walk length limit can be reduced once we know that the public key $(E, \psi)$ is a $(d, \epsilon)$-structure, using the following proposition.

**Proposition 48.** *Let $(E, \psi)$ be a $(d, \epsilon)$-structure. If $E$ is ordinary, then the length of the path from the curve to the bottom of the 2-isogeny volcano is not longer than* $\log(2\sqrt{p/d}) + 1$.

*Proof.* Let $\mu$ be the endomorphism obtained from $\psi$ as in Section 5.1. We know that the endomorphism ring of the curve considered contains $\mathbb{Z}[\mu]$. Besides $\mathbb{Z}[\pi_E] \subset \mathbb{Z}[\mu]$, and the conductor of $\mathbb{Z}[\pi_E]$ in $\mathbb{Z}[\mu]$ is the integer $|r|$ of Proposition 26, which is bounded by $2\sqrt{p/d}$. From Subsection 2.7.1 and the bound on the conductor of $\mathbb{Z}[\pi_E]$ in $\mathbb{Z}[\mu]$, the length of the path from the curve to the bottom of the volcano is not longer than $\log(2\sqrt{p/d}) + 1$. We can therefore reduce the walk length limit from $\log(p) + 1$ to $\frac{1}{2}(\log(p) - \log(d)) + 1$. □

**Direction** We can also avoid trying every direction in the first step of Sutherland algorithm, but instead choose a descending path directly from the start.

- If $-dp \not\equiv 1 \pmod 4$, then we know that the maximal order of $\mathbb{Q}(\sqrt{-dp})$ is $\mathbb{Z}[\sqrt{-dp}]$, hence the graph $\Gamma(\mathcal{D}_{2,\epsilon}(p))$ has only one level, and only horizontal isogenies. Hence we can choose a descending path in the full 2-isogeny graph by choosing the only neighbour that is not a $(d, \epsilon)$-structure.

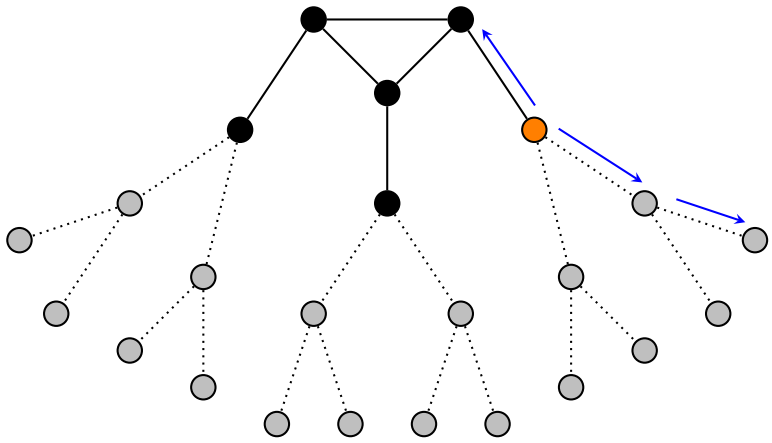


Figure 6.2: Illustration for the adaptation of Sutherland supersingularity checking algorithm. The graph above represents part of the 2-isogeny graph. The nodes in black are the curves having a 2-isogeny to their conjugate. The black edges are 2-isogenies which are also isogenies of $(2, \epsilon)$-structures. Dotted edges are regular 2-isogenies. The curve to be validated in highlighted in orange. The blue arrows represent the supersingularity checking steps. We first choose a descending direction using the structure of the graph of $(d, \epsilon)$-structures , then compute a sufficiently long path in the regular 2-isogeny graph.

- If $-dp \not\equiv 1 \pmod 4$ however, the maximal order of $\mathbb{Q}(\sqrt{-dp})$ is $\mathbb{Z}[\frac{1+\sqrt{-dp}}{2}]$, and the graph $\Gamma(\mathcal{D}_{2,\epsilon}(p))$ has two levels. Let $(E, \psi)$ be the $(d, \epsilon)$-structure considered, and $\mu$ its associated endomorphism.

  - If $(E, \psi)$ belongs to the upper level, i.e. if $\mu$ fixes the 2-torsion point wise, then it has two horizontal neighbours and one descending neighbour in $\Gamma(\mathcal{D}_{2,\epsilon}(p))$. Hence we choose the only neighbour $(E', \psi')$ whose associated endomorphism does not fix the 2-torsion (ensuring that it is on the lower level of the graph).
  - If $(E, \psi)$ belongs to the lower level, i.e. if $\mu$ does not fix the 2-torsion point wise, then it has one ascending neighbour in $\Gamma(\mathcal{D}_{2,\epsilon}(p))$, and two descending neighbours in the full 2-isogeny graph. Hence we choose any of the two neighbours not on the upper level, i.e. with associated endomorphism not fixing the 2-torsion.

Using the induced orientation in $\Gamma(\mathcal{D}_{2,\epsilon}(p))$ hence allows to choose the right path from the beginning within at most two steps in the graph, and omit the other two paths. Note that in Sutherland algorithm this is not possible without computing an orientation first.

### 6.5.5 Determining the level

If required, and only if $-dp \equiv 1 \pmod 4$, we can determine whether $(E, \psi)$ is in $\mathcal{D}^{\max}_{d,\epsilon}(p)$ or $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$ (defined in Definition 37). This is done by computing the action of $\mu$ on the 2-torsion (at the cost of one or two $d$-isogeny evaluations) or by computing the 2-neighbours of $(E, \psi)$ in $\Gamma_2(\mathcal{D}_{d,\epsilon}(p))$. Note that we have $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p+\epsilon)\mathbb{Z})^2$ from Proposition 28. Since 2 divides $(p+\epsilon)$ we obtain that $E(\mathbb{F}_{p^2})[2] = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ hence $E[2] = E[2](\mathbb{F}_{p^2})$, meaning that the full two-torsion is rational over $\mathbb{F}_{p^2}$. The procedure is described in Algorithm 15.

**Algorithm 15:** $\mathcal{D}^{\max}_{d,\epsilon}(p)$ or $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$

**Input:** $(E, \psi) \in \mathcal{D}_{d,\epsilon}(p)$
**Output:** "max" if $(E, \psi) \in \mathcal{D}^{\max}_{d,\epsilon}(p)$, "sub" if $(E, \psi) \in \mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$

```
μ ← π_p ∘ ψ
Compute the two-torsion E[2](F_{p^2}).
for P ∈ E[2] do
    if μ(P) ≠ P then
        return sub
return max
```

### 6.5.6 Validation algorithm for HD CSIDH

We now describe the full validation process for HD CSIDH public keys. To verify that a pair $(E, \psi)$ is a valid public key, we have to check that $(E, \psi)$ is a $(d, \epsilon)$-structure, then verify that $E$ is a supersingular elliptic curve. If needed we also determine if $(E, \psi)$ is in $\mathcal{D}^{\max}_{d,\epsilon}(p)$ or $\mathcal{D}^{\text{sub}}_{d,\epsilon}(p)$. To check that $(E, \psi)$ is a $(d, \epsilon)$-structure we use the results of Section 6.5.2. This step costs two $d$-isogeny computations. To verify that $E$ is a supersingular elliptic curve, we use the results of Section 6.5.3 and the adaptation of Sutherland's algorithm. This step costs $(\frac{1}{2}(\log(p) - \log(d)) + 5)$ 2-isogeny computations. We obtain the validation

algorithm described in Algorithm 16.

**Algorithm 16:** HD CSIDH public key validation

**Input:** $(E, \psi)$
**Output:** True or False

```
Check that (E, ψ) is a (d, ϵ)-structure as in Section 6.5.2
Compute the set 𝒯 of 2-neighbours of E.
Pick a descending neighbour E' as in paragraph Direction of
  Section 6.5.4.
i ← 1
while i < (½(log(p) − log(d))) do
    Compute the set 𝒯' of 2-neighbours of E'.
    if there exist E'' in 𝒯' such that E'' ≠ E' then
        E ← E'
        E' ← E''
        i ← i + 1
    else
        return False
return True
```

**Total cost** The total cost of the procedure is the cost of computing two $d$-isogenies and $(\frac{1}{2}(\log(p) - \log(d)) + 5)$ 2-isogenies. Considering that $d$ is in $\mathcal{O}(\log p)$, and that the cost of computing the 2-isogeny is asymptotically the cost of finding the roots of a quadratic (or cubic for the first step) polynomial which costs $\mathcal{O}(\log(p))$ operations in $\mathbb{F}_p$, we obtain a total asymptotic complexity of $\mathcal{O}(\log(p)^2)$ operations in $\mathbb{F}_p$.

### 6.5.7 CSIDH and HD CSIDH validation comparison

We give a comparison between CSIDH and HD CSIDH validation process in Table 6.1. It details the parameters on which the complexity depends, the complexity itself, and the main steps of the validation.

| Scheme | Process | Complexity |
|---|---|---|
| CSIDH | Supersingularity checking in $\mathbb{F}_p$ | $\mathcal{O}(\frac{\log(p)\log(n)}{\log(\log n)})$ **F** |
| HD CSIDH | Element of $\mathcal{D}_{d,\epsilon}(p)$ [1] | $\mathcal{O}(\log d)$ **F** [1] (or $\mathcal{O}(\log \sqrt{d})$ **F**) |
| | Supersingularity checking using special structure | $\mathcal{O}(\log(p)^2)$ **F** |
| HD CSIDH with Hasegawa | Supersingularity checking using special structure | $\mathcal{O}(\log(p)^2)$ **F** |

[1] Not needed when using Hasegawa parameters.

Table 6.1: Comparison for validations, where **F** stands for the cost of a multiplication in $\mathbb{F}_{p^2}$.

# Part IV

# Cryptanalysis

# Chapter 7

# Cryptanalysis for SIDH

**Abstract** To address the *general supersingular isogeny problem* of finding a path in the graph of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, Delfs and Galbraith use in [DG16] the action of the ideal class group on the subset of supersingular curves defined over $\mathbb{F}_p$. Since we proved in Section 5.2 the existence of free and transitive actions on other subsets of supersingular elliptic curves, we study how the Delfs–Galbraith algorithm can be generalized. We provide a generalized algorithm using several subsets at once. We study its complexity, the set of relevant parameters to be chosen, and measure the improvement offered by this enlarged approach. Finally, we focus on the parameters in SIDH and SIKE. We highlight a set of weak public keys, and propose a combination of the Van Oorschot and Wiener attack with the generalized Delfs–Galbraith algorithm.

The results of this section have been published in [CS21].

## 7.1 The Delfs–Galbraith algorithm

### 7.1.1 The general supersingular isogeny problem

**Definition 39** (The general supersingular isogeny problem)**.** Given two supersingular elliptic curves $E_1$ and $E_2$ defined over $\mathbb{F}_{p^2}$, compute an isogeny $\phi : E_1 \mapsto E_2$.

This problem is believed to be hard in the sense that the best classical (resp. quantum) algorithms to solve it have exponential (resp. subexponential) complexities in the size of the underlying finite field. In [DG16], Delfs and Galbraith proposed an algorithm to solve the general supersingular isogeny problem. It takes advantage of the free and transitive action of the ideal class group $\mathrm{Cl}(\sqrt{-p})$ on the subset of supersingular curves defined over $\mathbb{F}_p$.

Let $S_{p^2}$ be the set of supersingular curves over $\mathbb{F}_{p^2}$, up to isomorphism, and $S_p$ the subset of curves defined over $\mathbb{F}_{p^2}$. Let $E_1$ and $E_2$ be two curves in $S_{p^2}$.

In order to find an isogeny between $E_1$ and $E_2$, the Delfs–Galbraith algorithm has two phases:

The first phase computes a random non-backtracking isogeny walk from $E_1$ (resp. $E_2$) until landing on a curve $E_1'$ (resp. $E_2'$) in $S_p$. These walks yield isogenies $\phi_1 : E_1 \to E_1'$ and $\phi_2 : E_2 \to E_2'$. The isogeny graph on $S_{p^2}$ has excellent mixing properties, and since $\#S_{p^2} \approx p/12$ and $\#S_p = \mathcal{O}(\sqrt{p})$, this first phase takes an expected $\mathcal{O}(\sqrt{p})$ random isogeny steps.

The second phase finds an isogeny $\phi' : E_1' \to E_2'$ using the action of $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$ on $S_p$. It starts by selecting a set of primes $\mathcal{L}$ such that the $\mathcal{L}$-isogeny graph is connected, ensuring that a path between $E_1'$ and $E_2'$ exists. Under the Generalized Riemann Hypothesis, $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$ is generated by the set $\mathcal{L}$ of ideals of prime norm up to $6\log(|\Delta|)^2$, where $\Delta$ is the discriminant of $\mathbb{Q}(\sqrt{-p})$ (see [Bac84]) though in practice we do not need so many primes [DG16]. The $\mathcal{L}$-isogeny graph on $S_p$ is therefore connected, and we can use random walks in this subgraph to construct $\phi' : E_1' \to E_2'$. By the birthday paradox, this phase takes an expected $\mathcal{O}(\sqrt[4]{p})$ random steps before finding the collision yielding $\phi'$.

In total, $\mathcal{O}(\sqrt{p})$ isogeny steps (via roots of modular polynomials, not actual isogeny evaluations) are needed to find the isogeny $\phi = \phi_1 \circ \phi' \circ \widehat{\phi_2}$ from $E_1$ to $E_2$. These two steps are detailed in Algorithms 17 and 18. The concrete computation uses roots of modular polynomials in $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$.

**Algorithm 17:** Delfs–Galbraith path finding algorithm: Step 1

**Input:** A supersingular elliptic curve $E$ defined over $\mathbb{F}_{p^2}$, and a bound $B = 6\log(|\Delta|)^2$, where $\Delta$ is the discriminant of $\mathbb{Q}(\sqrt{-p})$.

**Output:** A supersingular elliptic curve $E'$ defined over $\mathbb{F}_p$ and path $S$ from $E$ to $E'$.

$j \leftarrow j(E)$
$S \leftarrow [\,]$
$found \leftarrow false$
$\ell \xleftarrow{R}$ prime $< B$
$\Phi_\ell \leftarrow ModularPolynomial(\ell)$
$j' \xleftarrow{R} Roots(\Phi(j, X), \mathbb{F}_{p^2})$
$Append(S, j')$
**if** $j' \in \mathbb{F}_p$ **then**
    $found \leftarrow true$
**while** *not* $found$ **do**
    $\ell \xleftarrow{R}$ prime $< B$
    $\Phi_\ell \leftarrow ModularPolynomial(\ell)$
    $j'' \xleftarrow{R} Roots(\Phi_\ell(j', X), \mathbb{F}_{p^2})$ with $j'' \neq j$  `// non-backtracking`
    $Append(S, j')$
    **if** $j' \in \mathbb{F}_p$ **then**
        $found \leftarrow true$
    $j \leftarrow j'$
    $j' \leftarrow j''$
**return** $(j, S)$

**Algorithm 18:** Delfs–Galbraith path finding algorithm: Step 2

**Input:** Two supersingular elliptic curves $E_1'$ and $E_2'$ defined over $\mathbb{F}_p$, and a bound $B = 6\log(|\Delta|)^2$, where $\Delta$ is the discriminant of $\mathbb{Q}(\sqrt{-p})$.

**Output:** A path $S$ in the $\mathcal{L}$-isogeny graph of elliptic curves defined over $\mathbb{F}_p$ from $E_1'$ to $E_2'$.

$\mathcal{L} \leftarrow \{\text{primes } \ell < B \mid (\frac{-p}{\ell}) = 1\}$
$S \leftarrow [\,]$
Take vertical 2-isogenies (if required) so that $E_1'$ and $E_2'$ are on the surface, i.e. endomorphism ring over $\mathbb{F}_p$ is the maximal order in $\mathbb{Q}(\sqrt{-p})$).
$disjoint \leftarrow true$
**for** $i \in \{1, 2\}$ **do**
    $j_i \leftarrow j(E_i')$
    $S_i \leftarrow [j_i]$
    $\ell \xleftarrow{R} \mathcal{L}$
    $\Phi_\ell \leftarrow ModularPolynomial(\ell)$
    $j_i' \xleftarrow{R} Roots(\Phi_\ell((j_i, X), \mathbb{F}_p))$
    $Append(S_i, j_i')$
**if** $j_1' \in S_2$ **then**
    $disjoint \leftarrow false$
    $k \leftarrow Index(S_2, j_1)$
    $S \leftarrow Cat(S_1, Reverse(S_2[1, ..., k]))$
    // $S$ is the concatenation of $S_1$ with $S_2$ from first to $k$th element, taken in reverse order.
**while** $disjoint$ **do**
    **for** $i \in \{1, 2\}$ **do**
        $\ell \xleftarrow{R} \mathcal{L}$
        $\Phi_\ell \leftarrow ModularPolynomial(\ell)$
        $j_i'' \xleftarrow{R} Roots(\Phi_\ell(j_i, X), \mathbb{F}_p)$ with $j_i'' \neq j_i$  // non-backtracking
        $Append(S_i, j_i'')$
        $j_i \leftarrow j_i'$
        $j_i' \leftarrow j_i''$
    **if** $j_1' \in S_2$ **then**
        $disjoint \leftarrow false$
        $k \leftarrow Index(S_2, j_1')$
        $S \leftarrow Cat(S_1, Reverse(S_2[1, ..., k]))$
        // $S$ is the concatenation of $S_1$ with $S_2$ from first to $k$th element, taken in reverse order.
**return** $S$

## 7.2 Generalization

The Delfs–Galbraith algorithm exploits the action of $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$ on $S_p$. Note that this is a special case of the group action we described in Section 5.2, namely the action of $\mathrm{Cl}(\mathbb{Q}(\sqrt{-p}))$ on $\mathcal{D}_{1,1}(p)$.

We proved in Section 5.2 that for any squarefree $d$ there is a free and transitive action of $\mathrm{Cl}(\sqrt{-dp})$ on $\mathcal{D}_{d,\epsilon}(p)$. We can hence extend the subset of related curves used in the second step of Delfs–Galbraith algorithm [DG16]. Making the distinguished set larger allows us to reduce the number of random steps to be taken before reaching it.

We generalize the Delfs–Galbraith algorithm by replacing the distinguished subgraph $\Gamma(\mathcal{D}_{1,\epsilon}(p))$ with a union of subgraphs $\sqcup_{d\in D}\Gamma(\mathcal{D}_{d,\epsilon}(p))$ where $D$ is a set of coprime squarefree integers prime to $p$. We further require that the set $D$ is such that for all pairs $(d, d') \in D^2$, $dd'$ is squarefree, there exists a $(d, d')$-crossroad (see Section 5.4). We study the new complexity of the attack and show that it reduces the number of operations needed to solve the general supersingular isogeny problem. Throughout this chapter we write $\overline{D_{d,\epsilon}(p)}$ for the underlying set of curves of $D_{d,\epsilon}(p)$, i.e. forgetting the data of the $d$-isogeny to the conjugate.

### 7.2.1 Generalized Delfs–Galbraith algorithm

Let $E_1$ and $E_2$ be two supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Let $D$ be defined as above. In order to find an isogeny between $E_1$ and $E_2$, the algorithm has two phases.

The first phase computes a random non-backtracking isogeny walk from $E_1$ (resp. $E_2$) until we land on a curve $E_1'$ (resp. $E_2'$) in $\sqcup_{d\in D}\overline{D_{d,\epsilon}(p)}$. The membership testing can be done using (the product of) modular polynomials $\Phi_d$ for $d \in D$. These walks yield isogenies $\phi_1 : E_1 \to E_1'$ and $\phi_2 : E_2 \to E_2'$. Since $\#S_{p^2} \approx p/12$ and $\#\sqcup_{d\in D}\overline{D_{d,\epsilon}(p)} = \mathcal{O}((\sum_{d\in D}\sqrt{d})\sqrt{p})$, this first phase takes an expected $\mathcal{O}(\sqrt{p}/(\sum_{d\in D}\sqrt{d}))$ random isogeny steps.

Let $d_1$ and $d_2$ in $D$ such that $E_1' \in \overline{D_{d_1,\epsilon}(p)}$ and $E_2' \in \overline{D_{d_2,\epsilon}(p)}$. The second phase starts by computing a $(d_1, d_2)$-crossroad $E_c$. Note that this could be precomputed. It then finds an isogeny $E_1' \to E_c$ in $\overline{D_{d_1,\epsilon}(p)}$ and an isogeny $E_2' \to E_c$ in $\overline{D_{d_2,\epsilon}(p)}$ using the action of $\mathrm{Cl}(\mathbb{Q}(\sqrt{-d_1p}))$ (resp. $\mathrm{Cl}(\mathbb{Q}(\sqrt{-d_2p}))$) acting on $\mathcal{D}_{d_1,\epsilon}(p)$ (resp. $\mathcal{D}_{d_2,\epsilon}(p)$). By the birthday paradox, this phase takes an expected $\mathcal{O}(\sqrt[4]{d_1p})$ (resp. $\mathcal{O}(\sqrt[4]{d_2p})$) random steps before finding the collision yielding the path.

These two steps are detailed in Algorithms 19 and 20. In total, $\mathcal{O}(\sqrt{p}/(\sum_{d\in D}\sqrt{d}))$ isogeny steps are needed to find an isogeny $\phi$ from $E_1$ to $E_2$, reducing the asymptotic complexity by a factor $(\sum_{d\in D}\sqrt{d})$. Compared with the original algorithm, less steps are needed to reach the subset of curves considered in Phase 1, but the isogeny steps are more expensive than before because of the need to test $d$-isogeny existence. In Phase 2, more steps are required because the subset is larger, with the cost of one step growing with $d$ as well. This means that the

elements of $D$ must be quite small for this approach to be effective: polynomial in $\mathcal{O}(\log p)$ or in $\mathcal{O}(B)$, for example. In practice, we would probably work with smaller $d$.

**Computing modular polynomials** Note that the modular polynomials $\Phi_\ell \pmod p$ can be precomputed and stored. If the storage capacity is not sufficient, $\Phi_\ell(j, X) \in \mathbb{F}_q[X]$ can also be computed modulo $p$ on the fly using [Sut13].

**Algorithm 19:** Generalized Delfs–Galbraith algorithm: Step 1

**Input:** Supersingular elliptic curve $E$ over $\mathbb{F}_{p^2}$, a bound $B = 6\log(|\Delta|)^2$, where $\Delta$ is the discriminant of $\mathbb{Q}(\sqrt{-\max\{d \in D\}p})$.

**Output:** A supersingular elliptic curves $E'$ in $\overline{D_{d,\epsilon}(p)}$ and path $S$ from $E$ to $E'$.

$\mathcal{L} \leftarrow \{\text{primes } \ell < B \mid (\frac{-p}{\ell}) = 1\}$
$S \leftarrow [j(E)]$
$j \leftarrow j(E)$
`// First step`
$\ell \xleftarrow{R} \mathcal{L}$
$\Phi_\ell \leftarrow ModularPolynomial(\ell)$
$j' \xleftarrow{R} Roots(\Phi_\ell(j, X), \mathbb{F}_{p^2})$
$Append(S, j')$
**if** $j' \in \sqcup_{d \in D} \overline{D_{d,\epsilon}(p)}$ **then**
    $found \leftarrow true$
`// Other steps`
**while** *not* $found$ **do**
    $\ell \xleftarrow{R} \mathcal{L}$
    $\Phi_\ell \leftarrow ModularPolynomial(\ell)$
    $j'' \xleftarrow{R} Roots(\Phi_\ell(j', X), \mathbb{F}_{p^2})$ with $j'' \neq j$
    $Append(S, j'')$
    **if** $j'' \in \sqcup_{d \in D} \overline{D_{d,\epsilon}(p)}$ **then**
        $found \leftarrow true$
    $j \leftarrow j'$
    $j' \leftarrow j''$
**return** $(j, S)$

### 7.2.2 Choosing the set $D$

The generalized Delfs–Galbraith algorithm is not worthwhile for large $d$ or large $D$. A balance needs to be found between the benefit of shorter walks and the higher cost of testing the membership in $\mathcal{D}_{d,\epsilon}(p)$ in Phase 1 on the one side, and the cost of longer walks in $\Gamma(\mathcal{D}_{d,\epsilon}(p))$ in Phase 2 on the other side.

Asymptotically, $\#\mathcal{D}_{d,\epsilon}(p)$ is in $O((\sum_{d \in D} \sqrt{d})\sqrt{p})$, so the expected number

of steps in phase 1 is reduced by a factor of $O(\sum_{d\in D}\sqrt{d})$. However, the individual steps become more expensive: if we use modular polynomials to check membership of each $\mathcal{D}_{d,\epsilon}(p)$, then the number of $\mathbb{F}_{p^2}$-operations per step grows linearly with $\sum_{d\in D} d$, overwhelming the benefit of the shorter walks. Asymptotically, therefore, there is no benefit in taking large $d$ or large $D$ in phase 1. (For more analysis of random walks into $\overline{D_{d,\epsilon}(p)}$, in different contexts, see [EHL+20] and [CLG09].)

The generalized Delfs–Galbraith algorithm can become interesting for $D$ consisting of a few small $d$, precisely because the asymptotic $\kappa(d,p) := \#\mathcal{D}_{d,\epsilon}(p)/\#\mathcal{D}_{1,\epsilon}(p) \approx \sqrt{d}$ no longer holds. For $d < 10$, we can have $\kappa(d,p)$ substantially greater than $\sqrt{d}$ (and also substantially less than 1). Let us illustrate this idea with an example:

**Example 13.** Let $p$ be the toy SIDH-type prime $2^{52}\cdot 3^{33} - 1$. Then $\kappa(5,p) \approx 4.916$. If we can test for an isomorphism or a 5-isogeny to the conjugate faster than we can compute six 2-isogenies, then we can take $D = \{1,5\}$ and walk into $\overline{D_{1,\epsilon}(p)} \sqcup \overline{D_{5,\epsilon}(p)}$ faster than walking into $\overline{D_{1,\epsilon}(p)}$ alone. This speedup is counterbalanced by a slowdown in Phase 2, because walking in $\Gamma(\overline{D_{5,\epsilon}(p)})$ costs more, and because the walks need to be a square-root of $\kappa(5,p)$ longer, though we can work modulo conjugation to mitigate this cost.

### 7.2.3 Comparisons

Complexity comparisons between the original walk and its generalization with a set $D$ of coprime squarefree integers $d$ prime to $p$ are summarized in Table 7.1.

## 7.3 Application to SIDH/SIKE cryptanalysis

### 7.3.1 Specific case: weak public keys in SIKE$p$434

As we noted in Section 5.3, the probability of a random walk in the supersingular $\ell$-isogeny graph hitting a vertex in $\overline{D_{d,\epsilon}(p)}$ is very low. It is even lower when we consider SIDH/SIKE graphs, which cover only a very small proportion of the full isogeny graph, resembling trees of walks of short, fixed length.

Nevertheless, when we look at specific SIKE graphs, we see that they contain sections of $\Gamma_2(\mathcal{D}_{d,\epsilon}(p))$ and $\Gamma_3(\mathcal{D}_{d,\epsilon}(p))$ for various $d$. For example, let us consider the starting curve in `SIKEp434`, defined in [JAC+17] as $y^2 = x^3 + 6x^2 + x$ over $\mathbb{F}_{p^2}$ for the prime

$$p = 2^{216}\cdot 3^{137} - 1$$

from the specification SIKE$p$434. This curve has a $d$-isogeny to its conjugate for $d \in D = \{5, 13, 17, 29, 37, 41\}$ (and possibly also for much higher, but less practical values of $d$). If we consider the 2-isogeny graph, then we find that $\Gamma_2(\mathcal{D}_{d,\epsilon}(p))$ passes through the starting curve and continues down through the tree towards a public key for $d = 17$ and 41. Hence, if we can find a 2-isogeny path from a `SIKEp434` public key to a vertex in the image of $\mathcal{D}_{17,\epsilon}(p)$ or $\mathcal{D}_{41,\epsilon}(p)$,

then we have an express route to the starting curve. Such an attack succeeds in a reasonable time with only a very small probability, but it is still devastatingly effective for a tiny proportion of `SIKEp434` keys.

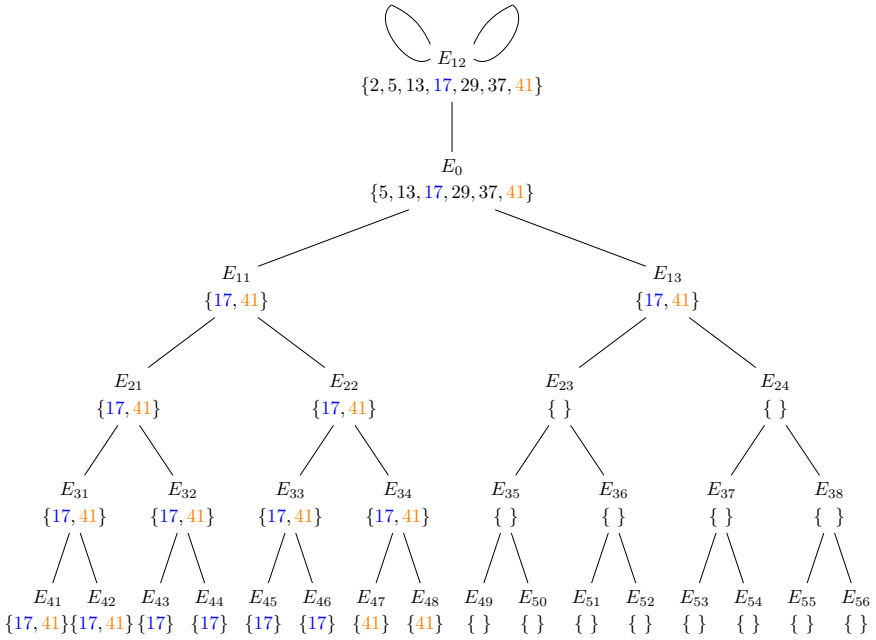



Figure 7.1: The beginning of the SIDH$p434$ 2-isogeny "tree", $E_0$ being the starting curve. Below the curves are indicated the $d < 42$ for which there exists a $d$-isogeny to the conjugate. The 17-spine, and 41-spine are highlighted in blue and orange. The curve $E_{12}$ is the curve $y^2 = x^3 + x$ which has additional automorphisms, and two endomorphisms of degree 2.

### 7.3.2 General case: SIDH, shortcut

We now consider the more general settings of SIDH. Suppose we are searching for a path from $E_1$ to $E_2$ in a SIDH graph. Unless $\log d$ is about the size of $\log p$, randomly scanning for near neighbours in $\overline{D_{d,\epsilon}(p)}$ will not be efficient. However, it can be combined with attacks against SIDH to produce occasional shortcuts in the pathfinding algorithms. The analysis of [ACC+18] suggests that the best classical attack on this problem is the van Oorschot–Wiener golden collision-finding algorithm, which computes a series of curves from $E_1$ and $E_2$ until a "golden" (essentially unique) collision is found. Scanning for curves in $\overline{D_{d,\epsilon}(p)}$ while searching for a golden collision allows us to use a possible shortcut: given partial paths $E_1 \to E_1'$ and $E_2 \to E_2'$ with $E_1'$ and $E_2'$ both in $\mathcal{D}_{d,\epsilon}(p)$ for some $d$, we can close the path between $E_1'$ and $E_2'$ in $O(\sqrt[4]{dp})$ steps using the ideal

class group action.

**Algorithm 20:** Generalized Delfs–Galbraith algorithm: Step 2

**Input:** Supersingular elliptic curves $E', E_c \in \overline{D_{d,\epsilon}(p)}$, a bound $B = 6\log(|\Delta|)^2$, where $\Delta$ is the discriminant of $\mathbb{Q}(\sqrt{-\max\{d \in D\}p})$.

**Output:** A path $S$ in $\mathcal{D}_{d,\epsilon}(p)$ from $E'$ to $E_c$.

$E_c \leftarrow$ `CurveFromjInvariant` (`FindCrossroad` $(d_1, d_2, p)$)
$\mathcal{L} \leftarrow \{\text{primes } \ell < B \mid (\frac{-p}{\ell}) = 1\}$ `// Precomputed`
$S \leftarrow [\,]$
Take vertical 2-isogenies (if required) so that $E'$ and $E_c$ are on the surface, i.e. their endomorphism ring over $\mathbb{F}_p$ is the maximal order in $\mathbb{Q}(\sqrt{-dp})$).
$j_1 \leftarrow j(E')$
$j_2 \leftarrow j(E_c)$
$S_1 \leftarrow [j_1]$
$S_2 \leftarrow [j_c]$
**for** $i \in \{1, 2\}$ **do**
    $\ell \xleftarrow{R} \mathcal{L}$
    $\Phi_\ell \leftarrow ModularPolynomial(\ell)$
    $j'_i \xleftarrow{R} Roots((j_i, X), \mathbb{F}_{p^2})$
    $Append(S_i, j'_i)$
**if** $j'_1 \in S_2$ **then**
    $disjoint \leftarrow true$
**while** $disjoint$ **do**
    **for** $i \in \{1, 2\}$ **do**
        $\ell \xleftarrow{R} prime$
        $\Phi_\ell \leftarrow ModularPolynomial(\ell)$
        $j''_i \xleftarrow{R} Roots(\Phi_\ell(j'_i, X), \mathbb{F}_p)$ with $j''_i \neq j_i$ `// non-backtracking`
        $Append(S_i, j''_i)$
        $j_i \leftarrow j'_i$
        $j'_i \leftarrow j''_i$
    **if** $j'_1 \in S_2$ **then**
        $disjoint \leftarrow false$
        $k \leftarrow Index(S_2, j'_1)$
        $S \leftarrow Cat(S_1, Reverse(S_2[1, ..., k]))$
        `//` $S$ `is the concatenation of` $S_1$ `with` $S_2$ `from first to` $k^{th}$ `element, taken in reverse order.`
**return** $S$

| | [DG16] | Generalized with a set $D$<br>$\sigma = (\sum_{d \in D} d)$<br>(Algorithms 19 and 20) |
|---|---|---|
| Distinguished subset graph<br>Size of the subgraph | $\overline{D_{1,\epsilon}(p)}$<br>$\mathcal{O}(\sqrt{p})$ | $\cup_{\mathcal{D}} \overline{D_{d,\epsilon}(p)}$<br>$\mathcal{O}(\sqrt{\sigma p})$ |
| Length of walk to the subset<br>Length of walk in the subset | $\mathcal{O}(p^{1/2})$<br>$\mathcal{O}(p^{1/4})$ | $\mathcal{O}((\frac{p}{\sigma})^{1/2})$<br>$\mathcal{O}((\sigma p)^{1/4})$ |
| Total cost | $\mathcal{O}(p^{1/2})$ | $\mathcal{O}(\frac{p}{\sigma})^{1/2})$ |

Table 7.1: Complexity comparison in term of isogeny steps between the original and the generalized Delfs–Galbraith path finding algorithms. The $d$ in the set $D$ should be chosen to be small, making the cost of testing membership in $\mathcal{D}_{d,\epsilon}(p)$ asymptotically negligible.

# Summary and perspectives

Several contributions have been presented in this thesis:

1. By presenting efficient dummy-free and derandomized implementations of CSIDH, we contributed to mitigating the relative slowness of CSIDH, as well as filling a gap between theory and practical implementations.

2. By highlighting and studying new subsets of supersingular elliptic curves having a free and transitive group action, we have contributed to a better understanding of the isogeny theoretical landscape. The generalization of CSIDH in Chapters 5 and 6 offers new alternatives and adaptability to isogeny-based key-exchange protocols, building a wider range of constructive options.

3. By proposing public key validation and compression in HD CSIDH, we reenforced the compactness and reusability of keys, two strengths of isogeny-based protocols.

4. By studying the cryptanalysis consequences on SIDH of the new free and transitive group actions, we have shown that they do not offer a significant asymptotic advantage to an attacker. In the specific case of SIKE434, the fact that only a very small subset of curves are vulnerable increases understanding of the security and confidence in the robustness of the protocol.

In the light of these contributions, we also identify several future directions to be studied. First, the parameters of the generalized Delfs–Galbraith algorithm can be fine-tuned to the specific parameters of SIDH and SIKE, in particular to find the optimal balance between the number and size of underlying parameter $d$. Then, more parametrized families for HD CSIDH in optimised curve form can be exhibited. Finally, isogeny computation algorithms in the specific case of HD CSIDH can be optimized to improve performances. The possibility to determine formulae using the Hasegawa parameters cn also be considered.

# Bibliography


[ABC+19] Diego F. Aranha, Pedro Y.S. Barbosa, Thiago N.C. Cardoso, Caio Lüders Araújo, and Paulo Matias. The return of software vulnerabilities in the Brazilian voting machine. *Computers & Security*, 86:335–349, 2019.

[ACC+18] Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes, and Francisco Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. In Carlos Cid and Michael J. Jacobson Jr., editors, *Selected Areas in Cryptography SAC*, volume 11349 of *Lecture Notes in Computer Science*, pages 322–343. Springer, 2018.

[AP13] Nadhem J. AlFardan and Kenneth G. Paterson. Lucky thirteen: Breaking the TLS and DTLS record protocols. In *Symposium on Security and Privacy, (SP) 2013*, pages 526–540. IEEE Computer Society, 2013.

[Bac84] Eric Bach. *Analytic methods in the analysis and design of number-theoretic algorithms*. Association for Computing Machinery (ACM) Distinguished Dissertation. MIT Press, 1984.

[BB03] David Brumley and Dan Boneh. Remote timing attacks are practical. In *Proceedings of the 12th USENIX Security Symposium*. USENIX Association, 2003.

[BBC+21] Gustavo Banegas, Daniel J. Bernstein, Fabio Campos, Tung Chou, Tanja Lange, Michael Meyer, Benjamin Smith, and Jana Sotáková. CTIDH: faster constant-time CSIDH. *Transactions on Cryptographic Hardware and Embedded Systems (CHES)*, 2021(4):351–387, 2021.

[BCC+13] Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Factoring RSA keys from certified smart cards: Coppersmith

in the wild. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 341–360. Springer, 2013.

[BDLS20] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. In Steven D. Galbraith, editor, *Algorithmic Number Theory Symposium (ANTS)*, volume 4 of *Open book series*, pages 39–55. Mathematical Sciences Publishers, 2020.

[BHKL13] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, and Tanja Lange. Elligator: elliptic-curve points indistinguishable from uniform random strings. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *Conference on Computer and Communications Security, CCS 2013*, pages 967–980. Association for Computing Machinery ACM, 2013.

[BJL+14] Aurélie Bauer, Eliane Jaulmes, Victor Lomné, Emmanuel Prouff, and Thomas Roche. Side-channel attack against RSA key generation algorithms. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems (CHES) 2014*. Springer, 2014.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247. Springer, 2019.

[BLMP19] Daniel J. Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny. Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019*, volume 11477 of *Lecture Notes in Computer Science*, pages 409–441. Springer, 2019.

[BLS11] Daniel J. Bernstein, Tanja Lange, and Peter Schwabe. On the correct use of the negation map in the Pollard rho method. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography (PKC) 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 128–146. Springer, 2011.

[BS20] Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020*, volume 12106 of *Lecture Notes in Computer Science*, pages 493–522. Springer, 2020.

[BV07] Johannes Buchmann and Ulrich Vollmer. *Binary Quadratic Forms: An Algorithmic Approach*. Springer, 2007.

[CCC+19] Daniel Cervantes-Vázquez, Mathilde Chenu, Jesús-Javier Chi-Domínguez, Luca De Feo, Francisco Rodríguez-Henríquez, and Benjamin Smith. Stronger and faster side-channel protections for CSIDH. In Peter Schwabe and Nicolas Thériault, editors, *LATINCRYPT 2019*, volume 11774 of *Lecture Notes in Computer Science*, pages 173–193. Springer, 2019.

[CD20] Wouter Castryck and Thomas Decru. CSIDH on the surface. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography (PQCrypto) 2020*, volume 12100 of *Lecture Notes in Computer Science*, pages 111–129. Springer, 2020.

[CH17] Craig Costello and Hüseyin Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017*, volume 10625 of *Lecture Notes in Computer Science*, pages 303–329. Springer, 2017.

[Che10] Amy Cheung. Consructing explicit isogenies using the modular curve $X_0(\ell)$. Master's thesis, University of Calgary, 2010.

[CJS14] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

[CK20] Leonardo Coló and David Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, 14(1):414–437, 2020.

[CKM+20] Fabio Campos, Matthias J. Kannwischer, Michael Meyer, Hiroshi Onuki, and Marc Stöttinger. Trouble at the CSIDH: Protecting CSIDH with dummy-operations against fault injection attacks. In *Fault Detection and Tolerance in Cryptography (FDTC) 2020*, pages 57–65. IEEE, 2020.

[CLG09] Denis X. Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *ASIACRYPT 2018*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, 2018.

[CLN16] Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016*, volume 9814 of *Lecture Notes in Computer Science*, pages 572–601. Springer, 2016.

[Cos20] Craig Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463. Springer, 2020.

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. `http://eprint.iacr.org/2006/291`.

[Cox13] David A. Cox. *Primes of the Form $x^2 + ny^2$: Fermat, Class Field Theory, and Complex Multiplication.* Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. John Wiley and Sons, 2nd edition, 2013.

[CS18] Craig Costello and Benjamin Smith. Montgomery curves and their arithmetic - the case of large characteristic fields. *Journal of Cryptographic Engineering*, 8(3):227–240, 2018.

[CS21] Mathilde Chenu and Benjamin Smith. Higher degree supersingular group actions. *Journal of Mathematical Cryptology*, 2021.

[CSCDJRH20] Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez. The SQALE of CSIDH: square-root Vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, 2020.

[DG16] Christina Delfs and Steven D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016.

[DG18] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. *EUROCRYPT 2019*, 11478:759–789, 2018.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

[DKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.

[DKS18] Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In Thomas Peyrin and Steven D. Galbraith, editors, *ASIACRYPT 2018*, volume 11274 of *Lecture Notes in Computer Science*, pages 365–394. Springer, 2018.

[DM20] Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography (PKC) 2020*, volume 12111 of *Lecture Notes in Computer Science*, pages 187–212. Springer, 2020.

[DMPS19] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019*, volume 11921 of *Lecture Notes in Computer Science*, pages 248–277. Springer, 2019.

[DPV19] Thomas Decru, Lorenz Panny, and Frederik Vercauteren. Faster seasign signatures through improved rejection sampling. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography (PQCrypto) 2019*, volume 11505 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 2019.

[dQKL+21] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Improved torsion-point attacks on SIDH variants. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021*, volume 12827 of *Lecture Notes in Computer Science*, pages 432–470. Springer, 2021.

[EHL+20] Kirsten Eisenträger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park. Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. In Steven D. Galbraith, editor, *Algorithmic Number Theory Symposium*, volume 4 of *Open book series*, pages 215–232. Mathematical Sciences Publishers, 2020.

[FLOR18] Armando Faz-Hernández, Julio César López-Hernández, Eduardo Ochoa-Jiménez, and Francisco Rodríguez-Henríquez. A faster software implementation of the supersingular isogeny Diffie-Hellman key exchange protocol. *IEEE Transaction on Computers*, 67(11):1622–1636, 2018.

[FM02] Mireille Fouquet and François Morain. Isogeny volcanoes and the SEA algorithm. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory (ANTS)*, volume 2369 of *Lecture Notes in Computer Science*, pages 276–291. Springer, 2002.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, 2013.

[Gal12] Steven D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.

[GLQ04] Josep Gonzalez, Joan-Carles Lario, and Jordi Quer. Arithmetic of $\mathbb{Q}$-curves. *Progress in Mathematics*, 224:125–139, 2004.

[GLS11] Steven D. Galbraith, Xibin Lin, and Michael Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. *Journal of Cryptology*, 24(3):446–469, 2011.

[GPST16] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91, 2016.

[GPSV18] Steven Galbraith, Lorenz Panny, Benjamin Smith, and Frederik Vercauteren. Quantum equivalence of the DLP and CDHP for group actions. *Journal of Mathematical Cryptology*, 2018.

[Has97] Yuji Hasegawa. $\mathbb{Q}$-curves over quadratic fields. *Manuscripta Mathematica*, 94(1):347–364, 1997.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography TCC 2017*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[HM89] James L. Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American Mathematical Society*, 2(4), 1989.

[JAC+17] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Aaron Hutchinson, Amir Jalali, Koray Karabina, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev, and David Urbanik. SIKE – supersingular isogeny key encapsulation, 2017. URL: `https://sike.org/`.

[JAMJ19] Amir Jalali, Reza Azarderakhsh, Mehran Mozaffari Kermani, and David Jao. Towards optimized and constant-time CSIDH on embedded devices. In Ilia Polian and Marc Stottinger, editors, *Constructive Side-Channel Analysis and Secure Design (COSADE)*, volume 11421 of *Springer Lecture Notes in Computer Science (LNCS)*, pages 215–231, 2019.

[JD11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography (PQCrypto) 2011*, pages 19–34, 2011.

[KHF+19] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. In *Symposium on Security and Privacy (SP) 2019*, pages 1–19. IEEE, 2019.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[Koh96] David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkeley, 1996.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *Society for Industrial and Applied Mathematics (SIAM), Journal of Computing*, 35(1):170–188, 2005.

[Kup13] Greg Kuperberg. Another Subexponential-time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. In Simone Severini and Fernando Brandao, editors, *Theory of Quantum Computation, Communication and Cryptography (TQC) 2013*, volume 22 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20–34. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik, 2013.

[Lan94] Serge Lang. *Algebraic Number Theory*, volume 110 of *Graduate Texts in Mathematics*. Springer, 1994.

[LH21] Jason T. LeGrow and Aaron Hutchinson. (short paper) analysis of a strong fault attack on static/ephemeral CSIDH. In Toru Nakanishi and Ryo Nojima, editors, *International Workshop on Security (IWSEC) 2021*, volume 12835 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 2021.

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[LSG+18] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown: Reading kernel memory from user space. In William Enck and Adrienne Porter Felt, editors, *USENIX Security Symposium 2018*, pages 973–990. USENIX Association, 2018.

[Mac98] I. G. Macdonald. *Symmetric functions and Hall polynomials*. Clarendon Press Oxford University Press, Oxford New York, 1998.

[MCR19] Michael Meyer, Fabio Campos, and Steffen Reith. On lions and elligators: An efficient constant-time implementation of CSIDH. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography (PQCrypto) 2019*, volume 11505 of *Lecture Notes in Computer Science*, pages 307–325. Springer, 2019.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *CRYPTO 1985*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.

[Mon87] Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48:243–234, 1987.

[NR19] Michael Naehrig and Joost Renes. Dual isogenies and their application to public-key compression for isogeny-based cryptography. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019*, volume 11922 of *Lecture Notes in Computer Science*, pages 243–272. Springer, 2019.

[OAT20] Hiroshi Onuki, Yusuke Aikawa, and Tsuyoshi Takagi. The existence of cycles in the supersingular isogeny graphs used in SIKE. In *International Symposium on Information Theory and Its Applications, (ISITA) 2020*, pages 358–362. IEEE, 2020.

[OAYT20] Hiroshi Onuki, Yusuke Aikawa, Tsutomu Yamazaki, and Tsuyoshi Takagi. A constant-time algorithm of CSIDH keeping two points. *Transactions on Fundamentals of Electronics, Communications, and Computer Science (IEICE)*, 103-A(10):1174–1182, 2020.

[OKS00] Katsuyuki Okeya, Hiroyuki Kurumatani, and Kouichi Sakurai. Elliptic curves with the Montgomery-form and their cryptographic applications. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 238–257. Springer, 2000.

[Onu21] Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields and Their Applications*, 69:101777, 2021.

[OT20] Hiroshi Onuki and Tsuyoshi Takagi. On collisions related to an ideal class of order 3 in CSIDH. In Kazumaro Aoki and Akira Kanaoka, editors, *Advances in Information and Computer Security*, pages 131–148. Springer, 2020.

[Pei20] Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020*, volume 12106 of *Lecture Notes in Computer Science*, pages 463–492. Springer, 2020.

[Pet17] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017*, volume 10625 of *Lecture Notes in Computer Science*, pages 330–353. Springer, 2017.

[Piz98] Arnold Pizer. Ramanujan graphs. *Computational perspectives on number theory 1995*, 7:159–178, 1998.

[Reg04] Oded Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. arXiv, 2004. `http://arxiv.org/abs/quant-ph/0406151`.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. `http://eprint.iacr.org/2006/145`.

[Sch95] René Schoof. Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux*, 7(1):219–254, 1995.

[Sho94] Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *Algorithmic Number Theory (ANTS)*, volume 877 of *Lecture Notes in Computer Science*, page 289. Springer, 1994.

[Sil09] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer New York, 2009.

[Smi16] Benjamin Smith. The $\mathbb{Q}$-curve construction for endomorphism-accelerated elliptic curves. *Journal of Cryptology*, 29(4):806–832, 2016.

[Sto09] Anton Stolbunov. Reductionist security arguments for public-key cryptographic schemes based on group action. In Stig F. Mjølsnes, editor, *Norsk informasjonssikkerhetskonferanse (NISK)*, 2009.

[Sto10] Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Advances in Mathematics of Communications*, 4(2), 2010.

[Sut12] Andrew V. Sutherland. Identifying supersingular elliptic curves. *LMS Journal of Computation and Mathematics*, 15:317–325, 2012.

[Sut13] Andrew Sutherland. On the evaluation of modular polynomials. In Everett W. Howe and Kiran S. Kedlaya, editors, *Algorithmic Number Theory Symposium (ANTS)*, volume 1 of *The Open Book Series*, page 531–555. Mathematical Sciences Publishers, 2013.

[Tan20] Seiichiro Tani. Quantum algorithm for finding the optimal variable ordering for binary decision diagrams. In Susanne Albers, editor, *Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 162 of *LIPIcs*, pages 36:1–36:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Tat66] John Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathematicae*, 2:134 – 144, 1966.

[TDEP21] Élise Tasso, Luca De Feo, Nadia El Mrabet, and Simon Pontié. Resistance of isogeny-based cryptographic implementations to a fault attack. In Springer Lecture Notes in Computer Science, editor, *Constructive Side-Channel Analysis and Secure Design (COSADE)*, 2021.

[Vit19] Vanessa Vitse. Simple oblivious transfer protocols compatible with supersingular isogenies. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje-eddine Rachidi, editors, *AFRICACRYPT 2019*, volume 11627 of *Lecture Notes in Computer Science*, pages 56–78. Springer, 2019.

[Voi17] John Voight. *Quaternion Algebras*, volume 288 of *Graduate Texts in Mathematics*. Springer International Publishing, 2017.

[Vé71] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes-rendu de l'académie des sciences de Paris*, 1971.

[Wat69] William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'ENS*, 2(4):521 – 560, 1969.

[Wes20] Benjamin Wesolowski. Efficient verifiable delay functions. *Journal of Cryptology*, 33(4):2113–2147, 2020.

ECOLE
DOCTORALE

**Titre :** Actions de groupe supersingulières et échages de clés post-quantiques

**Mots clés :** Cryptography post-quantique, Isogénies, Courbes elliptiques

**Résumé :** Alice et Bob souhaitent échanger des informations sans qu'un attaquant, même muni d'un ordinateur quantique, puisse les entendre. Pour cela, ils ont recours à la cryptologie et en particulier à un protocole d'échange de clés. Ces protocoles reposent sur la théorie des nombres et la géométrie algébrique. Cependant les protocoles actuellement utilisés ne résistent pas aux attaques quantiques, c'est pourquoi il est nécessaire de développer de nouveaux outils cryptographiques. L'un de ces outils repose sur les isogénies, c'est-à-dire des homomorphismes entre des courbes elliptiques. Dans cette thèse nous proposons une implémentation d'un des protocoles d'échange de clés basé sur les isogénies qui résiste aux attaques par canaux auxiliaires (étude de la durée d'exécution, de la consommation de courant et injection de fautes). Nous généralisons également ce protocole à un plus grand ensemble de courbes elliptiques.

**Title :** Supersingular Group Actions and Post-quantum Key Exchange

**Keywords :** Post-quantum cryptography, Isogenies, Elliptic curves

**Abstract :** Alice and Bob want to exchange information and make sure that an eavesdropper will not be able to listen to them, even with a quantum computer. To that aim they use cryptography and in particular a key-exchange protocol. These type of protocols rely on number theory and algebraic geometry. However current protocols are not quantum resistant, which is the reason why new cryptographic tools must be developed. One of these tools rely on isogenies, i.e. homomorphisms between elliptic curves. In this thesis the first contribution is an implementation of an isogeny-based key-exchange protocol resistant against side-channel attacks (timing and power consumption analysis, fault injection). We also generalize this protocol to a larger set of elliptic curves.

**Institut Polytechnique de Paris**
91120 Palaiseau, France