Doctoral theses at NTNU, 2023:100

Mattia Veroni

A study on tighter and more efficient isogeny-based cryptographic protocols

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of Science and Technology

Mattia Veroni

A study on tighter and more efficient isogeny-based cryptographic protocols

Thesis for the Degree of Philosophiae Doctor

Trondheim, March 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

ISBN 978-82-326-6469-6 (printed ver.)
ISBN 978-82-326-6453-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2023:100

Printed by NTNU Grafisk senter

Abstract

This PhD thesis addresses the following research questions:

- RQ 1: Can we prove tight reductions on isogeny-based schemes?
- RQ 2: How sound are the assumptions underlying some computational problems in isogeny-based cryptography?
- RQ 3: Can we obtain faster isogeny-based cryptography?

The findings and contributions of this thesis consist in five scientific papers. More specifically, this thesis presents an adaptation of Cohn-Gordon et al. [CCG+19] construction to supersingular elliptic curves over $\mathbb{F}_p$, obtaining an isogeny-based authenticated KEX protocol with an optimally tight proof. The thesis tests the reliability of certain assumptions and questions the security proof of the identification protocol based on SIDH. It also analyses the security proofs available in the literature for the SIDH-based identification protocol, together with their effects on the security of the digital signatures obtained via the Fiat-Shamir transform. A different approach to restore the security of an isogeny-based identification protocol is presented: relying on the Generalised Riemann Hypothesis, a new extractor is introduced, for which rigorous proof special-soundness property is given.

In one of the papers included in the thesis, there is a proposal of an isogeny-based signature scheme whose security relies on the computational supersingular isogeny problem. The protocol is obtained by applying the Fiat-Shamir transform to the SIDH-identification protocol, and then performing a series of optimisations both on the signature size and on the signing algorithm.

The thesis also presents a design of an algorithm to solve the constructive Deuring correspondence for general primes p, translating an ideal in the quaternion algebra ramified at p and ∞ into an isogeny. In that work several optimisations are applied for speeding up the existing algorithms that work for more general primes than the ones carefully crafted in SQISign.

Finally, the practicality of SIDH-based signatures is analysed in light of the new attacks against SIKE and the underlying KEX protocol. In particular, the last contribution shows how, despite the application of several optimisations to reduce the signature size and some minor improvements on the signing time, the design of efficient SIDH-based protocols is still an open problem.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree Philosophiae Doctor (PhD) at NTNU, Norwegian University of Science and Technology. The presented work was carried out at the Department of Information Security and Communication Technology (IIK), Trondheim, under the supervision of Professor Danilo Gligoroski and the co-supervision of Professor Colin Boyd and Professor Kristian Gjøsteen. The PhD course started in November 2018 and ended in December 2022, including one year of teaching duties.

Acknowledgements

This work would have not seen the light of day without the support, help and contribution of many people. Nonetheless, I aim for a short round of acknowledgements, since I am going to convey them analogically as soon as I will have the chance. I will owe you a home-made cake if I will not keep the promise.

First of all, I want to thank my main supervisor Danilo Gligoroski, for his constant support, enthusiasm and appreciation. We have not had many chances to work on research papers together, but I thank him for having given me the freedom of exploring my own interests and finding great collaborators in these past years.

The next in line is Kristian Gjøsteen, with whom I wrote my very first paper and who helped me kick-start this PhD during my first year, when Danilo was on sabbatical. I recommend talking to him for his capability of providing great insights on basically any cryptography-related topic, with the caveat that a chat can (and most surely will) turn into a one-hour lecture, in which much more knowledge than hoped for will be provided.

In the triad of supervisors, Colin Boyd comes last but definitely not least. I thank him for the opportunity he gave me to test and improve myself as lecturer. I have witnessed him being a great researcher and teacher, a loving husband, a caring father and an amazing human being. Without his and his wife Delyth's support, life in this last year would have been even gloomier. Thank you for begin such an admirable person.

I also want to thank Jiaxin Pan for giving me the chance to work with him as a Postdoc immediately after I will have defended this PhD thesis. Being the excellent researcher he is, I hope in a fruitful and amazing research collaboration.

Much of the work in this thesis would have not been possible, or would have had a much lower quality, without all of my co-authors. Our work together made me grow immensely, both as human being and as researcher. Thank you for sharing this scientific journey with me, and for showing me different approaches to research and collaboration.

Thanks all the office-mates for every good and bad moment. You taught me so much of you, myself and office-sharing in general. Many people have come and gone at the department, and some professional relationships transcended to friendships. I will do my best to keep you who I wholeheartedly call friends in my life, despite of geographical location, time-zone and stage in life. Because friendships have much in common with cryptographic papers: they require hard work to pass the test of time. Among many others, to Ali, Faiga, Georgios, Jana, Julie, Lea, Lise, Mayank, Murad, Pia, Shuang, Sonu, Stas, Tjerand: thank you for warming up these cold Norwegian years.

To my family and friends. This journey has been a great challenge. Distance, Covid, lockdowns and plenty of adversities often made me doubt myself and the direction my life was going. Last but not least, the end of an 8-year long relationship hit extremely hard, and made me lose track of who I was and what on Earth I was doing. Thanks for begin there, listening to my pain, showing me that life is worth living despite of all the suffering. You are my lifeline, a lighthouse that never fails to show me the way home.

And since I am always told I am too self-critical... a final thanks to myself, for not giving up.

Mattia Veroni
Trondheim, March 2023

Contents

Part I

Background and summary of contributions

Chapter 1

Introduction

Romeo and Juliet:
Had they known of encryption
They would be alive.

December 2022[1]

Despite its etymological origins in the Greek words *kryptós*, meaning "hidden" or "secret", and *graphein*, meaning "to write", the term *cryptography* has gained several different acceptations nowadays. Centuries ago, only few individuals of a certain lineage were in need of ways to secretly communicate, most often about military arrangements. Cryptography actually meant "secret writing" back then, and today we would refer to the process of "writing in secret" as *encryption* of a message. More to the point, we talk about *symmetric encryption*, since the conversion from plaintext to ciphertext and vice versa was done via the same pre-shared key. Early attempts such as the greek scytale and Caesar's cipher were rudimental and look ridiculous to our modern eyes, but they were commensurate with the threat models and tools available at that time.

Mathematical and algorithmic advances changed the field slowly but constantly, leading to the construction of very famous encryption machines such as Enigma (sadly, war was and still is a major promoter of cryptographic developments). Moving towards an interconnected world, with people who had possibly never met needing to communicate secretly, exchanging symmetric keys quickly became very expensive if not unfeasible, and new tools had to be developed. The pioneering works of Diffie and Hellman [DH76], and of Rivest, Shamir and Adleman [RSA78], opened up a whole new world in telecommunications: *public-key cryptography*, a.k.a. *asymmetric cryptography* in opposition to the techniques developed over the previous millennia.

The original goal of enabling secret communication still holds; nonetheless, over the past 40 years cryptography has been enriched with new features and flavours, most of which were unimaginable before the development of modern computers and smart devices. Among the countless applications of public-key cryptography, this thesis deals with five of them. Suppose that each user (in practice, a device) holds both a public key and a secret key, bound to each other via some hard mathematical problem. A *key exchange protocol* (KEX) allows users knowing each other's public keys to agree on a shared key that can be used for symmetric encryption/decryption. One can deploy a *public key encryption*

[1] Loving mathematics and computer science does not imply despising arts and bad jokes. As a homage to this centennial form of Japanese poetry, I have decided to start each chapter with a haiku. I hope this is not perceived as cultural appropriation and no reader takes offence in this; if only because maybe 20 people on Earth will read this thesis, which ensures a negligible probability of the aforementioned event taking place.

scheme (PKE) to encrypt outgoing messages using the partner's public key, and decrypt incoming messages using its private key. More unilaterally, a *key encapsulation mechanism* (KEM) requires the use of a partner's public key to create a ciphertext (encapsulation) containing a symmetric key chosen at random. Using an *identification protocol* (ID) a user can persuade another of its identity, which is tied to a public key, by proving knowledge of the secret key corresponding to that public key. Finally, a *digital signature scheme* (DS) allows a user to claim ownership (usually also preventing repudiation) of data.

1.1 Post-quantum cryptography

Until a few years ago, most of contemporary communication has been secured by protocols built on top of the aforementioned seminal works. The basic problems that still guarantee the security of these schemes are two: the discrete logarithm problem and the integer factorisation problem. If these two resist, by progressively increasing the key-size and applying some tweaks, online communication is safe.

In 1997, a ground-breaking result by Peter Shor [Sho97] was published, which included polynomial-time quantum algorithms for integer factorisation and for computing discrete logarithms. This means that in the future, as soon as the first large-scale[2] quantum computer will be built, the currently deployed cryptographic schemes securing our communications, online-banking, and internet access will become insecure. Researchers are still debating on when this quantum advent will happen, but they widely agree we should not be found unprepared.

In order to address the quantum menace, new cryptographic primitives are needed: algorithms that can be implemented on currently available devices, and still guarantee security against both classical and quantum adversaries. On 3 January 2017, NIST published a call[3] for new post-quantum standards in public-key encryption and digital signature algorithms. All submissions were encouraged to provide parameters for five different security levels. The levels (from 1 to 5) are defined as follows: breaking the hard problem underlying the security of the cryptographic scheme should require at least the same amount of resources necessary for

1 $\longrightarrow$ key search on a block cipher that uses 128-bit keys (such as AES-128)

2 $\longrightarrow$ collision search on a 256-bit hash function (such as SHA256)

3 $\longrightarrow$ key search on a block cipher that uses a 192-bit key (such as AES-192)

4 $\longrightarrow$ collision search on a 384-bit hash function (such as SHA384)

5 $\longrightarrow$ key search on a block cipher that uses 256-bit keys (such as AES-256)

[2] Capable of handling enough quantum gates to run Shor's algorithm on concrete instances of the factorisation and the discrete logarithm problems.

[3] https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardisation

At the first round, 69 submissions were accepted, 26 of which made it to the second step, 7 finalists and 8 alternate candidates survived after the third round, and the winners (1 PKE scheme and 3 DS scheme) were announced on 5 July 2022. The proposed protocols move towards new algebraic settings and hard mathematical problems, and can be categorised in five main families.

Isogeny-Based Cryptography is a very young field with roots in Elliptic Curve Cryptography (ECC). People often refer to Couveignes' work [Cou06] from 1997 as the kick-starter of cryptographic research in this area; a rather slow start, due to the impracticality of its first schemes. The research effort was significantly boosted by the NIST submission SIKE [JAC+17], the only isogeny-based cryptosystem in the competition. The scheme based its security on a variant of the classical problem of computing isogenies between supersingular elliptic curves. Although it required very short keys, its fastest implementation was still pretty slow compared to other candidates. Notice how the past-tense is strictly required here, as result of recent and powerful attacks on SIDH [CD22, MM22, Rob22]. Being the main topic of this thesis, we will defer a deeper treatment of this topic to the rest of this document.

Lattice-based cryptography deals with lattices, i.e discrete subgroup of the n-dimensional Euclidean space $\mathbb{R}^n$, with strong periodicity properties. Most of the lattices used in cryptography are (modular) integer lattices, since they are easy to handle and still provide hardness of certain problems. Among all possible lattices, ideal lattices play an important role in cryptography: some cryptosystems exploit their additional algebraic structure in order to get small key sizes and faster implementations. In a generic lattice-based scheme, the secret key is a lattice vector, and the public key corresponds to the multiplication of this vector by a large matrix, perturbed by some small secret error. Amongst the mathematical problems that guarantee the security of these schemes, it is worth mentioning the *Shortest Vector Problem* (given an integer lattice basis as input, find the shortest non-zero vector in the lattice) and the *Learning With Errors* problem (recover a secret $s \in \mathbb{Z}_q^n$ given a system of random linear equations on s that have been perturbed by adding a small unknown noise). Lattice-based cryptography is the most represented area at the late stages of the NIST competition, with three schemes [SAB+17, LDK+17, PFH+17] among the four ones chosen for standardisation.

Hash-based cryptography stems from Leslie Lamport's hash-based One-Time Signature scheme [Lam79], turned into a Multiple-Time Signature scheme by introducing Merkle trees. In the generic construction, one creates several key-pairs with the underlying OTS scheme and uses the digests of the public keys as leaves of a Merkle tree, whose root becomes a global public key. Hash-based constructions base their security on the strength of the cryptographic hash function that is used as building block. Hash-based signature schemes can be either stateful, if they require the maintenance of an internal state, or stateless. Despite being easier to implement, stateless schemes tend to be less efficient, and are currently further from standardisation if compared to the stateful ones. The only hash-based candidate in NIST's competition is the stateless SPHINCS+ [HBD+17], which was selected for standardisation.

Code-based cryptography is one of the most mature candidates for post-quantum schemes. In 1978, McEliece [McE78] introduced a public key cryptosystem based on error correction codes. The author's original idea was to use as ciphertext a word of a binary Goppa code (a linear error-correcting code) to which a random error vector was added. The polynomial-time algorithm to correctly decode the ciphertext is known only to authorised users, while adversaries are left to tackle a generic syndrome decoding problem. Several code-based candidates were submitted to NIST's competition, and all candidates in the fourth round are now code-based (despite not having been selected for standardisation). Together with the supposed robustness against quantum computers, the main advantage of this scheme comes from the simplicity of the involved operations, which leads to a fast encryption and decryption. The major drawback is located in the large size of the keys, which varies from 100 kB to several MB.

Multivariate cryptography bases it security on the Multivariate Quadratic polynomial problem, i.e. to find a solution to a system of multivariate quadratic polynomials. Algebraically, the structure corresponding to this system is the ideal generated by the polynomials defining the system, and therefore algebraic geometry is the most suitable mathematical tool to handle multivariate cryptography. The public key is a set of multivariate polynomials, encryption is the evaluation of these polynomials, decryption requires a trapdoor to invert the quadratic map to find the plaintext. Similarly, to sign here means to use the trapdoor to find a solution to the system with the message to be signed as a target, and verification is simply an evaluation of polynomials. Multivariate cryptosystems offer several advantages in terms of speed, given the simplicity of the involved operations (basically dealing with matrices and vectors) computational requirements and signature length, the main drawback with multivariate cryptosystems consists in the large size of the public keys, necessary to guarantee the security of the schemes.

1.2 History of isogeny-based cryptography in pills

As briefly mentioned in the previous section, the first isogeny-based cryptosystems was proposed by Couveignes in 1997 [Cou06], later made popular by Rostovtsev and Stolbunov [RS06] in 2006. Couveignes first introduced the notion of hard homogeneous space (HHS) as a counterpart of cryptographic protocols based on the discrete logarithm problem. A concrete instantiation was proposed for ordinary elliptic curves, which show a Diffie-Hellman like structure under the class-group action of their endomorphism rings. With a slight change of viewpoint, the scheme uses random walks in graphs of ordinary elliptic curves. Despite being innovative, the resulting scheme was way far from practical, and it was forgotten after its rejection from Crypto 1997. Moreover, Childs, Jao and Soukharev [CJS14] have shown a reduction from the security of the CRS scheme to the abelian hidden-shift problem, solvable using quantum algorithms with a time complexity of $L_q[1/2]$ [Kup03, Reg04].

Still in 2006, Charles, Goren and Lauter [CLG09] proposed a new construction from expander graphs, taking supersingular elliptic curves over $\mathbb{F}_p^2$ as vertices, and isogenies of prime degree $\ell \neq p$ between such curves as edges. More specifically, they designed a hash function whose collision resistance reduces to the difficulty of computing isogenies between supersingular elliptic curves. While very secure, with the best known attacks having exponential complexity, the scheme required about $2 \log p$ modular multiplications per input bit and was thus much less efficient than other provably secure hash algorithms. It is worth noticing that further results [KLPT14, PL17] have shown a potential backdoor threat in standardised parameters.

A major energy and interest boost resulted from the design of the Supersingular Isogeny Diffie-Hellman (SIDH) protocol [JD11], later reshaped as the PKE scheme SIKE [JAC+17] submitted to NIST's competition. In the 2011 paper, the authors provided protocols to achieve zero-knowledge identification, key-exchange and public-key encryption, all based on isogeny graphs from supersingular elliptic curves. Supersingular elliptic curves became of particular relevance for several reasons that will be made clear in Section 2.2, but let us state a few now. First of all, they are defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$, instead of $\overline{\mathbb{F}}_p$ as in the general case. Secondly, the isogeny graph $\mathcal{G}_{p^2}(\ell)$, that has for vertices the isomorphism classes of supersingular elliptic curves and for edges isogenies of degree ℓ, has very good and rapid mixing properties. Thirdly, one can efficiently evaluate isogenies of large (prime-power) degree by choosing a set of curves with smooth order (in practice, choosing p properly). Lastly, SIDH is not defined on an HHS: no group action is used, which seemed to suggest a higher resistance against quantum speed-ups in finding paths on the isogeny graph. We conclude this brief introduction to SIDH (and related schemes) by noting that the last point also an extremely negative side. In order to enable a Diffie-Hellman like structure, which normally requires the presence of a commutative group action, the action of isogenies on certain torsion points need to be exchanged. This extra information will turn out to be a fatal weakness of the scheme (see Section 4.3).

Many new isogeny-based schemes have been designed on the footprint of SIDH in the last years, but not all of them. In 2018, a paper by Castryck, Lange, Martindale, Panny and Renes [CLM+18] showed how to adapt the CRS contruction to supersingular elliptic curves restricted to $\mathbb{F}_p$. They introduce the group action of $\mathbb{F}_p$-rational endomorphisms on supersingular elliptic curves, which allows for the creation of Diffie-Hellman like protocols. While still subject to the reduction by Childs, Jao and Soukharev, which does not dramatically affect its security, it reaches practicality in terms of speed and key-size, and offers public-key validation that SIDH cannot guarantee. For this reason, and with the recovered commutative group action, a new line of research has started that uses CSIDH as a drop-in replacement for the Diffie-Hellman KEX.

1.3 Thesis Structure

Plenty of improvements, adaptations, new constructions and attacks have originated from the aforementioned protocols, some of which are included as novel results in this thesis. In order to understand them, we first need to investigate the algebraic and cryptographic landscape they belong to. This thesis is structured as follows:

- Chapter 1 sets the big picture of classical and post-quantum cryptography. After a gentle introduction to the main post-quantum families of protocols, we point at some milestones in the history of isogeny-based cryptography;
- Chapter 2 provides the reader with the algebraic preliminaries which our contributions are build upon;
- Chapter 3 recalls basic protocols and security notions, and describes the isogeny-based protocols most relevant to this manuscript;
- Chapter 4 summarises research questions and results of the papers collected in this thesis, and indicates how the most recent developments on SIDH and SIKE have affected the work hereby contained.

Chapter 2

Algebraic and geometric preliminaries

On projective plane
An elliptic curve stretches:
the endless embrace.

March 2019

This chapter is meant to be a sort of roadmap to the concept one needs to familiarise with in order to understand the context in which we develop our results. We cannot aim for completeness, but on the way we will refer the reader to several excellent books on these matters. Nonetheless, we provide some preliminaries on elliptic curves (Section 2.1), isogeny graphs (Section 2.2), algebras and quaternion algebras (Section 2.3) and we conclude with an overview of the Deuring correspondence (Section 2.4).

2.1 Elliptic curves

With the literature abounding in proofs of the results hereby collected, if after this section the reader will be left with an unsatisfied thirst for knowledge, I recommend Washington's [Was08] and Silverman's [Sil09] books on the theory and practice of elliptic curves. Note that only the truly crucial definitions will be written in the appropriate environment, both for typographical and environmental-friendly reasons (no pun intended).

The abelian group of elliptic curve points. Let us start with a geometric definition that might suffice for some readers.

Definition 2.1.1. *Let $\mathbb{K}$ be a field. An **elliptic curve** over $\mathbb{K}$ is a pair $(E, 0)$, where E is a smooth projective curve of genus 1 over $\mathbb{K}$, and 0 is a distinguished $\mathbb{K}$-rational point*[1] *on E.*

To those of you interested in a more down-to-earth definition of elliptic curves, with numbers and equations: let me first recall some definitions in preparation to Definition 2.1.2. The **projective plane** $\mathbb{P}^2(\mathbb{K})$ over $\mathbb{K}$ is the set of non-zero triplets $(x, y, z) \in \mathbb{K}^3$ modulo the equivalence relation $(x_1, y_1, z_1) \sim (\lambda x_2, \lambda y_2, \lambda z_2)$ for all $\lambda \in \mathbb{K}^\times$. Let us denote each equivalence class by $(x : y : z) := \{(\lambda x, \lambda y, \lambda z) : \lambda \in \mathbb{K}^\times\}$, called a **projective point**. The **finite points** in $\mathbb{P}^2(\mathbb{K})$ are those with $z \neq 0$ that can be represented as $(x/z : y/z : 1)$, while the **points at infinity** have form $(x : y : 0)$.

[1] The points on E with coordinates in $\mathbb{K}$ are called $\mathbb{K}$**-rational**; for any field $\mathbb{K} \subseteq \mathbb{K}_1$, one indicates with $E(\mathbb{K}_1)$ all points on E defined over $\mathbb{K}_1$.

The 2-dimensional **affine plane** over $\mathbb{K}$, namely $\mathbb{A}^2(\mathbb{K}) = \{(x, y) \in \mathbb{K}^2\}$, allows us to define the embedding

$$\begin{aligned} \iota : \mathbb{A}^2(\mathbb{K}) &\longrightarrow \mathbb{P}^2(\mathbb{K}) \\ (x, y) &\longrightarrow \iota(x, y) = (x : y : 1) \end{aligned}$$

that identifies $\mathbb{A}^2(\mathbb{K})$ with the set of finite points of $\mathbb{P}^2(\mathbb{K})$.

Given a homogeneous polynomial $f(x, y, z) \in \mathbb{K}[x, y, z]$, a **plane projective curve** $C/\mathbb{K}$ (over $\mathbb{K}$) is the locus of the points in $\mathbb{P}^2(\mathbb{K})$ which are zeros of f. For any field extension $\mathbb{F} \supseteq \mathbb{K}$, the **$\mathbb{F}$-rational points** of C form the set $C(\mathbb{F}) = \{P = (x : y : z) \in \mathbb{P}^2(\mathbb{F}) \mid f(P) = 0\}$. Given a point $P \in C$, a curve C is **non-singular at P** if the partial derivatives of F evaluated at P do not simultaneously vanish, i.e. if $(F_x(P), F_y(P), F_z(P)) \neq (0, 0, 0)$. If C has no singular point, then we call it a **non-singular curve**.

In order to properly define the genus of a curve, one would have to introduce many nasty details from algebraic geometry. Since this thesis only deals with genus-1 curves, an intuition should suffice. Topologically, the **genus** of a curve in an integer representing how many times a connected surface can be cut without disconnecting the resulting manifold. For example, a sphere has genus 0, since it cannot be cut along any curve without obtaining a disconnected result, while a torus has genus 1.

We are now ready for a second, more operational analogue of Definition 2.1.1.

Definition 2.1.2. *Let $\mathbb{K}$ be a field of characteristic different from 2 and 3. An **elliptic curve** E over $\mathbb{K}$ (denoted by $E/\mathbb{K}$) is the locus of points in $\mathbb{P}^2(\overline{\mathbb{K}})$ satisfying the **short Weierstrass equation***

$$y^2 = x^3 + Ax + B \tag{2.1}$$

*with $A, B \in \mathbb{K}$, of discriminant $\Delta_E = -16(4A^3 + 27B^2) \neq 0$[2], together with a special $\mathbb{K}$-rational point 0_E, called the **point at infinity**.*

There exists a generalised Weierstrass equation, but if $\mathrm{char}(\mathbb{K}) \neq 2, 3$ as in our case-study, it can be affinely transformed into a short Weierstrass equation. In this case, one can also prove that any elliptic curve as per Definition 2.1.1 is isomorphic to an elliptic curve as per Definition 2.1.2.

The point at infinity 0_E can be defined as follows. Consider the homogeneous form of Equation (2.1), i.e. $y^2z = x^3 + Axz^2 + Bz^3$, such that each point $(x, y) \in E$ corresponds to the point $(x : y : 1)$ in the projective version. In order to define the point at infinity, intersect the curve with $z = 0$, which implies that $x = 0$. Therefore $(0 : y : 0)$, with $y \neq 0$, lies at infinity, and by rescaling, we get that $0_E := (0 : 1 : 0)$ is the only point at infinity on E.

When we move away from homogeneous coordinates, we lose the point at infinity; we must reintroduce it as a formal point to define a binary operation on E and endow the elliptic curve with an additive group structure. In fact,

[2]This condition on Δ is equivalent to requiring the curve E to be non-singular, i.e. that the polynomial $x^3 + Ax + B$ does not have multiple zeros.

[Was08][Theorem 2.1] proves that *the elliptic curve points form an abelian group* with respect to the famous addition operation, visually defined via the secant line. The point 0_E acts like the identity of the group, and results from adding any two points of the form (x_1, y_1) and $(x_1, -y_1)$ (y_1 being possibly 0). Interestingly enough, three points (counted with multiplicity) add to 0_E if and only if they lie on the same straight line.

Scalar multiplication and supersingularity. Let E be an elliptic curve over $\mathbb{K}$; given an integer m, we define the **multiplication-by-m** (at times **scalar multiplication**) as the homomorphism $[m] : E \longrightarrow E$ adding m copies of P together. Its kernel, denoted by $E[m] := \{P \in E \mid [m]P = 0_E\}$, is called the m**-torsion subgroup of** E. We now state an important theorem that unravels the algebraic structure of torsion subgroups.

Theorem 2.1.3 ([Was08][Theorem 3.2)**.** *Let E be an elliptic curve over a field $\mathbb{K}$ and let m be a positive integer. If char$(\mathbb{K}) \neq 0$ or char$(\mathbb{K}) \nmid m$, then $E[n] \simeq \mathbb{Z}_n \oplus \mathbb{Z}_n$. If char$(\mathbb{K}) = p > 0$ and $p|n$, split $n = p \cdot n'$ with $p \nmid n'$. Then $E[n] \simeq \mathbb{Z}_{n'} \oplus \mathbb{Z}_{n'}$ or $\mathbb{Z}_{n'} \oplus \mathbb{Z}_n$.*

In particular, the p-torsion subgroup plays a crucial role in determining an elliptic curve property we are highly interested in through this document, that was already mentioned in Section 1.2.

Definition 2.1.4. *An elliptic curve is called **supersingular** if $E[p] \simeq 0$, or **ordinary** if $E[p] \simeq \mathbb{Z}_p$.*

Oddly enough at first sight, the definition of singular points is unrelated with Definition 2.1.4, and they actually have a very different aura. The former means something usually "bad", meaning that it deals with elliptic curves that somehow misbehave at some points. The latter is conversely very "good", because it turns out that supersingular elliptic curves have the largest possible endomorphism rings (see [Was08][Theorem 10.2] and the theory of complex multiplication) and are, for this and many other reasons, excellent for cryptographic purposes.

The non-singularity of an elliptic curve allows us to define an invariant with respect to isomorphisms (where an **isomorphism** between elliptic curves is defined as a morphism of curves of degree 1, which is invertible).

Definition 2.1.5. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over $\mathbb{K}$ with char$(\mathbb{K}) \neq 2, 3$. We define the* **j-invariant** *of E to be*

$$j_E = 1728 \, \frac{4A^3}{4A^3 + 27B^2}$$

It is important to notice that two particular values will be relevant in our case-study: $j = 0$, occurring when $A = 0$ in curves of the form $y^2 = x^3 + B$, and $j = 1728$, occurring when $B = 0$ in curves of the form $y^2 = x^3 + Ax$. Notice also that the isomorphisms in question are defined over $\overline{\mathbb{K}}$; if the field is not algebraically closed, there might not exist rational functions over $\mathbb{K}$ able to transform a curve into the other.

The Frobenius endomorphism and point counting. Another prominent map on elliptic curves defined over $\mathbb{F}_q$ is the **Frobenius endomorphism** π_q (at times the subscript q is omitted when clear from the context), that maps $(x, y) \mapsto (x^q, y^q)$. Recurring in several aspects of elliptic curve and isogeny-based cruptography, we now see its first appearance in determining the cardinality of an elliptic curve. In first approximation of this number, we must mention Hasse's famous theorem [Was08][Theorem 4.2]: the number of $\mathbb{F}_q$-rational points of an elliptic curve E defined over a finite field $\mathbb{F}_q$ satisfies $|q + 1 - \#E(\mathbb{F}_q)| \leq 2\sqrt{q}$. The range $[q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$ is also known as **Hasse's interval**. It is interesting to see how the cardinality of an elliptic curve is not immediately determined by q and by the curve's equation. To this day, determining the cardinality of an elliptic curve is still somewhat slow, with the fastest algorithm of complexity $O(\log^3 p)$ (or $O(\log^2 p)$ in some special cases [YM21]).

It turns out ([Was08][Theorem 4.10]) that the Frobenius endomorphism satisfies the quadratic equation $X^2 - tX + q = 0$ for some $|t| \leq 2\sqrt{q}$. We call t the **trace** of the Frobenius endomorphism, and $X^2 - tX + q$ the **characteristic polynomial of Frobenius**. A more precise formulation of Hasse's theorem tells us that $\#E(\mathbb{F}_q) = q + 1 - t$, where $|t| \leq 2\sqrt{q}$ is the trace of π_q. This leads to an opening for point counting algorithms: one can try compute t modulo several different primes, and reconstruct the value in the target interval using the Chinese Remainder Theorem. This line of thinking was initiated in 1985 with Schoof's algorithm [Sch85], which was further improved by Atkin and Elkies [Sch95], by Bostan-Morain-Salvy-Schost [BMSS08] and by Lercier-Sirvent [LS08] to say a few. We conclude this subsection linking the trace t of π_q with the number of points on a supersingular elliptic curve.

Theorem 2.1.6 ([Was08][Theorem 4.31)**.** *Let E be an elliptic curve over $\mathbb{F}_{p^n}$ for some prime p and $n \in \mathbb{N}$. Then E is supersingular if and only if $t \equiv 0 \mod p$, which happens if and only if $\#E(\mathbb{F}_{p^n}) \equiv 1 \mod p$.*

2.2 Isogenies and isogeny graphs

Many of the definitions contained in this subsection are available in slightly different formulations that turn out to be equivalent. Here we adopt an algebraic point of view, following [Was08][Chapter 12.2].

Definition 2.2.1. *Given two elliptic curves E_1, E_2 over a $\mathbb{K}$, an **isogeny** $\varphi : E_1 \longrightarrow E_2$ is a surjective morphism that induces a group homomorphism from $E_1(\mathbb{K})$ to $E_2(\mathbb{K})$. The curves E_1 and E_2 are called **isogenous**. A famous result by Tate [Tat66] tells us that two elliptic curves over $\mathbb{K}$ are isogenous over $\mathbb{K}$ if and only if they have the same number of $\mathbb{K}$-rational points.*

An isogeny can be represented via rational functions as $\varphi(x, y) = (r_1(x), y \cdot r_2(x))$; if r_1, r_2 take coefficients in $\mathbb{F}$, then we say that φ is **defined over** $\mathbb{F}$ (not necessarily equal to $\mathbb{K}$). The **degree** of φ is defined as

$$deg(\alpha) = \max\{\deg(p_1(x)), \deg(q_1(x))\},$$

where $r_1(x) = p_1(x)/q_1(x)$. Sometimes we talk about an ℓ-isogeny to indicate that the latter has degree ℓ.

The typical operation defined over isogenies is composition. Two isogenies $\varphi_1 : E \longrightarrow E_1$ and $\varphi_2 : E_1 \longrightarrow E_2$ can be composed as homomorphisms. The resulting isogeny $\varphi_2 \circ \varphi_1 : E \longrightarrow E_2$ has degree $\deg(\varphi_2 \circ \varphi_1) = \deg(\varphi_1) \cdot \deg(\varphi_2)$ For any ℓ-isogeny $\varphi : E_1 \to E_2$ there exists a **dual isogeny** $\hat{\varphi} : E_2 \to E_1$ of degree ℓ such that their composition is the multiplication-by-ℓ (on the corresponding elliptic curves).

If the derivative $r_1'(x)$ is not identically zero, we say that α is **separable**, and **inseparable** otherwise. For example, the Frobenius endomorphism π_q is an inseparable isogeny from $E(\overline{\mathbb{F}_q})$ to itself: in this case, $r_1(x) = x^q$ by definition and $r_1'(x) = qx^{p-1}$ is identically 0 in $\mathbb{F}_q$. On the contrary, the multiplication-by-m map ($m \notin \{0, p\}$) is separable and has degree m^2. More in general, it is possible to prove that any inseparable isogeny φ in characteristic p can be uniquely decomposed as $\varphi = \phi \circ \pi_p^r$ for some $r \in \mathbb{N}$ and some separable isogeny ϕ.

Theorem 2.2.2 ([Was08][Proposition 12.8]). *If an isogeny $\varphi : E_1 \longrightarrow E_2$ is separable, then $\deg(\varphi) = \#\ker(\varphi)$. Otherwise, $\deg(\varphi) > \#\ker(\varphi)$. In particular, the kernel of an isogeny is a finite subgroup of $E_1(\overline{\mathbb{K}})$.*

The kernel of a separable isogeny is actually very important: as proven in [Was08][Proposition 12.12], the image of an isogeny is uniquely determined (up to isomorphisms) by the kernel of the isogeny itself. Mutatis mutandis, the j-invariant of the image curve of an isogeny is uniquely determined by the kernel of the isogeny itself. In practice, given any finite subgroup $G \subset E(\overline{\mathbb{F}}_p)$, there exist a unique (up to isomorphism) elliptic curve $E_2 \simeq E_1/G$ and a separable isogeny $\varphi : E_1 \to E_2$ such that $\ker(\varphi) = G$. Given the kernel of an isogeny, one can use Vélu's formulae [Vél71] to efficiently[3] compute the isogeny φ together with its codomain curve E_2 in $O(\#\ker(\varphi))$ bit operations. When $\ker(\varphi)$ is a cyclic group, we say that φ is a cyclic isogeny.

Isogeny graphs. We now have a brief look into isogeny graphs, whose vertices are elliptic curves and edges are isogenies of a fixed degree. More precisely, we consider every curve and every isogeny up to isomorphism.

Let us fix a prime ℓ and a field $\mathbb{K}$. We have seen in Definition 2.1.5 that j-invariants identify isomorphism classes of elliptic curves, so each vertex of the graph can be represented uniquely via a j-invariant. Two j-invariants are ℓ-isogenous if there is an ℓ-isogeny between any two curves with such j-invariants. One way check this, at least when ℓ is small enough, is to see whether the ℓ-th modular polynomial Φ_ℓ vanishes at a pair of j-invariants. Modular polynomials can also be used to define isogeny graphs as in the following.

Definition 2.2.3. *[Sut13, Definition 3]) The ℓ-**isogeny graph** has vertex set $\mathbb{K}$ and directed edges (j_1, j_2) with multiplicity equal to the multiplicity of j_2 as root of $\Phi_\ell(j_1, Y)$.*

[3] when ℓ is small enough and p is within a few thousand bits

We have mentioned in Definition 2.2.1 how Tate's theorem proves that elliptic curves are isogenous if and only if they have the same cardinality. This implies that, for fields of positive characteristic p, the isogeny graph can be split in two components: a supersingular one (where all elliptic curves have $p+1$ points by Theorem 2.1.6) and an ordinary one.

Basically all recent and practical cryptosystems that involve an isogeny graph are built on its supersingular component. In fact, the ordinary component is quite problematic: the vertices have degree $(\ell+1)$ only for finitely many choices of ℓ, in most cases they are either isolated or 2-regular. The structure corresponding to an ordinary isogeny graph is known as volcano.

Definition 2.2.4. *An ℓ-volcano V of depth d is a connected undirected graph whose vertices are partitioned into levels $V_0, \ldots, V_d$ such that:*

- *the surface V_0 is a regular graph of degree at most 2, possibly with loops;*
- *for $i > 0$, each vertex at level V_i has exactly one edge to a node in level V_{i-1};*
- *for $i < d$, each vertex in V_i has degree $\ell + 1$.*

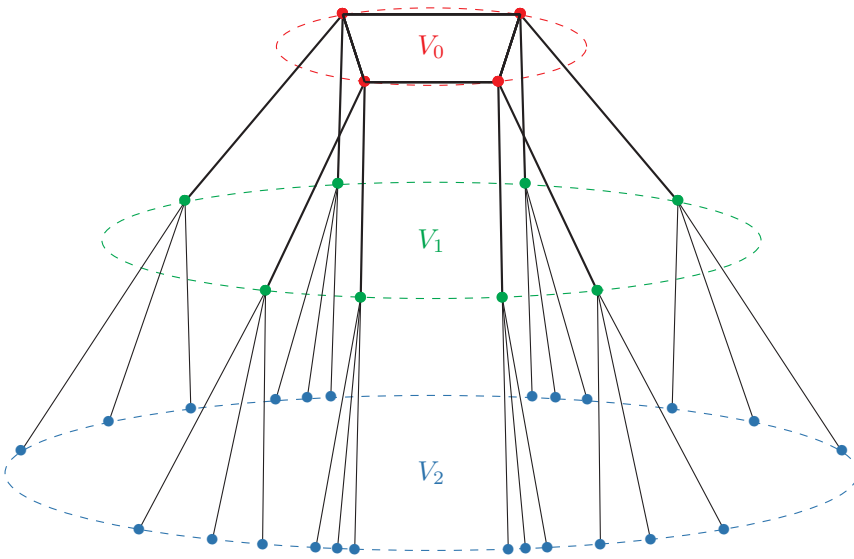

Figure 2.1: A 3-isogeny volcano of depth 2, with 3-isogenies drawn in black (thickness changes only for visibility reasons).

A single volcano does not seem to guarantee good mixing properties, nor a short diameter. Ordinary isogeny graphs are still impractical, with the most recent and efficient implementation (to the best of our knowledge) of the CRS cryptosystems to be found in [DKS18]. The reader further interested in isogeny volcanoes can take [Sut13] as a good starting point.

On the contrary, the supersingular component is excellent for cryptographic purposes. Let us re-define it introducing some notation, and highlight some of its interesting features. Let $p \geq 5$ be a fixed prime, and $\ell \neq p$ another prime.

Definition 2.2.5. *The* ***supersingular ℓ-isogeny graph*** $\mathcal{G}_{p^2}(\ell)$ *has for vertices the $\mathbb{F}_{p^2}$-isomorphism classes of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, and for edges all ℓ-isogenies up to post-composition with $\mathbb{F}_{p^2}$ isomorphisms.*

Let us now list some cryptographically handy properties of $\mathcal{G}_{p^2}(\ell)$.

- *The vertices are defined over $\mathbb{F}_{p^2}$.* In general, j-invariants of elliptic curves over $\mathbb{F}_p$ are defined over the algebraic closure $\overline{F_p}$. From [Sil09, Chapter 5, Thm. 3.1], we know that every supersingular j-invariant in $\overline{\mathbb{F}}_p$ actually lies in $\mathbb{F}_{p^2}$. This also means that every supersingular elliptic curve can be defined over $\mathbb{F}_{p^2}$ with a change of coordinates.
- *The vertex set is almost as large as p.* There are quite many j-invariants for a large prime p: more precisely, there are

$$\left\lfloor \frac{p}{12} \right\rfloor + \begin{cases} 0 & p \equiv 1 \mod 12 \\ 1 & p \equiv 5, 7 \mod 12 \\ 2 & p \equiv 11 \mod 12 \end{cases}$$

  isomorphism classes of supersingular elliptic curves over $\overline{\mathbb{F}}_p$ (and thus over $\mathbb{F}_{p^2}$ for the previous point).
- *The graph is undirected and connected (away from curves with extra automorphisms, such as those with $j = 0$ or $j = 1728$).* The existence of the dual isogeny implies that the graph is undirected: if the ℓ-isogeny φ links E_1 with E_2, the dual isogeny $\hat{\varphi}$ has degree ℓ and links E_2 with E_1. Moreover, while the ordinary isogeny graph is connected only for a few choices of ℓ, an algorithmic proof by Mestre [Mes86] shows that supersingular isogeny graphs are always connected (by isogenies that are defined over $\overline{\mathbb{F}_p}$ in general).
- *The graph is $(\ell+1)$-regular.* This means that every vertex has degree (the number of edges to and from it) $\ell + 1$. We have seen that the ℓ-torsion subgroup $E[\ell]$ is isomorphic to $(\mathbb{Z}/\ell\mathbb{Z}) \times (\mathbb{Z}/\ell\mathbb{Z})$, so there are $\ell+1$ distinct subgroups of order ℓ in $E[\ell]$. Since an isogeny is uniquely determined by its kernel, we also have $\ell + 1$ possible isogenies from any vertex in $\mathcal{G}_{p^2}(\ell)$.
- *The supersingular ℓ-isogeny graph is a Ramanujan graph.* The definition of Ramanujan graphs is a bit more involved and deals with eigenvalues of the adjacency matrix associated with the graph. Let us just say that this property implies optimal expansion factor (i.e. short diameter, i.e. each vertex can be reached from any other vertex after a relatively short walk) and excellent mixing properties (the distribution of the vertices reachable with $O(\log n)$ steps is close to uniform). We refer the reader interested in the definitions and the proof that supersingular isogeny graphs are Ramanujan towards Pizer's work [Piz90].

We conclude this subsection talking about endomorphisms.

Definition 2.2.6. *An **endomorphism** is an isogeny from an elliptic curve E to itself; the set of endomorphisms of E, together with the zero map and equipped with pointwise addition and composition, forms the **endomorphism ring** $End(E)$ of E.*

Being isogenies, endomorphisms are defined over the algebraic closure of the underlying field; when E is defined over the finite field $\mathbb{F}_q$, we denote by $\mathrm{End}_q(E)$ the subring of endomorphisms defined over $\mathbb{F}_q$, called the $\mathbb{F}_q$-rational endomorphism ring. The simplest example of endomorphism is the scalar multiplication; in fact, it is typically the case that $\mathrm{End}(E) = \mathbb{Z}$ in characteristic 0 (otherwise we have Complex Multiplication (CM) curves). The analysis is not as simple for elliptic curves defined over finite fields though, that always have some non-scalar endomorphisms. Actually, we have already seen the Frobenius endomorphism π_q as an example of a typically non-scalar endomorphism, so we know at least that $\mathbb{Z}[\pi_q] \subset \mathrm{End}_q(E)$.

For ordinary curves, it holds that $\mathrm{End}_p(E) = \mathrm{End}(E)$, while for supersingular curves we have that $\mathrm{End}_p(E) \subset \mathrm{End}(E)$. In particular, $\mathrm{End}(E)$ is an order in a quaternion algebra, while $\mathrm{End}_p(E)$ is an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{p})$. A classical result by Deuring [Deu41] links endomorphism rings to the realm of quaternion algebras, revealing that $\mathrm{End}(E)$ is a maximal order in $B_{p,\infty}$, the quaternion algebra ramified at p and at ∞. Providing a roadmap to the background of the aforementioned result — and understanding what it actually means — is the motivation behind the upcoming section.

2.3 Quaternion algebras

In order to have a clear picture of the algebraic structure of endomorphism rings in characteristic p, let us first take a digression on quaternion algebras and orders. In fact, we would really like to simply assume a communal understanding of the fact that $\mathrm{End}(E)$ over $\mathbb{F}_p$ is isomorphic either to an order in an imaginary quadratic field, or to an order in the quaternion algebra ramified at p and ∞. But we should not. One of the goals of this first part of the manuscript is to provide at least some insights on the algebraic concepts not all cryptographers may know (or remember on the spot). Since this manuscript targets an audience educated in cryptography but not too well-versed in quaternion algebras, let us take some time (and space) to provide the reader with an overview of some definitions and theorems that a quaternion algebra apprentice (such as myself a year ago, and now to various extents) would seek for first. The main reference for this section is the amazing book "Quaternion algebras", by John Voight [Voi21].

2.3.1 Valuations, local fields and p-adic numbers

Definition 2.3.1. *For any field $\mathbb{F}$, a **valuation of** $\mathbb{F}$ is a map $\nu : \mathbb{F} \longrightarrow \mathbb{R} \cup \{\infty\}$ such that*

1. *$\nu(x) = \infty$ if and only if $x = 0$;*
2. *it is a group homomorphism, i.e. $\nu(xy) = \nu(x) + \nu(y)$ for all $x, y \in \mathbb{F}$;*
3. *$\nu(x + y) \geq min\{\nu(x), \nu(y)\}$ for all $x, y \in \mathbb{F}$.*

It follows from point 2. that $\nu(1) = 0$; by convention, we set $k + \infty = \infty + k = \infty$. A valuation ν is **discrete** if the value group $\nu(\mathbb{F}^\times)$ is discrete in $\mathbb{R}$, i.e. at every $x \in \nu(\mathbb{F}^\times)$ there always exists a neighborhood with radius $\epsilon > 0$ whose only point in common with $\nu(\mathbb{F}^\times)$ is x. In other words, $\nu(\mathbb{F})$ has no accumulation points. We will only take into account discrete valuations. Since every discrete subgroup of $\mathbb{R}$ is isomorphic to $\mathbb{Z}$, we can think of our valuations as maps $\nu : \mathbb{F} \longrightarrow \mathbb{Z} \cup \{\infty\}$.

Definition 2.3.2. *An **absolute value** on $\mathbb{F}$ is a map $|\cdot| : \mathbb{F} \longrightarrow \mathbb{R}^+ \cup \{0\}$ that*

1. *is positive-definite, i.e. $|x| = 0$ if and only if $x = 0$;*
2. *is multiplicative, i.e. $|xy| = |x| \cdot |y|$ for all $x, y \in \mathbb{F}$;*
3. *satisfies the triangular inequality $|x + y| \leq |x| + |y|$ for all $x, y \in \mathbb{F}$.*

*We say that an absolute value is **non-archimedean** if it satisfies the **ultrametric inequality** if $|x + y| \leq \max\{|x|, |y|\}$ for all $x, y \in \mathbb{F}$, otherwise it is said **archimedean**. A field is archimedean if its absolute value is.*

Note how the defining properties of valuations and absolute values seem to mirror each other. This happens because there is a relation between valuations and absolute values, and thus between $\nu(\mathbb{F})$ as an additive subgroup of $\mathbb{R}$ and $|\mathbb{F}|$ as a multiplicative subgroup of $\mathbb{R}$. For any real $c > 1$, every valuation ν corresponds to an absolute value

$$|x| := \begin{cases} c^{-\nu(x)} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

Conversely: for any $s \in \mathbb{R}^+$, every non-Archimedean absolute value $|\cdot|$ corresponds to a valuation

$$\nu_s := \begin{cases} -s \log |x| & \text{if } x \neq 0 \\ \infty & \text{if } x = 0 \end{cases}$$

For example, the trivial valuation $\nu(x) = 0$ for all $x \in \mathbb{F}$ corresponds to the trivial absolute value $|x| = 1$ for all $x \in \mathbb{F}$. *Any absolute value $|x| := c^{-\nu(x)}$ arisen from a valuation is non-Archimedean*, and viceversa.

We say that two valuations ν_1, ν_2 are **equivalent** if there exists $\lambda \in \mathbb{R}^+$ such that $\nu_2(x) = \lambda\nu_1(x)$ for all $x \in \mathbb{F}$. By the correspondence with absolute values, we can analogously say that two absolute values $|\cdot|_1, |\cdot|_2$ are **equivalent** if there exists an $s \in \mathbb{R}^+$ such that $|x|_1 = |x|_2^s$ for all $x \in \mathbb{F}$. A **place** or **prime** of $\mathbb{F}$ is an equivalence class of absolute values. Since the absolute values in the same equivalence class are either all Archimedean or all non-Archimedean, it makes sense to say that a place is either Archimedean or non-Archimedean.

Given a field $\mathbb{F}$ and a nontrivial discrete valuation ν, the **discrete valuation ring** (DVR for short) of ν is $\mathcal{R} := \{x \in \mathbb{F} : \nu(x) \geq 0\}$. Its invertible elements, collected in $\mathcal{R}^\times$, are those with norm equal to 0: since $\nu(1) = 0$ and $\nu(1) = \nu(xx^{-1}) = \nu(x) + \nu(x^{-1})$, it follows that $\nu(x^{-1}) = -\nu(x)$, and thus $\nu(x) = 0$ (otherwise either x or x^{-1} would be in $\mathcal{R}$ despite having negative norm). A valuation ring is a **local ring**, i.e. a ring with a unique maximal ideal (either left or right if the ring is non-commutative). It is also an **integral domain**, i.e. a commutative ring with no zero-divisors. Putting these notions together, we get that a valuation ring is a **local domain**, i.e. a commutative ring with no zero-divisors and the unique maximal ideal

$$\mathfrak{p} = \mathcal{R} \setminus \mathcal{R}^\times = \{x \in \mathbb{F} : \nu(x) > 0\}$$

Any element $\pi \in \mathfrak{p}$ of minimal norm is called a **uniformiser**. By comparing valuations, one can prove that $\pi\mathcal{R} = (\pi) = \mathfrak{p}$. As for any nontrivial maximal ideal, the quotient $\mathcal{K} = \mathcal{R}/\mathfrak{p}$ is a field, called the **residue field** of $\mathcal{R}$. $\mathcal{R}$ is a **complete DVR** if every Cauchy sequence (a sequence of numbers whose distance monotonically tends to 0) of its elements converges in $\mathcal{R}$. The DVR $\mathcal{R}$ is **compact** if it is complete and its residue field $\mathcal{K} = \mathcal{R}/\mathfrak{p}$ is a finite field.

As a worked example, for any prime p and any integer n, we define the **p-adic order** $ord_p(n)$ of n is the exponent of the largest power of p dividing n. By convention, we set $ord_p(0) = \infty$. One can extend this definition to any $a/b \in \mathbb{Q}$ by defining $ord_p(\frac{a}{b}) = ord_p(a) - ord_p(b)$. It is easy to see that *the p-adic order extended to* $\mathbb{Q}$ *is a valuation*, that we denote by ν_p.

$$ord_p : \mathbb{Z} \longrightarrow \mathbb{N} \cup \{\infty\} \qquad\qquad \nu_p : \mathbb{Q} \longrightarrow \mathbb{N}$$
$$n \mapsto \begin{cases} \infty & \text{if } n = 0 \\ \max\{e \in \mathbb{N} : p^e | n\} & \text{if } n \neq 0 \end{cases} \qquad\qquad x = \frac{a}{b} \mapsto ord_p(a) - ord_p(b)$$

We can now define the **p-adic norm** $|\cdot|_p$ as $p^{-\nu_p(x)}$ for any $x \in \mathbb{Q} \setminus \{0\}$, and 0 otherwise. *The p-adic norm is an absolute value on* $\mathbb{Q}$.

$$|\cdot|_p : \mathbb{Q} \longrightarrow \mathbb{N}$$
$$x \mapsto \begin{cases} 0 & \text{if } x = 0 \\ p^{-\nu_p(x)} & \text{if } x \in \mathbb{Q} \setminus \{0\} \end{cases}$$

We are used to defining the field of real numbers $\mathbb{R}$ as the completion of $\mathbb{Q}$ with respect to the absolute value $|\cdot|$. In a nutshell, this means that each Cauchy sequence over $\mathbb{Q}$ converges to an element in $\mathbb{R}$. However, this is not the only way to complete the field of rational numbers. By completing $\mathbb{Q}$ w.r.t. $|\cdot|_p$, we obtain the **field of p-adic numbers** $\mathbb{Q}_p$. Given a prime p and a rational $x \in \mathbb{Q}$, the **p-adic representation** of x is $x = \frac{x_{-m}}{p^m} + \cdots + \frac{x_{-1}}{p} + x_0 + x_1 p + \cdots + x_n p^n$, and we will write $x = (x_n x_{n-1} \dots x_1 x_0 . x_{-1} x_{-2} x_{-3} \dots)_p$. If we restrict ourselves to the integers, we obtain the **ring of p-adic integers** $\mathbb{Z}_p$, which consists of the elements $x \in \mathbb{Q}$ with $|x|_p \leq 1$ (or equivalently, with $\nu_p(x) > 0$). The ring of p-adic integers is a complete DVR, and its residue field is $\mathbb{Z}/p\mathbb{Z}$. If we had followed an algebraic approach to define the p-adic numbers, $\mathbb{Q}_p$ would have turned out to be the field of fractions of $\mathbb{Z}_p$ (as an integral domain), i.e. the smallest field containing all fractions of the form a/b with $a, b \in \mathbb{Z}_p$.

Later on, we will want to work with finite-degree field extensions of $\mathbb{Q}$, which are called **number fields**, and thus we now see how to extend $\mathbb{Q}_p$. Let $\mathbb{K}$ be a degree n extension of $\mathbb{Q}_p$ and call it a **p-adic field** (not to be confused with the field of p-adic numbers $\mathbb{Q}_p$). The **ring of integers** $O_\mathbb{K}$ of $\mathbb{K}$ is the set of all elements $x \in \mathbb{K}$ with minimal polynomial in $\mathbb{Z}_p[t]$, and it is therefore the integral closure of $\mathbb{Z}_p$ in $\mathbb{F}$. In order to extend the p-adic norm to $\mathbb{K}$, we first need to introduce the **norm** $N_{\mathbb{K}/\mathbb{Q}_p} : \mathbb{K} \longrightarrow \mathbb{Q}_p$, which maps $x \in \mathbb{K}$ to the determinant of the matrix M_x representing the left multiplication-by-x. Then, the **normalized $\mathfrak{p}$-adic absolute value** is defined as $|x|_\mathfrak{p} = |N(x)|_p$. The ring of integers $O_\mathbb{K}$ consists of the elements $x \in \mathbb{K}$ with $|x|_\mathfrak{p} \leq 1$. As it happens for $\mathbb{Q}_p$, one can show that $\mathbb{K} \simeq O_\mathbb{K} \otimes_\mathbb{Z} \mathbb{Q} \simeq O_\mathbb{K} \otimes_\mathbb{Z} \mathbb{Q}_p$.

The prime p generates the ideal $p\mathbb{Z}_p$ which is unique and prime in $\mathbb{Z}_p$, but the ideal $pO_\mathbb{K}$ that generates in $O_\mathbb{K}$ might not be prime. A **uniformiser** of $O_\mathbb{K}$ is an element $\pi \in O_\mathbb{K}$ of maximal $|\pi|_\mathfrak{p} < 1$. There are several uniformisers, but they all can be written as $N(\pi) = up^f$ for different units $u \in \mathbb{Z}_p^\times$ but for a unique $f \in \mathbb{N}$. Therefore, the quantity $q := |\pi|_\mathfrak{p}^{-1} = p^f$ is invariant w.r.t. the choice of the uniformiser. The group of units is characterized by $O_\mathbb{K}^\times = x \in \mathbb{K} : |x|_\mathfrak{p} = 1$, and $\mathbb{K}^\times$ can be decomposed as

$$\mathbb{K}^\times = \pi^\mathbb{Z} O_\mathbb{K}^\times := \bigsqcup_{m \in \mathbb{Z}} \pi^m O_\mathbb{K}^\times$$

2.3.2 Algebras

From now on, we consider finite fields of characteristic $\text{char}(p) > 3$, since it is our case of interest.

Definition 2.3.3. *An **algebra over** $\mathbb{F}$ is a ring A equipped with a homomorphism from $\mathbb{F}$ to A such that $Im(\mathbb{F}) \subset Z(A)$, i.e. the image of $\mathbb{F}$ is in the center of A, i.e. the elements in the image of $\mathbb{F}$ commute with any other element of A. An algebra A is a **division algebra** if it is a division ring, i.e. every non-zero element has a two-sided multiplicative inverse, so that division is always well-defined.*

It is very useful to think of an algebra in terms of a $\mathbb{F}$-vector space that is also a ring. For example, the **dimension** $\dim_{\mathbb{F}} A$ (or simply $\dim A$) of an $\mathbb{F}$-algebra is its dimension as a vector space over $\mathbb{F}$, and it therefore makes sense to talk about basis. Moreover, it follows that all homomorphisms of $\mathbb{F}$-algebras are linear, once seen as homomorphic transformations of vector spaces.

Example 2.3.4. The first classical example of an algebra is the ring $M_n(\mathbb{F})$ of square matrices over a field $\mathbb{F}$. It is an $\mathbb{F}$-algebra of dimension n^2 with respect to matrix addition and multiplication, and for $n > 1$, one can easily prove that it is neither a division algebra nor commutative. A second example are the well known Hamilton's quaternions $\mathbb{H} := \mathbb{R}[i, j, k]/\langle i^2 = j^2 = k^2 = ijk = -1\rangle$, which form a non-commutative division $\mathbb{R}$-algebra. Both examples will come in handy in the following sections.

An **homomorphism** of $\mathbb{F}$-algebras is a ring homomorphism that acts like the identity when restricted to $\mathbb{F}$. The **endomorphism ring of** A, denoted by $\mathrm{End}(A)$, is the set of all homomorphisms from A to itself with respect to composition. If we take the automorphisms from A to A w.r.t. composition, we get the **automorphism group** $\mathrm{Aut}(A)$.

Algebras have a nice structure: any unitary n-dimensional $\mathbb{F}$-algebra can be represented as a subalgebra of $M_n(\mathbb{F})$, and here we show how. For each $\alpha \in A$, consider the linear operator $L_\alpha : A \longrightarrow A$ defined via the left multiplication-by-α, mapping $x \mapsto \alpha x$. Let $\{a_1, \dots a_n\}$ be a basis of A as a $\mathbb{F}$-vector space, and let M_α be the **matrix representation** of L_α with respect to this basis. The matrix representation of A is the matrix representation of the map $L : \alpha \in A \mapsto M_\alpha \in M_n(\mathbb{F})$.

An algebra is **simple** if, as a ring, it has no non-trivial two-sided ideals. Wedderburn shows that any simple algebra is isomorphic to $M_n(D)$, where D is a division algebra over $\mathbb{F}$. Furthermore, n and D are uniquely determined by A up to isomorphisms. But what is the structure of $M_n(D)$? One can furthermore prove that if D is a division algebra, then $M_n(D)$ is a simple $\mathbb{F}$-algebra for any $n \in \mathbb{N}$.

An $\mathbb{F}$-algebra A is **central** if its center $Z(A)$ consists exactly of $\mathbb{F}$. One can prove that $Z(M_n(D)) = Z(D)$ for every division algebra. This fact, together with Wedderburn's structure theorem, allows us to prove that we can study central simple algebras (**CSA** for short) instead of central division algebras.

Since algebras can be seen as sorts of vector spaces, can we also "combine" them? Yes, we can. Let A, B be two finite $\mathbb{F}$-algebras. The **direct sum** $A \oplus B$ is an $\mathbb{F}$-algebra of dimension $\dim(A) + \dim(B)$: it is the direct sum of A and B as vector spaces, and it becomes a ring if we consider the component-wise multiplication. Since we can represent the two algebras A and B as submatrices in $M_{\dim(A)}$ and $M_{\dim(B)}$ respectively, we can intuitively see that

$$A \oplus B \simeq \left\{ \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} : a \in A, b \in B \right\} \subset M_{\dim(A)+\dim(B)}$$

Similarly, one can prove the **tensor product** $A \otimes B$ to be an $\mathbb{F}$-algebra of dimension $\dim(A)\cdot\dim(B)$: it is the tensor product of A and B as vector spaces, and it becomes a ring if we consider the multiplication $(a\otimes b)\times(c\otimes b) := ac\otimes bd$. This multiplication is first defined on the basis vectors, and then extended by linearity to any vector in $A \otimes B$. One can easily see that

$$A \otimes B \simeq \left\{ a \odot b : a \in A, b \in B \right\} \subset M_{\dim(A)\cdot\dim(B)}$$

where $\odot$ represents the Kronecker product which results in the block matrix consisting of all submatrices obtained by multiplying the matrix representing a by each element of the matrix representing b.

It is possible to define a new algebra by **extending the scalars** of an existing one, i.e. to redefine an $\mathbb{F}$-algebra A over an extension field $\mathbb{K}$ of $\mathbb{F}$. From the tensor product of A and $\mathbb{K}$ to the $\mathbb{K}$-algebra $A \otimes_{\mathbb{F}} \mathbb{K}$ of dimension $\dim(A)$, there exists a canonical isomorphism that leaves the basis of A unchanged and maps the scalars to $\mathbb{K}$, obtaining a $\mathbb{K}$-vector space instead of a $\mathbb{F}$-vector space. More explicitly, let $\{a_1, \dots a_m\}$ be a basis of A, let $\{e_1, \dots e_n\}$ be a basis of $\mathbb{K}$ as a vector space over $\mathbb{F}$, and let $\{a_1, \dots a_m\} \otimes \{e_1, \dots e_n\}$ be the vector basis of $A \otimes_{\mathbb{F}} \mathbb{K}$ as a mn-dimensional vector space over $\mathbb{F}$. With a slight abuse of notation, the isomorphism is defined as follows:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_{i,j}(a_i \otimes e_j) \longmapsto \big(\sum_{j=1}^{n} \lambda_{1,j}e_j\big)a_1 + \big(\sum_{j=1}^{n} \lambda_{2,j}e_j\big)a_2 + \cdots + \big(\sum_{j=1}^{n} \lambda_{m,j}e_j\big)a_m$$

where $\sum_{j=1}^{n} \lambda_{i,j}e_j$ is represented as a field-element $\mathbb{K}$ for each $i = 1, 2, \dots m$.

Endomorphism algebras. As another example of algebra that is even more relevant for this thesis, let us now and forever fix $\mathbb{F} = \mathbb{Q}$, and let us talk about endomorphism algebras. As we said in Definition 2.2.6, the set of all endomorphisms of E, equipped with pointwise addition and composition, forms the ring $\mathrm{End}(E)$. The endomorphism ring, being an abelian group, is a $\mathbb{Z}$-module[4]: for all $\alpha, \beta \in \mathrm{End}(E)$ and for all $m, n \in \mathbb{Z}$, we have that

$$(m+n)\alpha = m\alpha + n\alpha, \quad m\alpha + m\beta = m(\alpha+\beta), \quad m(n\alpha) = (mn)\alpha, \quad 1\alpha = \alpha.$$

But $\mathrm{End}(E)$ is more than just a $\mathbb{Z}$-module: being a ring, the composition (its multiplication) is compatible with its structure as a $\mathbb{Z}$-module, and it becomes a $\mathbb{Z}$-algebra. In fact, $\mathbb{Z}$ can be seen as a subring of $\mathrm{End}(E)$ under the injective map sending $m \mapsto [m]$. We can now use the tensor product to lift this $\mathbb{Z}$-algebra to a $\mathbb{Q}$-algebra, obtaining the **endomorphism algebra of** E defined as $\mathrm{End}(E)_{\mathbb{Q}} := \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$, whose elements are of the form $m\alpha$ for $m \in \mathbb{Q}$ and $\alpha \in \mathrm{End}(E)$.

If E is defined over a finite field $\mathbb{F}_q$, then one can prove ([Voi21, Lemma 42.1.5]) that the endomorphism algebra $\mathrm{End}(E)_{\mathbb{Q}}$ is either an imaginary

[4]For those unfamiliar with this term, a module is the generalisation of a vector space to a base ring, so that vectors consist of ring elements instead of field elements.

quadratic field or a quaternion algebra. More to the point, the endomorphism ring $\mathrm{End}_q(E)$ is either an order in an imaginary quadratic field or an order in a quaternion algebra. It is the Frobenius endomorphism that gets to set the rules: if $\pi \in \mathbb{Z}$, the endomorphism ring is non-commutative and we get a quaternionic structure. On the other hand, if $\pi \notin \mathbb{Z}$, then $\mathrm{End}_q(E)$ is an order in the imaginary quadratic field $\mathbb{Q}(\pi) \simeq \mathbb{Q}(\sqrt{t^2 - 4q})$ that contains $\mathbb{Z}[\pi] \simeq \mathbb{Z}[\sqrt{t^2 - 4q}]$, where t is the trace of π.

If we take ordinary elliptic curves over finite fields or supersingular elliptic curves defined over $\mathbb{F}_p$ with $p > 3$, we fall under the first case, that in fact establishes the foundations of CRS and CSIDH (presented in Section 3.3). The second case sets instead the background for SIDH (see Section 3.2), KLPT (see Section 3.4) and related algorithms. We delay our investigation on the latter case until Section 2.3.3 and the remainder of this section.

We immediately continue our analysis of the imaginary quadratic field case, recalling some results and definitions that provide the reader with a roadmap to the group action behind the CSIDH algorithm. Most of the definitions here below will be rephrased (and better explained) for quaternion algebras, because that is where we will build the more "advanced" of our results. Let E be a supersingular elliptic curve over $\mathbb{F}_p$ and let $A = \mathrm{End}(E)_{\mathbb{Q}} \simeq \mathbb{Q}(\sqrt{-d}) = \mathbb{K}$ be the imaginary quadratic field that the endomorphism algebra of E is isomorphic to[5].

- A **lattice** in $\mathbb{K}$ is a free $\mathbb{Z}$-module of rank 2. An **order** in $\mathbb{K}$ is a lattice that is also a subring. A **fractional left $\mathcal{O}$-ideal** is a lattice in $\mathbb{K}$ closed under multiplication to the left by $\mathcal{O}$. We denote fractional ideals in quadratic fields by $\mathfrak{a}, \mathfrak{b}, \dots$.
- Let $d < 0$ be a square-free integer and $\mathbb{K} = \mathbb{Q}(\sqrt{-d})$ be an imaginary quadratic field. The **discriminant of** $\mathbb{K}$ is

$$\Delta_{\mathbb{K}} := \begin{cases} -d & d \equiv 1 \mod 4 \\ -4d & \text{otherwise} \end{cases}$$

  There exists a unique maximal order $\mathcal{O}_{\mathbb{K}}$ in $\mathbb{K}$ given by

$$\mathcal{O}_{\mathbb{K}} = \mathbb{Z}\left[\frac{\Delta_{\mathbb{K}} + \sqrt{\Delta_{\mathbb{K}}}}{2}\right].$$

  Any other order in $\mathbb{K}$ can be written $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_{\mathbb{K}}$, where $f = [\mathcal{O}_{\mathbb{K}} : \mathcal{O}]$ is the **conductor** of $\mathcal{O}$, and it has discriminant $\Delta = f^2 \cdot \Delta_{\mathbb{K}}$.
- The **conjugation map** $\overline{a + b\sqrt{-d}} = a - b\sqrt{-d}$ is an automorphism of $\mathbb{K}$ that allows to define the **norm** $N(\alpha) := \alpha\overline{\alpha} = a^2 + b^2 d$. The norm of a fractional ideal $\mathfrak{a}$ is then defined as $N(\mathfrak{a}) := \mathfrak{a}\overline{\mathfrak{a}}$, or more practically as the $\gcd\{N(\alpha) : \alpha \in \mathfrak{a}\}$.

[5] more in general, A could simply be a $\mathbb{Q}$-algebra of dimension r.

- A fractional ideal is **integral** if it is contained in $\mathcal{O}$. A fractional $\mathcal{O}$-ideal $\mathfrak{a}$ is **invertible** if there exists another fractional $\mathcal{O}$-ideal $\mathfrak{a}^{-1}$ such that $\mathfrak{a}\mathfrak{a}^{-1} = \mathcal{O}$. If $\gcd\{N(\mathfrak{a}), f\} = 1$, then $\mathfrak{a}$ is invertible. We denote by $\mathcal{I}$ the set of invertible fractional $\mathcal{O}$-ideals.
- A fractional ideal is **principal** if it is of the form $\alpha\mathcal{O}$ for $\alpha \in \mathbb{K}^*$; all principal ideals, collected in the set $\mathcal{P}(\mathcal{O})$, are invertible. The quotient $cl(\mathcal{O}) := \mathcal{I}(\mathcal{O})/\mathcal{P}(\mathcal{O})$ is an abelian group under ideal multiplication, and it is called the **ideal-class group of** $\mathcal{O}$. Mutatis mutandis, it can be seen as the class set of the equivalence relation $\mathfrak{a} \sim \mathfrak{b} \iff \mathfrak{a} = \mathfrak{b}c$ for some $c \in \mathbb{K}^\times$. The cardinality of $cl(\mathcal{O})$ is approximatively $\sqrt{\Delta_\mathbb{K}}$.

Now, let us focus our attention on how the ideal class-group of $\mathcal{O}$ acts on supersingular elliptic curves over $\mathbb{F}_p$. Let $\mathcal{E}\ell\ell_p(\mathcal{O})$ be the set of supersingular elliptic curves over $\mathbb{F}_p$ with $\mathrm{End}_p(E)$ isomorphic to an order $\mathcal{O}$ in an imaginary quadratic field and let $E \in \mathcal{E}\ell\ell_p(\mathcal{O})$. The action of an $\mathcal{O}$-ideal $\mathfrak{a}$ on E, denoted by $\mathfrak{a} * E$, is defined by translating $\mathfrak{a}$ into an isogeny: compute the **ideal kernel** $E[\mathfrak{a}] := \cap_{\alpha\in\mathfrak{a}}\{ker(\alpha)\}$ and define the isogeny $\varphi_\mathfrak{a} : E \to E_\mathfrak{a} \simeq E/E[\mathfrak{a}]$.

Any $\mathcal{O}$-ideal $\mathfrak{a}$ can be decomposed as the product of $\mathcal{O}$-ideals $\mathfrak{a} = (\pi_p\mathcal{O})^r\mathfrak{a}_s$, where π_p is the p-th Frobenius endomorphism and $\mathfrak{a}_s \not\subseteq \pi_p\mathcal{O}$. Translated into an isogeny, $\varphi_\mathfrak{a}$ has a separable part with kernel $\cap_{\alpha\in\mathfrak{a}_s} ker(\alpha)$, and a purely inseparable part consisting of r iterations of π_p. The isogeny $\varphi_\mathfrak{a}$ and the codomain $\mathfrak{a}*E$ are both defined over $\mathbb{F}_p$ and are unique up to $\mathbb{F}_p$-isomorphism. It follows that multiplying ideals and composing isogenies are equivalent operations in the two realms.

We conclude this subsection recalling a fundamental result by Schoof [Sch87, Theorem 4.5] as stated by Castryck et al. [CLM+18, Theorem 7] on the ideal-class group action of $cl(\mathcal{O})$ on the set of supersignular elliptic curves over $\mathbb{F}_p$. Let $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ be the set of elliptic curves defined over $\mathbb{F}_p$ whose endomorphism ring is isomorphic to $\mathcal{O}$ such that the Frobenius endomorphism π_p corresponds to π.

Theorem 2.3.5. *Let $\mathcal{O}$ be an order in an imaginary quadratic field and $\pi \in \mathcal{O}$ such that $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ is non-empty. Then the ideal class group $cl(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ via the map*

$$\begin{aligned} cl(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) &\longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) \\ ([\mathfrak{a}], E) &\longrightarrow [\mathfrak{a}] * E. \end{aligned}$$

We defer further details on CSIDH and the practical aspects of the ideal-class group evaluation until Section 3.3.

2.3.3 Quaternion algebras

A **quaternion algebra** B is a central simple algebra of dimension four, which is constructed as an extension of $\mathbb{Q}$ by adjoining three non-square elements $i = \sqrt{a}$, $j = \sqrt{b}$ and k such that $k^2 = (ij)(-ji) = -ab$ for some $a, b \in \mathbb{Q}^\times$. We sometimes denote this quaternion algebra by the **Hilbert symbol** $\left(\frac{a,b}{\mathbb{Q}}\right)$ (that

we will write as $H_{\mathbb{Q}}(a,b)$ for typographical reasons), which indicates that B admits a basis $\{1,i,j,k\}$ such that

$$i^2 = a, \quad j^2 = b, \quad ij = k = -ji. \tag{2.2}$$

The basis elements $\{i,j\}$ are called the **standard generators** of the quaternion algebra, since $\{i,j\}$ equivalently represent the same quaternion algebra with respect to the basis $\{1,i,j,ij\}$, given that they satisfy Equation (2.2) (see [Voi21, Lemma 2.2.5]). It follows from the definition that $H_{\mathbb{Q}}(a,b)$ contains three quadratic subalgebras, as depicted in Figure 2.2.

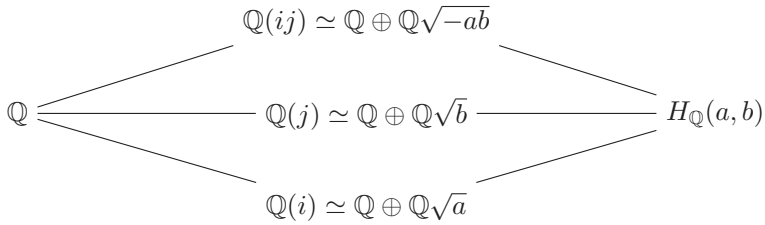

Figure 2.2: Three quadratic subalgebras of $H(a,b)$.

In order to extend scalars, i.e. to redefine the same algebra on an extension field $\mathbb{K} \supset \mathbb{Q}$, one can use the canonical isomorphism between $H_{\mathbb{Q}}(a,b) \otimes_{\mathbb{Q}} \mathbb{K}$ and $H_{\mathbb{K}}(a,b)$ that leaves the basis unchanged, but spans a $\mathbb{K}$-vector space instead of a $\mathbb{Q}$-vector space by taking coefficients over $\mathbb{K}$. More explicitly, let $\{e_1, \dots e_d\}$ be a d-vector basis of $\mathbb{K}$ over $\mathbb{Q}$, and let $\{1,i,j,k\} \otimes \{e_1, \dots e_d\}$ be the $4d$-vector basis of $H_{\mathbb{Q}}(a,b) \otimes_{\mathbb{Q}} \mathbb{K}$ over $\mathbb{Q}$. The isomorphism maps

$$\sum_{v \in \{1,i,j,k\}} \sum_{m=1}^{d} \lambda_{v,m}(v \otimes e_m) \longmapsto \sum_{m=1}^{d} \lambda_{1,m} e_m + (\sum_{m=1}^{d} \lambda_{i,m} e_m)i + \\ + (\sum_{m=1}^{d} \lambda_{j,m} e_m)j + (\sum_{m=1}^{d} \lambda_{k,m} e_m)k$$

There is a particular case that will be crucial when we will define ramification of quaternion algebras. When we extend scalars of a quaternion algebra $B = H_{\mathbb{Q}}(a,b)$ to $\mathbb{R}$, we obtain $B_\infty := B \otimes_{\mathbb{Q}} \mathbb{R}$.

We have defined quaternion algebras as 4-dimensional central simple $\mathbb{F}$-algebras. This characterization actually comes from a corollary of a theorem due to Wedderburn and Artin [Voi21, Theorem 7.1.1]. This corollary says more: a quaternion algebra is either a division algebra or it is isomorphic to $M_2(\mathbb{F})$.

2.3.4 Standard involution, trace and norm

Before restricting our attention to quaternion algebras, let us first define some invariants (w.r.t. the chosen basis) of elements in an algebra A over $\mathbb{F}$. Let

us fix an element $\alpha \in \mathbb{F}$, and let $M_\alpha \in M_n(\mathbb{F})$ be the matrix associated to the linear operator L_α defined as in Section 2.3.2. Then the **trace** of α is the trace of the matrix M_α, i.e. the sum of the elements on its main diagonal. The **norm** of α is the norm of M_α, i.e. the determinant of M_α. The **characteristic polynomial** of α is the characteristic polynomial of M_α, i.e. $p_\alpha(t) := p_{M_\alpha}(t) = \det(tI - M_\alpha) \in \mathbb{F}[t]$, whose roots are exactly the eigenvalues of M_α. If $n = 2$, then $p_\alpha(t) = t^2 - \mathrm{trd}(M_\alpha) + det(M_\alpha)$. In general, notice that the constant term of the characteristic polynomial is always equal to $(-1)^n \det(A)$. The **minimal polynomial** of α is the monic polynomial $\mu_\alpha := \mu_{M_\alpha}$ over $\mathbb{F}$ of minimal degree such that $\mu_{M_\alpha}(M_\alpha) = 0$. By Cayley-Hamilton's theorem, the minimal polynomial always divides the characteristic polynomial.

Given an $\mathbb{F}$-algebra A, an **involution** $^-\colon A \longrightarrow A$ is an $\mathbb{F}$-linear map such that

$$1)\ \overline{1} = 1 \qquad 2)\ \overline{\overline{\alpha}} = \alpha \ \text{ for all } \alpha \in A \qquad 3)\ \overline{\alpha\beta} = \overline{\beta}\overline{\alpha} \ \text{ for all } \alpha, \beta \in A$$

Moreover, if $\alpha\overline{\alpha} \in \mathbb{F}$ for all $\alpha \in A$, then we talk about **standard involution**, in which case

- $\overline{\alpha} + \alpha \in \mathbb{F}$ (since $\mathbb{F} \ni (\alpha + 1)(\overline{\alpha + 1}) = (\alpha + 1)(\overline{\alpha} + 1) = \alpha\overline{\alpha} + \alpha + \overline{\alpha} + 1$ and $\mathbb{F}$ is additively closed);
- $\alpha\overline{\alpha} = \overline{\alpha}\alpha$.

Given a standard involution, we define the **reduced trace** and the **reduced norm** of α respectively as

$$\mathrm{trd}(\alpha) = \alpha + \overline{\alpha} \quad \text{ and } \quad \mathrm{nrd}(\alpha) = \alpha\overline{\alpha}.$$

It easily follows from the definition that the trace is $\mathbb{F}$-linear, i.e. $\mathrm{trd}(m\alpha+n\beta) = m\mathrm{trd}(\alpha) + n\mathrm{trd}(\beta)$ for all $m, n \in \mathbb{F}$, and that the norm is multiplicative, i.e. $\mathrm{nrd}(\alpha\beta) = \mathrm{nrd}(\alpha)\mathrm{nrd}(\beta)$. The norm induces an inner product as follows:

$$(\alpha, \beta) := \frac{1}{2}(\mathrm{nrd}(\alpha + \beta) - \mathrm{nrd}(\alpha) - \mathrm{nrd}(\beta)) = \frac{1}{2}\mathrm{trd}(\alpha\beta) \tag{2.3}$$

Let us now see some interesting properties, which actually hold for every CSA of degree n over $\mathbb{F}$:

1. by noticing that $\alpha^2 - (\alpha + \overline{\alpha})\alpha + \alpha\overline{\alpha}$ is identically zero, we conclude that any $\alpha \in A$ is a root of $x^2 - \mathrm{trd}(\alpha)x + \mathrm{nrd}(\alpha)$, the so-called **reduced characteristic polynomial of** α, that is also its minimal polynomial if $\alpha \notin \mathbb{F}$.
2. the invertible elements of B are all and only the ones of non-zero norm: $\alpha \in A^\times \iff \mathrm{nrd}(\alpha) \neq 0$. In fact, if α is invertible, then

$$1 = \mathrm{nrd}(1) = \mathrm{nrd}(\alpha\alpha^{-1}) = \mathrm{nrd}(\alpha)\mathrm{nrd}(\alpha)^{-1} \Longrightarrow \mathrm{nrd}(\alpha) \neq 0$$

Viceversa, if $\mathrm{nrd}(\alpha) \neq 0$, then the constant term a_0 of the characteristic polynomial $p_\alpha(x) = t^n + a_{n-1}x^{n-1} + \cdots + a_1 t + a_0$ is non-zero. Given that $p_\alpha(\alpha) = 0$ by definition, then $\alpha(\alpha^{n-1} + a_{n-1}x^{n-2} + \cdots + a_1) = -a_0$, and

$$\alpha \text{ has inverse } \alpha^{-1} = \frac{(\alpha^{n-1} + a_{n-1}x^{n-2} + \cdots + a_1)}{-a_0}$$

In particular, if B is a quaternion algebra, the **quaternion involution** (sometimes called conjugation) $\bar{}: B \longrightarrow B$ of an element $\alpha = t + xi + yj + zk$ is defined as $\overline{\alpha} = t - xi - yj - zk$. Following from the definitions of reduced trace and norm, we can see that $\mathrm{trd}(\alpha) = \alpha + \overline{\alpha} = 2t$, and that $\mathrm{nrd}(\alpha) = t^2 - ax^2 - by^2 + abz^2$.

2.3.5 Ramification

We have seen how to construct the field of p-adic numbers $\mathbb{Q}_p$ by completing $\mathbb{Q}$ w.r.t. the p-adic absolute value $|\cdot|_p$, and we know that $\mathbb{R}$ is obtained by completing $\mathbb{Q}$ w.r.t. the usual absolute value $|\cdot| = |\cdot|_\infty$. This slight abuse of notation will help us in consistently defining the ramification of quaternion algebras.

We are interested in whether we can invert multiplication in a quaternion algebra B over $\mathbb{Q}$ when we extend its scalars to $\mathbb{Q}_p$, obtaining the quaternion algebra $B_p := B \otimes_{\mathbb{Q}} \mathbb{Q}_p$. If B_p is a division algebra, then we say that B is **ramified** at p; if not, i.e. if B_p is isomorphic to $M_2(\mathbb{Q}_p)$ as we saw in Section 2.3.3, then we say that B is **split** or **unramified**. In particular, by saying that B is ramified at ∞ we mean that $B_\infty := B \otimes_{\mathbb{Q}} \mathbb{Q}_\infty = B \otimes_{\mathbb{Q}} \mathbb{R}$ is a division algebra. A quaternion algebra is ramified only at a finite and even number of places (so either a prime or ∞ for $\mathbb{Q}$), as shown in [Voi21, Lemma 14.5.3, Theorem 14.6.1]. A quaternion algebra is uniquely determined (up to isomorphisms) by the set of places at which it ramifies. The product of the primes at which B ramifies is called the **discriminant of** B.

2.3.6 Orders, ideals and a group action

Of all the subsections of Section 2.3, all necessary to understand what will be discussed below, this one contains the most relevant objects for this thesis. We now specialise the upcoming definitions to the quaternion algebra $B_{p,\infty}$ ramified at p and ∞. There are two motivations behind this subsection: the first one is that endomorphism rings of supersingular elliptic curves are orders in $B_{p,\infty}$, the second one is that kernels of isogenies can be described as ideals of the endomorphism ring.

Let us start off with the definition of an order, which is totally analogous to the usual one for algebraic number fields.

Definition 2.3.6. *A **lattice** in $B_{p,\infty}$ is a finitely generated $\mathbb{Z}$-submodule of rank 4. An **order** $\mathcal{O}$ in a quaternion algebra $B_{p,\infty}$ is a lattice that is also a subring of $B_{p,\infty}$.*

The simplest example of order one can find is $\mathcal{O} = \mathbb{Z} + \mathbb{Z}i + \mathbb{Z}j + \mathbb{Z}k$, but in general we write $\mathcal{O} = \mathbb{Z}\alpha_1 + \mathbb{Z}\alpha_2 + \mathbb{Z}\alpha_3 + \mathbb{Z}\alpha_4$ for any basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of $\mathcal{O}$. Let $D := \big((\alpha_i, \alpha_j)\big)_{i,j\in\{1,2,3,4\}}$ be the matrix whose elements are computed via the inner product defined in Equation (2.3). The (**reduced**, some textbooks would specify) **discriminant** of $\mathcal{O}$ is the quantity $\mathrm{disc}(\mathcal{O}) := \sqrt{|\det(D)|}$, and is independent of the chosen basis. An order is **maximal** if it is not strictly contained in any other order. If $\mathcal{O}_1 \subset \mathcal{O}_2$, then the index $N = [\mathcal{O}_2 : \mathcal{O}_1]$ is the order (cardinality, in this case) of the quotient $\#(\mathcal{O}_2/\mathcal{O}_1)$. The index satisfies the equality $\mathrm{disc}(\mathcal{O}_2) = N^2\mathrm{disc}(\mathcal{O}_1)$, and two orders are equal if and only if $N = 1$ (i.e. $\mathrm{disc}(\mathcal{O}_1) = \mathrm{disc}(\mathcal{O}_2)$) [Voi21, Lemma 15.5.1].

Given any lattice I (repetita iuvant, a $\mathbb{Z}$-submodule of rank 4 in $B_{p,\infty}$), the **left order** of I is the set $\mathcal{O}_L(I) := \{\alpha \in B_{p,\infty} : \alpha I \subseteq I\}$, and the **right order** of I is the set $\mathcal{O}_R(I) := \{\alpha \in B_{p,\infty} : I\alpha \subseteq I\}$. Both sets actually turn out to be orders in the quaternion algebra, and can be intuitively seen as the largest subrings of $B_{p,\infty}$ that turn I into a module (over $\mathcal{O}_L$ or $\mathcal{O}_R$ respectively).

Definition 2.3.7. *Let $\mathcal{O}$ be an order in $B_{p,\infty}$. A **left fractional $\mathcal{O}$-ideal** is a lattice I in $B_{p,\infty}$ such that $\mathcal{O} \subseteq \mathcal{O}_L(I)$. Analogously, a **right fractional $\mathcal{O}$-ideal** is such that $\mathcal{O} \subseteq \mathcal{O}_R(I)$.*

A simple example of a left fractional $\mathcal{O}$-ideal is a principal ideal $\mathcal{O}\alpha$, where we take any $\alpha \in B_{p,\infty}^\times$ and we multiply to the left by every element in $\mathcal{O}$. These ideals are called fractional because they can all be obtained as $d^{-1}\alpha\mathcal{O}$ for some $\alpha \in \mathcal{O}$ and $d \in \mathbb{N}^+$. In other words, all the ideals we will deal with are fractional, so we will simply drop the adjective from now on.

An ideal I is **integral** if it is contained in its left (or equivalently, right) order, and is said **two-sided** if $\mathcal{O}_L(I) = \mathcal{O}_R(I)$. The reduced norm of I is a positive generator of the submodule generated by the set $\{\mathrm{nrd}(\alpha) : \alpha \in I\}$; equivalently, it is the g.c.d. of the norms of all the elements in I. One may try to multiply two ideals I and J, but this operation is not well-defined in general. This is possible only when $\mathcal{O}_R(I) = \mathcal{O}_L(J)$, in which case we say the ideals are **compatible** and the product IJ is the ideal generated by the products of pairs in $I \times J$. One can easily check that $\mathcal{O}_L(IJ) = \mathcal{O}_L(I)$ and that $\mathcal{O}_R(IJ) = \mathcal{O}_R(J)$.

At this point, one may ask whether inverses exist or not. An ideal I is **invertible** if there exists I^{-1} such that

$$II^{-1} = \mathcal{O}_L(I) = \mathcal{O}_R(I^{-1}) \quad \text{and} \quad I^{-1}I = \mathcal{O}_R(I) = \mathcal{O}_L(I^{-1}). \tag{2.4}$$

Borrowing the definitions in Section 2.3.4, the conjugate[6] ideal $\overline{I} := \{\overline{\alpha} : \alpha \in I\}$ satisfies $\mathcal{O}_L(I) = \mathcal{O}_R(\overline{I})$ and $\mathcal{O}_R(I) = \mathcal{O}_L(\overline{I})$. Moreover, $I\overline{I} = \mathrm{nrd}(I)\mathcal{O}_L(I)$ (and analogously for the reversed order), which put together with Equation (2.4) allows for computing the inverse ideal in terms of the conjugate ideal as follows:

$$I^{-1} = \frac{1}{\mathrm{nrd}(I)}\overline{I}$$

[6]It might seem weird at first to define the conjugate using the involution map and keeping the same symbol. It is actually not, since the involution map defined in Section 2.3.4 is at the same time a conjugate map that flips the signs of all imaginary parts, so to say.

For a given order $\mathcal{O}$, let us denote by $\mathcal{I}(\mathcal{O})$ the set of invertible left $\mathcal{O}$-ideals. We say that two orders $\mathcal{O}_1, \mathcal{O}_2$ are **connected** if there exists an invertible ideal I such that $\mathcal{O}_L(I) = \mathcal{O}_1$ and $\mathcal{O}_R(I) = \mathcal{O}_2$; at times we will say that I is an $\mathcal{O}_1\mathcal{O}_2$**-ideal** meaning that $\mathcal{O}_1$ and $\mathcal{O}_2$ are connected by I. The set of orders connected to $\mathcal{O}$ is the **genus of** $\mathcal{O}$. Two orders are **of the same type** if they are conjugated[7] as subgroups: there exists an invertible quaternion $\alpha \in B_{p,\infty}^{\times}$ such that $\mathcal{O}' = \alpha^{-1}\mathcal{O}\alpha$. Note that, if $\mathcal{O}, \mathcal{O}'$ are of the same type and α realises this conjugation, then $\mathcal{O}\alpha$ is a principal ideal connecting $\mathcal{O}$ with $\mathcal{O}'$.

Two left $\mathcal{O}$-ideals might have right ideals of the same type: two left $\mathcal{O}$-ideals I, J are **equivalent** if there exists an invertible quaternion $\beta \in B_{p,\infty}^{\times}$ such that $J = I\beta$.

2.4 The Deuring correspondence

Now that we have some intuition on quaternion algebras, ideals and orders, let us put them into practice. In this section, we resume our analysis of the link between endomorphism rings of supersingular elliptic curves and isogeny graphs, better equipped with our quaternionic notions.

A classical result by Deuring [Deu41] bridges endomorphism rings and quaternion orders: the endomorphism ring $\mathrm{End}(E)$ for a supersingular curve E is isomorphic to a maximal order in $B_{p,\infty}$, the quaternion algebra ramified at p and at ∞. The connection is unique, up to elliptic curve isomorphisms and Galois conjugacy of supersingular j-invariants. But this is not all, folks. In fact, we can draw lines between quantities and properties of supersingular elliptic curves and their counterparts in the quaternion algebra setting. For example, every ideal corresponds to an isogeny: for a given integral left $\mathcal{O}_0$-ideal I, we can compute the corresponding separable isogeny

$$\phi_I : E_0 \longrightarrow E \simeq E_0/E_0[I]$$

by computing its kernel $E_0[I]$ as

$$E_0[I] = \{P \in E_0 \mid \alpha(P) = 0 \text{ for all } \alpha \in I\}.$$

Viceversa, every isogeny corresponds to an ideal: given the kernel G_φ of a supersingular elliptic curve φ, the **kernel ideal** corresponding to φ is defined as

$$I(G_\varphi) := \alpha \in \mathrm{End}(E) : \alpha(G) = 0.$$

We summarise the most relevant analogies in Table 2.1.

It is of course nice and important to have abstract connections, but we would really like to get something useful out of them, building new cryptanalytic tools and cryptosystems. So far, this correspondence has been exploited pursuing two goals. The first is cryptanalysis: with KLPT [KLPT14], later refined in [EHL+18, Wes22], the Deuring correspondence is used to assess the

[7] Conjugation appears once again with yet another slightly different meaning. This is just a hint of the hardest problem of all: the common notation problem!

Supersingular elliptic curves		**Quaternion algebras**	
Supersingular j-invariant over $\mathbb{F}_{p^2}$ up to Galois conj.	$j(E_0)$	$\mathcal{O}_0$	(Maximal order in $B_{p,\infty}$ up to isom.)
Isogeny from E_0 to E_1	φ	I_φ	$\mathcal{O}_0\mathcal{O}_1$-connecting ideal
Isogeny kernel	$\ker(\varphi)$	$I(\ker(\varphi))$	Kernel ideal
Isogeny degree	$\deg(\varphi)$	$\mathrm{nrd}(I_\varphi)$	Ideal orm
Dual isogeny	$\hat{\varphi}$	$\overline{I_\varphi}$	Conjugate ideal
Endomorphism	θ	$\mathcal{O}_0 \cdot \theta$	Principal ideal
Isogenies from E_0 to E_1	φ, ϕ	$I_\varphi \sim I_\phi$	Equivalent ideals
Supersingular j-invariants	$\{j\}$	$cl(\mathcal{O})$	Class of $\mathcal{O}_0\mathcal{O}$-ideals
Composition	$\varphi \circ \phi$	$I_\phi \cdot I_\varphi$	Ideal multiplication

Table 2.1: The Deuring correspondence in more details.

security of cryptosystems based on supersingular isogeny problems, drawing an equivalence between the supersingular isogeny path and endomorphism ring problems. The second is the design of new protocols for digital signatures [GPS20, DKL+20a, DLW22, GPV21], encryption [DdF+21] and key-exchange [Ler21].

In this thesis, we are particularly interested in the so called **constructive Deuring correspondence**: given a maximal order $\mathcal{O}$ in $B_{p,\infty}$, find a curve E over $\mathbb{F}_p$ with $\mathrm{End}(E) \simeq \mathcal{O}$. Oddly enough, despite this connection being called "correspondence", the other directions is at times referred to as the **endomorphism ring computation problem**. In fact, the state of the art marks a clear distinction: certain problems are hard for elliptic curves and isogenies, but are easy when translated to their quaternionic equivalent. Speeding up the translation from ideals to isogenies under the Deuring correspondence is therefore a quite relevant task for the applicability of the above (and future) results.

Chapter 3

Isogeny-based cryptographic protocols

Isogenies bloom
In supersingular graph:
Cryptographic keys.

December 2022

We start this chapter by recalling in Section 3.1 some basic cryptographic definitions and security properties. Then we overview four isogeny-based protocols: SIDH in Section 3.2, CSIDH in Section 3.3, KLPT in Section 3.4 and SQISign in Section 3.5. These are probably the four most prominent schemes in isogeny-based cryptography, and surely those that have been most relevant to the results collected in this thesis.

3.1 Cryptographic foundations

Even though we could assume the reader is familiar with these basic concepts of public-key cryptography, we hereby define key-exchange, encryption and digital signature protocols, together with the relevant security notions and some security models. In some cases, extending a protocol between two parties to one between multiple parties may not be trivial, but extending the definitions is; let us thus focus on a two-party scenario from now on.

Remark 3.1.1. Notation is a very sensitive topic in research, particularly in cryptography. The attentive reader will soon realise that we have used the same symbol sk for secret keys, decryption keys, decapsulation keys and signing keys, and the symbol pk for public keys, encryption keys, encapsulation keys and verification keys. This choice was made to underline the secrecy of the key material throughout the following protocols, and to facilitate some connections we will draw between the different mechanisms.

3.1.1 Key-exchange

Let us now define a KEX protocol. With respect to other definitions to come, this is not a technical one. This is because we want to capture the most general meaning of key-exchange, without making otherwise necessary distinctions that would inevitably specialise it.

Definition 3.1.2. *A **key-exchange**[1] **protocol** (KEX for short) is a mechanism by which two parties connected via an adversarially-controlled network can agree on a shared key known only to the two parties, for subsequent cryptographic (typically symmetric) use.*

In the simplest case, each party holds a pair $(\mathsf{sk}, \mathsf{pk})$ of **secret key** and a **public key**, linked by some mathematical operation that is easy in one direction (from sk to pk) and hard in the other. This will be the case in all future definitions that will involve key-pairs.

Let $\mathcal{P}_A, \mathcal{P}_B$ be the two parties involved in the protocol, and let $\mathcal{A}$ be the adversary. Assume that each party holds some **long-term key** material and produces some **ephemeral key** material during the protocol run to establish a shared key K. There is a set of the extra security goals that a KEX may have on top of key-establishment, among which we have:

- **key confirmation**: once the KEX has been completed, each party has assurance that the other one is in possession of K;
- **authentication**: $\mathcal{P}_A$ is persuaded that the other party it has shared key with is the intended $\mathcal{P}_B$, and viceversa. Authentication can be **explicit** if achieved using another mechanism on top a KEX (e.g. a digital signature), or **implicit** if it is argued directly from a successful KEX run;
- **forward-secrecy**: even if $\mathcal{A}$ compromises the long-term key material used in a KEX protocol, the shared keys based on that material are still known to $\mathcal{P}_A$ and $\mathcal{P}_B$ only;
- **key compromise impersonation (KCI) resistance**: if $\mathcal{A}$ compromises the long-term key material of a party, it cannot impersonate that party in a subsequent KEX run.

Once we have set some security goals, we have to define what the adversary can do. Trivially, the security strength of the protocol depends on how powerful adversary it can stand. Is the adversary simply allowed to eavesdrop the conversation, or can it change the order of the messages in the protocol, tamper with them, or do even more? One represents the adversarial capabilities as a list of queries that, together with the security goals, form a **security model**. There are several security models available in the literature, all capturing somewhat different adversaries and security notions. The Bellare-Rogaway model [BR94] is considered to be the first one in practical provable security, and has been extended in several ways throughout the years. We provide as example the the security model we used in our contribution presented in Paper 1, in which we use a modified version of the Real-or-Random (ROR) model.

Suppose that we have a **certificate authority** CA that holds and issues certificates for the long-term public keys of the parties $\mathcal{P}_1, \dots \mathcal{P}_n$ in the set $\mathcal{P}$. At any time and without limitations, a new party can join by communicating a

[1] sometimes called *key-establishment*

long-term public key to CA. Different parties are not required to hold different long-term public keys, but each party can register to only key at a time.

Each party $\mathcal{P}_i$ is represented as a set of **oracles** $\{\pi_i^1, \pi_i^2, \ldots, \pi_i^k\}$, one for each of the k key-exchange sessions the user can participate to. Each oracle $\pi_i^s = (P_i^s, \psi_i^s, K_i^s, \mathsf{sent}_i^s, \mathsf{recv}_i^s, role_i^s)$ maintains an internal state consisting of:

- the identity of the party P_i^s that $\mathcal{P}_i$ intends to exchange key with;
- the session-key state $\psi_i^s \in \{\emptyset, accept, reject\}$, indicating whether a session key was computed and has been accepted or rejected;
- the session key K_i^s, non-empty if and only if $\psi_i^s = accept$;
- the transcript sent_i^s collecting all the messages sent by the oracle;
- the transcript recv_i^s collecting all the messages received by the oracle;
- the role $role_i^s$ of the oracle (whether initiator "`init`" or responder "`resp`").

We call the **view** view_i^s of $\mathcal{P}_i$ over the session s as the ordered set of messages sent and received by $\mathcal{P}_i$. Two oracles π_i^s and π_j^t are called **partner oracles** if they have participated in the same KEX session; more formally, if

1. $P_i^s = P_j$ and $P_j^t = P_i$, i.e. if they are the intended peer of each other;
2. $\psi_i^s = \psi_j^t = accept$, i.e. they both accepted the session key;
3. $\mathsf{view}_i^s = \mathsf{view}_j^t$, i.e. the messages sent and received by P_i match with the ones respectively received and sent by P_j during the key-exchange session;
4. they have specular roles.

We will later limit the adversary to test only fresh oracles: an oracle is **fresh** if and only if its session key has not been revealed, its partner oracle has not been corrupted or tested and the partner's session key has not been revealed. A party is said to be **honest** if it has not been corrupted yet, i.e. if all its oracles are fresh.

In this model, an adversary $\mathcal{A}$ has full control over the network and interacts with oracles through the following queries:

- *activate* an oracle π_i^s either as initiator or responder by sending a message on behalf of a peer P_j;
- *reveal the long-term secret key* of a party $\mathcal{P}_i$: the party is then **corrupted** and all its oracles will answer $\perp$ to each later query;
- *register the long-term public key* for a new user. No knowledge of the corresponding secret key is required and the public key is distributed to all other users;
- *reveal the session key* k_i^s stored in the internal state of any oracle π_i^s, which is is now said to be **revealed**.
- *test an oracle* π_i^s, which outputs the key if $\psi_i^s = accept$, and $\perp$ otherwise. If a key is output, then it is either the real session-key or a random key, according to a random bit set at the beginning of the security game (as per definition of the ROR-model).

Once the environment has been set up, we run the following *AKE security game* $G_\Pi(\mu, k)$ between the adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, with simulates the protocol between μ parties that can engage in at most k sessions each:

1. **Setup.** The challenger $\mathcal{C}$ tosses a coin $b \xleftarrow{\$} \{0,1\}$ and sets up μ parties, providing each of them with a long-term key pair (sk_i, pk_i) and with k oracles.
2. **Adversarial queries.** The adversary $\mathcal{A}$, knowing the public keys of all parties, can make any number of the previously defined queries. $\mathcal{C}$ only allows fresh oracles to be tested.
3. **ROR guess.** After some queries, the adversary $\mathcal{A}$ will eventually output b', its guess on b tossed in the Setup phase. If the tested oracles are fresh and $b' = b$, then $\mathcal{A}$ wins the security game.

An adversary can actually break the KEX protocol in three different ways: it can break the soundness of the protocol (by tricking two oracles into computing different session keys), the relation between two partner oracles or the secrecy of the shared key (by winning the AKE security game). We say that the KEX protocol is **AKE-secure** if a polynomial-time adversary has a negligible probability of breaking the protocol in any of the aforementioned ways.

3.1.2 Public-key encryption and key encapsulation

Let us start with the definition of PKE and KEM schemes, this time more rigorously: being more precise will allow us to easily compare the two schemes together and with digital signatures.

Definition 3.1.3. *Let λ be the security parameter. Let K_e be the set of* ***encryption keys****, let K_d be the set of* ***decryption keys****, let X be the set of* ***plaintexts*** *(ore generically "messages") and let Y be the set of* ***ciphertexts****. A* ***public-key encryption scheme*** *(PKE for short) consists of three algorithms:*

- *a* ***key-generation*** *algorithm* KeyGen *which, depending on the security parameter λ, outputs a pair $(\mathsf{sk}, \mathsf{pk}) \in (K_d, K_e)$ where* sk *is sampled uniformly at random and* pk *corresponds to* sk*;*
- *an* ***encryption*** *algorithm* Enc*, which produces a ciphertext $y \in Y$ on input a message $x \in X$ and an encryption key $\mathsf{pk} \in K_e$;*
- *a* ***decryption*** *algorithm* Dec*, which recovers the plaintext x on input the ciphertext y and the decryption key* sk *corresponding to the encryption key used for encrypting x.*

In addition, it must be computationally hard to retrieve the decryption key knowing solely the encryption key.

Together with the fundamental security goal of **confidentiality** (information should be hidden from the adversary), there is a set of the extra security notions and properties one might go after when designing a PKE scheme:

- **semantic security**: any information on the plaintext that can be computed from the ciphertext can also be computed without the ciphertext;
- **non-malleability**: it is infeasible to take the ciphertext of a message and transform it into the ciphertext of a distinct related message without knowing the initial plaintext.

In security games for PKE schemes, we use a notion that is equivalent to semantic security: a PKE scheme has (**cipthertext**) **indistinguishability** (IND) if an adversary cannot distinguish between the encryption of any two messages sampled at random from an adversarially chosen distribution. Let us briefly indulge ourselves with the IND-CPA security game, first defining indistinguishability under chosen-plaintext attack, and later modifying it to achieve two stronger security notions:

1. The challenger $\mathcal{C}$ generates a key pair $(\mathsf{sk}, \mathsf{pk})$ based on the security parameter λ, store sk and communicates pk to the adversary.
2. The adversary $\mathcal{A}$ may submit and receive answer to a polynomial number of encryptions queries.
3. $\mathcal{A}$ submits two distinct chosen plaintexts x_0, x_1 to the challenger.
4. The challenger tosses a coin $b \xleftarrow{\$} \{0,1\}$ and sends the challenge ciphertext $y = \mathsf{Enc}(x_b, \mathsf{pk})$ to $\mathcal{A}$.
5. After number of additional computations or encryptions but on x_0, x_1, $\mathcal{A}$ outputs a guess for the value of b.

A PKE scheme is **Indistinguishable under Chosen-Plaintext Attack** (IND-CPA) if the adversary has a negligible probability of correctly guessing the bit b.

We can strengthen this security notion by allowing decryption queries: a PKE scheme is **Indistinguishable under Chosen-Ciphertext Attack** (IND-CCA1) if the adversary is given the additional ability of making decryption queries in Step 2 of the above game. We can take another step further: a PKE scheme is **Indistinguishable under adaptive Chosen-Ciphertext Attack** (IND-CCA2) if the adversary can make decryption queries both in Step 2 and in Step 5 of the above game ($\mathcal{C}$ does not answer decryption queries on the challenge ciphertext y, because this would trivially reveal b).

We now mix the two concepts of key exchange and public-key encryption defining key encapsulation. In fact, KEMs allow to establish a key for subsequent cryptographic use (as in KEX) through encryption and decryption mechanisms (as in PKE).

Definition 3.1.4. *Let λ be the security parameter. Let K_{encap} be the set of* ***encapsulation keys****, let K_{decap} be the set of* ***decapsulation keys****, let R be the* ***randomness*** *set and let Y be the set of* ***ciphertexts****. A* ***public-key encryption scheme*** *(PKE for short) consists of three algorithms:*

- *a* ***key-generation*** *algorithm* KeyGen *which, depending on the security parameter λ, outputs a pair $(\mathsf{sk}, \mathsf{pk}) \in (K_{decap}, K_{encap})$ where* sk *is sampled uniformly at random and* pk *corresponds to* sk*;*
- *an* ***encapsulation*** *algorithm* Encap*, which produces a ciphertext $y \in Y$ and a symmetric key k on input an encapsulation key* pk *and a random element in R (so to produce different symmetric keys when the input* pk *does not change);*
- *a* ***decapsulation*** *algorithm* Decap*, which computes the symmetric key k on input the ciphertext y and the decapsulation key* sk *corresponding to the encapsulation key used to encapsulate.*

In addition, it must be computationally hard to retrieve the decapsulation key from the encapsulation key only.

The same security notions we described for PKE schemes hold for KEMs too.

3.1.3 Digital signatures

Definition 3.1.5. *Let λ be the security parameter. Let K_s be the set of* ***signing keys****, let K_v be the set of* ***verification keys****, let M be the set of messages and let S be the set of* ***signatures****. A* ***digital signature scheme*** *consists of three algorithms:*

- *a* ***key-generation*** *algorithm* KeyGen *which, depending on the security parameter λ, outputs a pair $(\mathsf{sk}, \mathsf{pk}) \in (K_s, K_v)$ where* sk *is sampled uniformly at random and* pk *corresponds to* sk*;*
- *a* ***signing*** *algorithm* Sign*, which produces a signature $\sigma \in S$ on input a message $m \in M$ and a signing key $\mathsf{sk} \in K_s$. In case the algorithm is randomised, signing the same message with the same key may produce different signatures;*
- *a* ***verification*** *algorithm* Vrfy*, which given a verification key $\mathsf{pk} \in K_v$, a message $m \in M$ and a signature $\sigma \in S$, outputs 1 if σ was created using the message m and the key* sk *corresponding to* pk*, and 0 otherwise.*

In addition, it must be computationally hard to retrieve the signing key knowing solely the verification key.

The most important security goal for digital signature schemes is **unforgeability**: it must be computationally hard for an adversary to produce a valid signature on any message that had not been previously signed. Let us properly define the **existential unforgeability under chosen message attack** (or EUF-CMA in short) security with yet another game involving an adversary $\mathcal{A}$.

1. **Setup.** We run the KeyGen algorithm to produce a pair sk, pk, and provide the adversary $\mathcal{A}$ with the verification key pk.
2. **Signing queries.** The adversary $\mathcal{A}$ can adaptively submit distinct messages m to be signed. In other words, it can decide subsequent queries after having seen the answer to the previous ones. For each message, we compute a signature $\sigma \leftarrow \mathsf{Sign}(m, \mathsf{sk})$ is computed and send it to $\mathcal{A}$. We record all queries and answers produced during this phase in the set $Q := \{(m_i, \sigma_i)\}_i$.
3. **Output.** Finally, $\mathcal{A}$ outputs a message with a forged signature (m^*, σ^*). We say that the adversary $\mathcal{A}$ wins the $\mathsf{EUF\text{-}CMA}$ game if $m^* \notin Q$ and $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) = 1$.

A signature scheme is $\mathsf{EUF\text{-}CMA}$**−secure** if for all polynomial time adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in winning the above game is negligible in the security parameter λ:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EUF\text{-}CMA}}(\lambda) = \mathrm{negl}(\lambda).$$

3.2 SIDH

In this section we provide an overview of the — now insecure — Supersingular Isogeny Diffie-Hellman (SIDH) key-exchange protocol. The idea is to let Alice and Bob choose a secret walk in two distinct isogeny graphs over the same vertex set. Through an intermediate information exchange, they will later be able to agree on a shared secret. We refer the curious reader to the excellent paper by Costello [Cos19] for more details and toy examples.

After introducing the parameters, we will indicate with A and B the degrees of the two parties' secret isogenies. This allows us to lighten the notation and more easily compare the protocols contained in this chapter and in Chapter 4).

Parameters. We denote by $\mathsf{pp} := (\ell_A, \ell_B, e_A, e_B, f, p, E_0, P_A, Q_A, P_B, Q_B)$ the tuple of public parameters in SIDH, where

- ℓ_A and ℓ_B are two small distinct primes (typically 2 and 3),
- e_A and e_B are positive integers such that $A := \ell_A^{e_A} \approx B := \ell_B^{e_B}$,
- f is a small cofactor coprime to A and B,
- p is a prime of the form $p = A \cdot B \cdot f \pm 1$,
- E_0 is a supersingular elliptic curve over $\mathbb{F}_{p^2}$,
- $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ are basis of the torsion subgroups $E_0[A]$ and $E_0[B]$ respectively.

The condition $A \approx B$ implies similar security guarantees for Alice and Bob. The corresponding torsion subgroups are $E_0[A] \cong (\mathbb{Z}/A\mathbb{Z}) \times (\mathbb{Z}/A\mathbb{Z})$ and that $E_0[B] \cong (\mathbb{Z}/B\mathbb{Z}) \times (\mathbb{Z}/B\mathbb{Z})$; for this specific choices of A and B, we have simple and large enough subgroups to work with and sample keys from.

Key generation. Alice randomly picks two integers $m_A, n_A \in \mathbb{Z}/A\mathbb{Z}$ not both divisible by ℓ_A and secretly computes the point $R_A = [m_A]P_A + [n_A]Q_A$ of order A. Then, by using Vèlu's formulae, she computes the isogeny $\varphi_A : E_0 \longrightarrow E_A := E_0/\langle R_A \rangle$. Alice's secret key is the pair $\mathsf{sk}_A = (m_A, n_A)$, while her public key $\mathsf{pk}_A = (E_A, P'_B, Q'_B)$ consists of $E_A := E_0/\langle R_A \rangle$ and the torsion point images $(P'_B, Q'_B) = (\varphi_A(P_B), \varphi_A(Q_B))$. Analogously, Bob randomly picks a secret key $\mathsf{sk}_B := (m_B, n_B) \in (\mathbb{Z}/B\mathbb{Z})^2$ and computes the public key $\mathsf{pk}_B = (E_B, P'_A, Q'_A)$ from the isogeny $\varphi_B : E_0 \longrightarrow E_B := E_0/\langle R_B \rangle$.

We note that isogeny computation and evaluation are usually performed at the same time, following an optimised strategy described in [JAC+17], which is recalled and further optimised for parallel isogenies computation in Paper 3.

In many papers, the SIDH secret key is reduced to a single integer, by sampling a single coefficient n and computing the secret kernel generator as $P + [n]Q$ (by assuming that one of the two coefficients is invertible). This reduces the number of possible secret isogenies from $\ell^{e-1}(\ell + 1)$ to ℓ^e. Despite this leaving the security of the protocol basically unaffected, we prefer to stick with the "complete" version in our description.

Shared key computation. Once Bob's public key has been retrieved, Alice computes $R'_A := \varphi_B(A)$, which is a kernel generator for the isogeny $\varphi'_A : E_B \longrightarrow E_0/\langle A, B \rangle$. By taking the points $P'_A, Q'_A \in E_B$ in Bob's public key, she uses her secret key to compute $R'_A := [m_A]P'_A + [n_A]Q'_A = [m_A]\varphi_B(P_A) + [n_A]\varphi_B(Q_A) = \varphi_B([m_A]P_A + [n_A]Q_A) = \varphi_B(R_A)$. This is a kernel generator of the isogeny φ'_A, through which she computes the j-invariant $j(E_{BA})$ of the image curve $E_{BA} = E_B/\langle R'_A \rangle$. Analogously, Bob computes the isogeny φ'_B via the kernel generator $R'_B = [m_B]P'_B + [n_B]Q'_B$, and then the j-invariant of the image curve $E_{AB} := E_A/\langle R'_B \rangle$. Note that Alice and Bob do not necessarily reach the same elliptic curve, but two curves that are guaranteed to be isomorphic to $E_0/\langle R_A, R_B \rangle$, and thus have the same j-invariant.

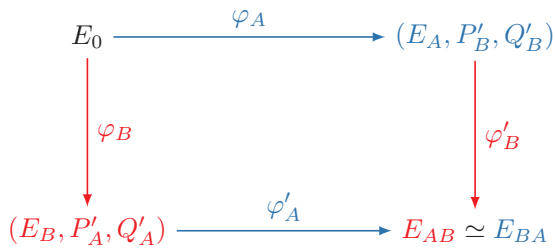

Figure 3.1: The SIDH key exchange. Elements in blue are computed by Alice, those in red are computed by Bob.

Security. The security of SIDH is based on the computational and decisional SIDH problems, which until not so long ago were believed to be hard.

Unlike the usual Diffie-Hellman key-exchange protocol, SIDH is not role-symmetric, due to the different degrees of the isogenies Alice and Bob use in the graph. One can state the CSSI problem regardless of the role, but would then

have to formally introduce role-specific formulations of the decisional problem; for simplicity, we provide Alice's variants in both cases.

Definition 3.2.1 (Computational Supersingular Isogeny Diffie-Hellman problem)**.** *Let* pp *be the tuple of public parameters in SIDH. Let* $\varphi_A : E_0 \longrightarrow E_A$ *be an isogeny whose kernel is generated by* $R_A := [m_A]P_A + [n_A]Q_A$*, following the key-generation algorithm of SIDH. Given* E_A *and the values* $\varphi_A(P_B), \varphi_A(Q_B)$*, the Computational Supersingular Isogeny Problem CSSI requires to determine a generator of the subgroup* $\langle [m_A]P_A + [n_A]Q_A \rangle$*.*

Definition 3.2.2 (A-Decisional Supersingular Isogeny Diffie-Hellman problem)**.** *Let* pp *be the tuple of public parameters in SIDH. Suppose that* E_B *and the images* P'_A, Q'_A *are known and generated according to the key-generation algorithm in SIDH. Then, given*

- *a curve* E*,*
- *a basis pair* $P, Q \in E[B]$*,*
- *a curve* $\overline{E}$*,*

determine whether the tuple $(E_0, E, E_B, \overline{E})$ *is a valid SIDH tuple, in the sense that there is a map* $\varphi : E_0 \longrightarrow E$ *of degree dividing* A *and kernel generator* R_A*, which sends* P_B *to* P*,* Q_B *to* Q *and such that* $\overline{E} \cong E_0/\langle R_A, R_B \rangle$*.*

The KEM SIKE and its concrete parameters. The protocol submitted to NIST's competition is a key-encapsulation algorithm derived from SIDH, called SIKE. It is obtained from the KEX $SIDH$ in two steps.

1. *KEX* $\longrightarrow$ *PKE:* an intermediate IND-CPA public-key encryption scheme is derived from the key exchange in the following standard way. In order to encrypt a message m under Bob's public key, Alice first generates a random key pair $(\mathsf{sk}_r, \mathsf{pk}_r)$, that she uses to complete the key exchange and obtain a secret j-invariant. She then hashes the j-invariant and XORes the result with the message m, obtaining c. The ciphertext consists of (pk_r, c). Bob can complete the key-exchange using pk_r, hash the resulting j-invariant and XOR it with c, thus decrypting to m.
2. *PKE* $\longrightarrow$ *KEM:* an IND-CCA1 key-encapsulation mechanism is obtained by applying the Hofheinz, Hövelmanns and Kiltz transform [HHK17] to the intermediate IND-CPA PKE from the previous step. Here, Alice encrypts a random message m under Bob's public key pk_B (taking the $H(m \parallel pk_B)$ as the secret key in the encryption). She then computes the shared key as the hash of m concatenated with the ciphertext, and sends the ciphertext to Bob. Bob decrypts and checks whether the public key in the encryption (note that now he can recompute r and repeat the encryption himself) matches the public key in the ciphertext. Finally, he computes the key either as the hash of m concatenated with the ciphertext, or as the hash of a random string concatenated with the ciphertext.

In Table 3.1 we indicate the values of p, e_A and e_B in SIKE that were recommended for different security levels.

NIST's security level	Parameter set	p
1	SIKEp434	$2^{216} \cdot 3^{137} - 1$
2	SIKEp503	$2^{250} \cdot 3^{159} - 1$
3	SIKEp610	$2^{305} \cdot 3^{192} - 1$
5	SIKEp751	$2^{372} \cdot 3^{239} - 1$

Table 3.1: Recommended SIKE parameters for different NIST security levels.

3.3 CSIDH

As we said in the introduction, Castryck, Lange, Martindale, Panny and Renes managed in [CLM+18] to successfully adapt the CRS construction to the supersingular setting, considering supersingular elliptic curves over $\mathbb{F}_p$ instead of ordinary curves. In this section we describe their CSIDH protocol, whose underlying algebraic foundations have been provided in Section 2.3, more specifically in Section 2.3.2. Before getting into the protocol details, we need to describe how to sample an ideal and compute its action, because it will play the role of secret key.

3.3.1 Ideal sampling and evaluation

As we said in Section 2.3.2, CSIDH is based on the action of the class group of $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$ on the isomorphism classes of supersingular elliptic curves. We have defined ideals and their correspondence to isogenies, but we still need to define practical ways to sample and evaluate them.

In order to properly sample uniformly at random from $cl(\mathbb{Z}[\sqrt{-p}])$, one would need to compute its exact structure. An algorithm by Hafner and McCurley [HM89] can solve this task in subexponential time, but becomes impractical for orders of large discriminants (as in the CSIDH case) if one would need to change the order. By heuristically assuming that the ideals $\mathfrak{l}_i = (\ell_i, \pi_p - \lambda^2)$ do not have very small order and are evenly distributed in the class group, the authors of [CLM+18] argue that two ideals produced as $\mathfrak{l}_1^{e_1}\mathfrak{l}_2^{e_2}\cdots\mathfrak{l}_n^{e_n}$ for uniformly random e_i lie in the same class with negligible probability. In practice, the exponents e_i are sampled from $\{-m, \dots m\}$, where $m \in \mathbb{N}$ is such that $2m + 1 \geq \sqrt[n]{\#cl(\mathcal{O})}$.

Now that we have an ideal of the form $\prod \mathfrak{l}_i^{e_i} \in cl(\mathcal{O})$, we translate it to an isogeny in order to evaluate its action on a curve in $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ (the set of elliptic curves defined over $\mathbb{F}_p$ whose endomorphism ring is isomorphic to $\mathcal{O}$

[2] $\lambda \in \mathbb{Z}/\ell\mathbb{Z}$ is an eigenvalue of π_p in the ℓ-torsion subgroup.

such that the Frobenius endomorphism π_p corresponds to π). The fastest way to evaluate the action of $\mathfrak{l}_i$ on E is to find a basis of the torsion subgroup $E[\ell_i]$ (over some extension of $\mathbb{F}_p$ if necessary), compute the eigenspaces of π and apply Vélu's formulae to a basis point of the eigenspace to obtain the codomain. The optimal scenario is when the torsion subgroup is defined over a small extension field (more specifically, when $\lambda = 1$ we can work with points defined over $\mathbb{F}_p$), and when $p/\lambda = -1$ (so that the eigenspace of $\pi_p \mod \ell$ is defined over $\mathbb{F}_p$ if we work with Montgomery curves, and we Vélu's formulae remain efficient). Note that these conditions can be enforced by carefully choosing the base prime p, as successfully done in CSIDH.

We now describe how ideal evaluation works. Let $(e_1, e_2, \dots e_n)$ be the list of exponents in $\prod \mathfrak{l}_i^{e_i}$; while some $e_i \neq 0$,

1. sample $x \xleftarrow{\$} \mathbb{F}_p$ representing an $\mathbb{F}_p$-rational point $P \in E$ (curves in Montgomery form allow for nice and efficient x-only arithmetic);
2. use P to compute as many isogenies as possible: define k as the product of all ℓ_i dividing the order of P, compute $Q = [(p+1)/k]P$, and for each $\ell_i | k$,
   a) compute the isogeny φ of kernel $[k/\ell_i]Q$;
   b) push Q through φ, update the curve to the image curve $E \leftarrow \varphi(E)$, reduce the corresponding e_i by 1 (or increment by 1 if $e_i < 0$) and update $k \leftarrow k/\ell_i$;
3. once P has been exhausted and at least one e_i is still different from 0, repeat from 1.

3.3.2 The protocol

Now that we know (or at least have some intuition on) hot to sample and evaluate random isogenies by sampling and translating ideals in $cl(\mathcal{O})$, let us define how the protocol works. We start with a careful parameter selection that guarantees efficient sampling and translation of ideals into isogenies. Then we see how keys are generated and how the key-exchange is performed.

Parameters. Choose several distinct and small odd primes ℓ_i such that $p = 4 \cdot \ell_1 \cdot \ell_2 \cdot \dots \cdot \ell_n - 1$ is a large prime. In practice, the proof-of-concept implementation that meets NIST's security level 1 uses 74 primes ℓ_i: the first 73 are the smallest odd primes, while $\ell_{74} = 587$. The fixed starting curve is the supersingular elliptic curve in Montgomery form $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$, with endomorphism ring isomorphic to the order $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$. A positive integer m such that $2m + 1 \geq \sqrt[n]{\#cl(\mathcal{O})}$ is fixed.

Key generation. Alice's secret key sk_A is a tuple $(a_1, \dots, a_n)$, where each a_i is sampled uniformly at random from $\{-m, \dots, m\}$. This tuple corresponds to the ideal class $\mathfrak{a} = \mathfrak{l}_1^{a_1}\mathfrak{l}_2^{a_2} \cdots \mathfrak{l}_n^{a_n} \in cl(\mathcal{O})$. The public key pk_A is the Montgomery coefficient $A \in \mathbb{F}_p$ of the elliptic curve $E_\mathfrak{a} := \mathfrak{a} * E_0 : y^2 = x^3 + Ax^2 + x$, obtained

by applying the action of $\mathfrak{a}$ to the curve E_0. Analogously, Bob randomly picks a secret key $\mathsf{sk}_B := (b_1, \ldots, b_n) \xleftarrow{\$} \{-m, \ldots, m\}^n$ and computes the public key $\mathsf{pk}_B = B$, the Montogmery coefficient of the curve $E_{\mathfrak{b}} := (\mathfrak{l}_1^{b_1} \cdots \mathfrak{l}_n^{b_n}) * E_0$.

Shared key computation. The only step left for Alice to compute the shared key is to evaluate the action of her secret ideal $\mathfrak{a}$ on $E_{\mathfrak{b}}$; analogously, Bob will evaluate the action of $\mathfrak{b}$ on $E_{\mathfrak{a}}$. They will both share the secret $\mathfrak{a} * E_{\mathfrak{b}} = E_{\mathfrak{ab}} = \mathfrak{b} * E_{\mathfrak{a}}$.

If we drew the key exchange diagram, we would obtain something shaped as in Figure 3.1, with the crucial difference that no torsion point images are needed to enable the key exchange, which can now be non-interactive. The reason is the existence of the commutative action of the ideal class group, a strong point of resemblance to the classical Diffie-Hellman protocol. More importantly, the lack of torsion point information seems enough to prevent the applicability of Castryck and Decru's attack to CSIDH.

Security. As for SIDH, we can state a computational problem that should secure against key-recovery attacks, and a decisional problem that secures the key-exchange; what follows is formulation of the former one.

Definition 3.3.1 (CSIDH key-recovery)**.** *Given two supersingular elliptic curves E_0, E defined over $\mathbb{F}_p$ with the same F_p-rational endomorphism ring $\mathcal{O}$, find an efficient representation of an ideal class $[\mathfrak{a}]$ of $\mathcal{O}$ such that $E_{\mathfrak{a}} = E$.*

3.4 KLPT

A case in which dealing with quaternion algebras allows for tackling isogeny-related problems is the application of the KLPT algorithm [KLPT14]. The motivation behind this work lies in the equivalence of categories provided by the Deuring correspondence: finding an ℓ-isogeny between two supersingular elliptic curves E_1, E_2 is analogous to determining an ℓ-power $\mathcal{O}_1\mathcal{O}_2$-ideal, where $\mathcal{O}_1 \simeq \mathrm{End}(E_1)$ and $\mathcal{O}_2 \simeq \mathrm{End}(E_2)$. More to the point, their algorithm was designed as an attack to the Charles-Goren-Lauter hash function [CLG09], and aims to solve the following problem:

Definition 3.4.1 (Quaternion ℓ-isogeny path problem)**.** *Given a prime p, a maximal order $\mathcal{O}_0$ of $B_{p,\infty}$ and a left $\mathcal{O}_0$-ideal I, find an equivalent left $\mathcal{O}$-ideal $J \sim I$ of norm ℓ^e for some $e \in \mathbb{N}$.*

Under the Deuring correspondence, we have that $\mathcal{O}_0 \simeq \mathrm{End}(E_0)$ and the left $\mathcal{O}_0$-ideal I corresponds to an isogeny $\varphi_I : E_0 \longrightarrow E$ of any degree; the problem in Definition 3.4.1 is equivalent to that of finding a ℓ^e-isogeny $\phi_J : E_0 \longrightarrow E$. Definition 3.4.1 can be easily generalised to the powersmooth-norm case, which is the one we focus our attention on.

First of all, we tip our hats at the paper's authors, since the KLPT algorithm is quite complicated to get the first (and second, in most cases third) time one wraps their mind around it. The many pieces and sub-algorithms seem to

magically fit, and understanding all the steps does not necessarily mean getting why the algorithm works. Since it is science after all, let us try to explain how and why each step works, starting from what we want and how to get it. The description is based on our implementation of the KLPT algorithm provided in Paper 4.

Step 0: the setup. On input

- p, the field characteristic
- $\mathcal{O}_0$, a special extremal order in the quaternion algebra $B_{p,\infty}$
- I, a left $\mathcal{O}_0$-ideal
- $\{i, j\}$, generators in the basis $\{1, i, j, ij\}$ of $B_{p,\infty}$

KLPT outputs a left $\mathcal{O}_0$-ideal J equivalent to I, with powersmooth norm t.

Based on [KLPT14, Lemma 5], we know that the ideal

$$\chi_\alpha(I) := I\frac{\overline{\alpha}}{\mathrm{nrd}(I)}$$

has norm $\mathrm{nrd}(\alpha)/\mathrm{nrd}(I)$ for any quaternion $\alpha \in I$, and it is equivalent to I. Thus, the ultimate goal becomes that of finding $\beta \in I$ of norm $t \cdot \mathrm{nrd}(I)$ for some powersmooth integer t, so that $J := I\overline{\beta}/\mathrm{nrd}(I)$ is an equivalent ideal of norm t. The authors approach a solution by taking several intermediate steps that involve solving certain norm equations over $\mathcal{O}_0$. Unfortunately, little can we do in the general case. The problem is made solvable by restricting to some special (literally) cases, trying to reshape the generic equation into one that we can handle. In particular, Cornacchia's algorithm [Cor08] that solves Diophantine equations of the form $x^2 + y^2 = m$ will be crucial to get us by.

Remark 3.4.2. For the original version of the KLPT algorithm, the norm of the output ideal was expected to be around $p^{7/2}$. The algorithm was later improved in [DLW22] to output ideals of norm $t \approx p^{5/4}$.

Remark 3.4.3. The order $\mathcal{O}_0$ is less general than it may seem. In fact, it must be **special p-extremal**, which means that it must contain a subring $\mathbb{Z}\langle\omega_1, \omega_2\rangle$ with $\mathrm{nrd}(\omega_1) = q, \mathrm{nrd}(\omega_2) = p$ for q coprime to p. This is required essentially to afford solving norm equations for any element $\alpha = x + y\omega_1 + z\omega_2 + k\omega_1\omega_2$ in the subring, since its norm $\mathrm{nrd}(\alpha) = (x^2 + qy^2) + p(z^2 + qk^2)$ can be manipulated to resemble a Diophantine equation. One can trivially solve the general case for a non-special order $\mathcal{O}_1$ by forcing the solution via a special $\mathcal{O}_0$ as in Figure 3.2, at the cost of allowing large norm outputs.

Step 1: computing an ideal I_δ of prime norm p_δ equivalent to I. With equivalentPrimeIdeal, one searches for an ideal I_δ equivalent to I with a small prime norm. One may ask why shifting to another ideal; the motivation lies in Step 3. Since the generic norm equation cannot be efficiently solved, we want to perform some modular reductions and reduce it to a Cornacchia-friendly

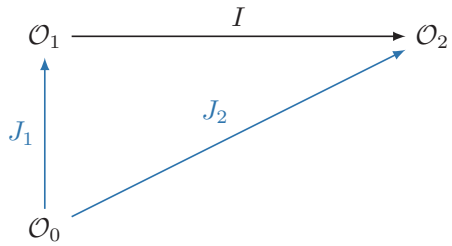

Figure 3.2: The trivial strategy to generalise KLPT for non-special orders: use KLPT from a special $\mathcal{O}_0$ to compute J_1 and J_2, then output ${J_1}^{-1}J_2$.

equation. Thus the need for an ideal I_δ of prime norm p_δ, in which we can reduce norm equations modulo p_δ.

In light of [KLPT14, Lemma 5], we shift the problem to that of finding $\delta \in \mathcal{O}_0$ of norm $\mathrm{nrd}(I) \cdot p_\delta$ for a small[3] prime p_δ. Since there are sufficiently many such quaternions (especially if we let p_δ be large enough) and generating them is also quite efficient if the basis of I is short, one can

1. compute a Minkowski-reduced basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of I
2. compute a bound $m \in \mathbb{N}$
3. generate quaternions using coefficients in $[-m, m]$ until one is met with norm $\mathrm{nrd}(I) \cdot p_\delta$ for some prime $p_\delta < p^{5/9}$.

Let us define $I_\delta := \chi_\delta(I)$ the resulting I-equivalent ideal of prime norm p_δ.

Notice how the lattice structure of ideals has proven itself to be very handy. The integer m must be carefully chosen in such a way that $[-m, m]$ contains sufficiently many primes to guarantee a good chance of finding one. See [KLPT14, Section 3.1] for some heuristic arguments; in Paper 4 we set $m = \max(\lfloor \frac{\log p}{10} \rfloor, 7)$.

Step 1.2 (optional): looking for a target powersmooth norm t. Given the state of the art on isogeny computation and evaluation, it is more efficient to compute a long sequence of small-degree isogenies than to compute a few high-degree ones. For this reason, we would like our final ideal J to be decomposable into a sequence of two-by-two compatible ideals, each of norm $\ell_i^{e_i}$ for some distinct small primes ℓ_i. In the original algorithm, t was supposed to be given or to be computed along the way as $t = t_1 \cdot t_2$, first finding a suitable t_1 in Step 2.1 and then a suitable t_2 in Step 2.3. A key to our improvement in Paper 4 is that we allow the isogenies to be defined over a larger extension fields than the quadratic one. In some cases, it is indeed more efficient to compute more isogenies of small base degree allowing further extensions than it is to reach the desired degree via isogenies of larger base degree. Not to lose efficiency, we

[3] In our implementation, we settle for any prime $p_\delta < p^{5/9}$

can compute t while keeping track of the smallest necessary extension for each prime ℓ_i, and we fine-tune the parameters to empirically reach the optimal balance. See Paper 4 for details on the computation of t.

Step 2: looking for an element $\beta \in I_\delta$ of powersmooth norm $t \cdot p_\delta$. We now work in I_δ and look for an element of good norm there: since I_δ and I are equivalent, we can use elements in I_δ instead of those in I to compute ideals equivalent to I.

First, we rearrange the elements of I_δ in order to simplify our quest. Since all elements in I_δ have norm multiple of p_δ by definition, we keep generating random elements $\alpha \in I_\delta$ until we find one that satisfies $\gcd(\mathrm{nrd}(\alpha), p_\delta^2) = p_\delta$. This allows us to write the ideal as $I_\delta = \mathcal{O}_0 \cdot p_\delta + \mathcal{O}_0 \cdot \alpha$, or $I_\delta = \mathcal{O}_0\langle p_\delta, \alpha\rangle$ in short.

What we just did is incredibly useful, because we can now try to find β by solving norm equations that would be unapproachable otherwise.

- **Step 2.1**: looking for an element $\gamma \in \mathcal{O}_0$ of norm $p_\delta \cdot t_1$ for some powersmooth t_1.

  We start off in $\mathcal{O}_0$ because norm equations can be solved quite easily in there, being the order special: given any quaternion $\gamma = a+b\cdot i+c\cdot j+d\cdot ij$ in $\mathcal{O}_0$, its norm is $\mathrm{nrd}(\gamma) = a^2 + b^2 + p \cdot c^2 + p \cdot d^2$. Finding an element of norm $p_\delta \cdot t_1$ thus amounts to solving the norm equation $p_\delta \cdot t_1 = a^2 + b^2 + p \cdot (c^2 + d^2)$. A solution γ can be produced by randomly sampling pairs of small integers (c, d) until one successfully solves the Diophantine equation

  $$a^2 + b^2 = p_\delta \cdot t_1 - p \cdot (c^2 + d^2)$$

  using Cornacchia's algorithm. The guessing game presented above is implemented in the $\mathsf{RepresentInteger}(p_\delta \cdot t_1, p, i, j)$ algorithm.

- **Step 2.2**: looking for integers c, d satisfying $\gamma \cdot (c \cdot j + d \cdot ij) = \alpha \mod p_\delta\mathcal{O}_0$, where α is a quaternion such that $I_\delta = \mathcal{O}_0\alpha + \mathcal{O}_0 p_\delta$.

  Now we have $\gamma \in \mathcal{O}_0$ with a good norm, but we are actually looking for an element in I_δ of good norm. Being I_δ a left $\mathcal{O}_0$-ideal, we want to find an element in I_δ to multiply to the right of γ. By first taking the operations mod $p_\delta\mathcal{O}_0$, we look for integers c, d such that $\gamma\cdot(c\cdot j+d\cdot ij) = \alpha \mod p_\delta\mathcal{O}_0$. This is implemented in the function $\mathsf{IdealModConstraint}(I_\delta, \gamma, \alpha, p_\delta, p, i, j)$.

- **Step 2.3**: looking for $\nu \in \mathcal{O}_0$ with powersmooth norm t_2 and an integer h not divisible by p_δ such that $\nu = h \cdot (c \cdot j + d \cdot ij) \mod p_\delta\mathcal{O}_0$.

  Once again, we translate this problem to a norm equation, and solve it step-by-step with different modular reductions:

  1. Write $\nu = h \cdot (c \cdot j + d \cdot ij) + p_\delta(x + y \cdot i + z \cdot j + k \cdot ij)$. Switching to the norm equation

     $$t_2 = p_\delta^2(x^2 + y^2) + p \cdot \big((h \cdot c + z \cdot p_\delta)^2 + (h \cdot d + k \cdot p_\delta)^2\big), \quad (3.1)$$

     our goal becomes that of finding h, x, y, z, k.

2. Taking Equation (3.1) modulo p_δ, we obtain

$$p \cdot h^2 \cdot (c^2 + d^2) = t_2 \mod p_\delta. \tag{3.2}$$

   By making sure that t_2 is a quadratic residue modulo p_δ (possibly multiplying t_2 by some small primes), we solve the equation for h.

3. Now that we have h from Equation (3.2), we can take Equation (3.1) modulo p_δ^2 and solve for (c, d)

$$p \cdot h^2 \cdot (c^2 + d^2) + 2 \cdot p \cdot h \cdot p_\delta \cdot (z \cdot c + k \cdot d) = t_2 \mod p_\delta^2. \tag{3.3}$$

   A solution is obtained by sampling either c or d at random and solving for the other variable.

4. Rewrite Equation (3.1) as a Diophantine equation:

$$x^2 + y^2 = \frac{(t_2 - p \cdot \left((h \cdot c + z \cdot p_\delta)^2 + (h \cdot d + k \cdot p_\delta)^2\right)}{p_\delta^2}$$

   and solve it with Cornacchia's algorithm. If no solution can be found, repeat from Step 3.1 to find different values for (c, d).

This is implemented in $\mathsf{StrongApproximation}(p_\delta, p, t_2, c, d, i, j)$.

Step 3: compute $J \sim I$ of powersmooth norm. Finally, we can compute $\beta = \gamma \cdot \nu$, which lies in I_δ since ν does, and has norm $\mathrm{nrd}(\beta) = \mathrm{nrd}(\gamma \cdot \nu) = \mathrm{nrd}(\gamma) \cdot \mathrm{nrd}(\nu) = p_\delta \cdot t_1 \cdot t_2$. Through β we can finally compute $J = I_\delta \cdot \overline{\beta}/\mathrm{nrd}(I)$ by multiplying each basis element of I_δ by $\overline{\beta}/\mathrm{nrd}(I)$.

3.5 SQISign

The SQISign [DKL+20b, FLW22] protocol is an isogeny-based digital signature scheme, classically obtained from an identification protocol via the Fiat-Shamir transform. Our main goal for this section is to provide the reader with an overview of this construction, providing a description of its algorithms based on the preliminaries in Chapter 2 and on the KLPT algorithm in Section 3.4. Let us start off with the identification protocol.

Parameters and Key generation. Choose a prime p, a supersingular elliptic curve E_0 over $\mathbb{F}_p$ of known special extremal endomorphism ring $\mathcal{O}_0$ and an integer $d = 2^e$ such that e is the diameter of $\mathcal{G}_p(\ell)$. For a given security level λ, choose an odd number d_{ch} of λ bits.

To generate a key pair, one first selects a left $\mathcal{O}_0$-ideal I of small prime uniformly at random. Then, a left $\mathcal{O}_0$-ideal J equivalent to I is computed via the KLPT algorithm, where J has reduced norm equal to a power of 2. Finally, the secret isogeny $\varphi_{\mathsf{sk}} : E_0 \longrightarrow E_A$ corresponding to J is computed, and the public key is set to $\mathsf{pk} = E_A$.

Interactive Σ-protocol. The interactive part of the identification protocol goes as follows:

1. *Commitment*: the prover $\mathcal{P}$ samples at random a secret isogeny $\varphi_{\mathsf{com}} : E_0 \longrightarrow E_{\mathsf{com}}$ and sends E_{com} to the verifier $\mathcal{V}$.
2. *Challenge*: $\mathcal{V}$ sends a cyclic d_{ch}-isogeny $\varphi_{\mathsf{ch}} : E_{\mathsf{com}} \longrightarrow E_{\mathsf{ch}}$ to $\mathcal{P}$. The implicit request is to produce an isogeny from E_{com} to E_{ch}.
3. *Response*: a honest prover is able to compute a d-isogeny $\varphi_{\mathsf{resp}} : E_A \longrightarrow E_{\mathsf{ch}}$ equivalent to $\varphi_{\mathsf{ch}} \circ \varphi_{\mathsf{com}} \circ \hat{\varphi}_{\mathsf{sk}}$, and send φ_{resp} to $\mathcal{V}$.
4. *Verification*: $\mathcal{V}$ accepts if φ_{resp} is a d-isogeny from E_A to E_{ch} and $\hat{\varphi}_{\mathsf{ch}} \circ \varphi_{\mathsf{resp}}$ is cyclic.

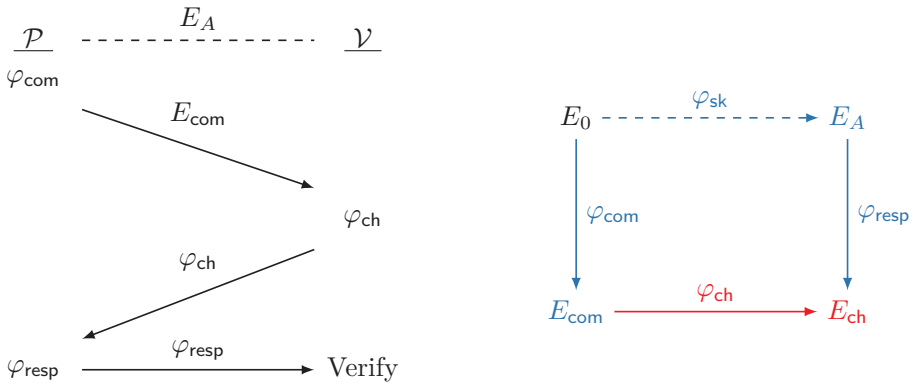

Figure 3.3: The identification protocol at the basis of SQISign. On the left the protocol flow, on the right the corresponding movements on the isogeny graph, with elements in blue produced by $\mathcal{P}$ and those in red produced by $\mathcal{V}$.

A very crucial and delicate operation in the protocol is the response computation. Having just talked about the KLPT algorithm in Section 3.4, a question may arise: since we need to compute an isogeny equivalent to $\varphi_{\mathsf{ch}} \circ \varphi_{\mathsf{com}} \circ \hat{\varphi}_{\mathsf{sk}}$, why don't we translate the three isogenies to three ideals $I_{\mathsf{ch}}, I_{\mathsf{com}}, I_{\mathsf{sk}}$ and use KLPT to compute an ideal equivalent to $I_{\mathsf{ch}} \circ I_{\mathsf{com}} \circ \overline{I}_{\mathsf{sk}}$? As noted in [DKL+20b], a direct application of the KLPT algorithm would leak information on the secret key. Thus, many improvements and ad-hoc adjustments were made to the generalised KLPT algorithm, in order for it to work for arbitrary maximal orders instead of special p-extremal orders. This effort lead to the SigningKLPT algorithm [DKL+20b, Algorithm 5], which takes the left $\mathcal{O}$-ideal $I_{\mathsf{ch}} \circ I_{\mathsf{com}} \circ \overline{I}_{\mathsf{sk}}$ and outputs an equivalent ideal J of prime-power norm (exactly 2^e).

The result was not flawless. First of all, a distinguishability issue was spotted in [FLW22], which broke the zero-knowledge property of the digital signature. In the first paper, the authors formulate a computational assumption on the indistinguishability of SigningKLPT outputs from random ideals of equal norm. This assumption fails due to the subalgorithm RepresentInteger (recall Section 3.4, Step 2.1), which outputs quaternions from a space smaller than necessary to guarantee indistinguishability at later steps. The weakness was not critical enough to lead to a full key-recovery attack, but sufficient to invalidate the computational HVZK property of the scheme. The indistinguishability was restored in [FLW22] with a modified RepresentInteger algorithm. Secondly, SQISign's main bottleneck lied in the large norm of the endomorphisms output by SigningKLPT, which lead to a signing time of 2 seconds. The efficiency of the signing algorithm was improved in [FLW22] by allowing a variant of KLPT to work with endomorphisms of smaller norm. These and many other small modifications prompt a different protocol version, and this is the reason behind the very general protocol description provided above. We refer the reader interested in all the juicy details on all the other algorithms, the parameter selection and the timing analysis to the two original works [DKL+20b, FLW22], or to Antonin Leroux' excellent PhD thesis.

Chapter 4

Summary of Results Contributing to the Thesis

Self-doubt takes its toll
Imposter fears fill my mind.
Will my work suffice?

December 2022

This is the conclusive chapter of this thesis. In Section 4.1 we list the research questions that drove the research presented in this manuscript. We then present the results hereby collected, summarising them in Section 4.2 and linking them to the relevant research questions. Given the apparently unusual proportion between published and unpublished papers, we wrote Section 4.3 to explain why some of our contributions cannot be published as they are, in light of the recent attacks on SIDH.

4.1 Research questions

The scope of this PhD thesis was not entirely defined at its early stage, since it originates from a generic call for an investigation of post-quantum primitives. Due to their broad and open nature, the research questions were refined only in fieri, once isogeny-based cryptography gained the upper hand in the candidate's interests. Here is the final set of research questions that drove the investigation presented in this document.

RQ 1: can we prove tight reductions on isogeny-based schemes? There is an incredibly large amount of classical and post-quantum protocols available in the literature. When choosing one over another, theoretically-sound parameters should always be considered. In fact, the parameter's size that guarantees a certain security level depends on the security proof of the protocol under examination. When writing a security proof, one must define

- a security model, in which the adversarial capabilities (represented as various types of queries that it can make) are described and limited at various extents;
- a sequence of games that result in a *security reduction*, a set of operations that turn an adversary into a hard-problem solver.

The "quality" of a reduction can be measured by computing its security loss: if $t_{\mathcal{A}}$ and $\epsilon_{\mathcal{A}}$ are the adversarial running time and success probability respectively, and $t_{\mathcal{B}}$ and $\epsilon_{\mathcal{B}}$ are the reduction's running time and success probability, we define the *security loss* L via the equation

$$\frac{t_{\mathcal{A}}}{\epsilon_{\mathcal{A}}} = L \, \frac{t_{\mathcal{B}}}{\epsilon_{\mathcal{B}}}.$$

Tight reductions have a constant security loss.

In some cases, however, it is impossible to obtain a tight reduction. In a *simple scheme* the adversary is run only once, in comparison to other protocols which use the Forking Lemma in order to run multiple copies of the adversary. A linear loss in the number of participants to the protocol is inherent to simple schemes, while applying the Forking Lemma tends to result in non-tight proof. Whenever a constant tightness loss cannot be achieved by construction, we focus on *optimal tightness* instead, aiming to prove that non-constant tightness loss cannot be lowered in order of magnitude.

This question is interesting both in the classical and in the post-quantum scenario, but we focused our attention on the case of isogeny-based post-quantum protocols. In fact, many post-quantum schemes incur in significant security losses, and some protocols do not even have a security proof.

RQ 2: how sound are the assumptions underlying some computational problems in isogeny-based cryptography? Being still "in its infancy", isogeny-based cryptography relies on problems that still need to be crypt-analysed and to stand the test of time. Recent attacks [CD22, MM22, Rob22] on SIDH, outlined at the end of this chapter, are clear evidence that we need to better understand there problems and assess their strength. More specialised work can be done considering specific assumptions made when proving security of a protocol: since this research field is progressing at a fast pace, new information on the algebraic and geometric structures of isogeny graphs can strengthen or hinder some conjectures. In addition to a cryptographic analysis, it is therefore important to follow an algebraic/geometric/number theoretic approach in assessing the security of cryptographic schemes, particularly in isogeny-based cryptography, an area that is still relatively youthful compared to other post-quantum alternatives.

RQ 3: can we obtain faster isogeny-based cryptography? One of the major drawbacks in choosing isogeny-based cryptographic schemes is still their relative slowness compare to other post-quantum alternatives. There are two ways that lead to improvements on existing schemes:

- *algorithmic improvements.* As often happens in algorithmics, several techniques and tricks can be implemented to speed up computations or reduce sizes. Some are of mathematical lineage, others are more cryptographic in nature, others again purely stem from computer science.

- *security proof improvements.* Correlated with our first research question, better performances can result from having a tighter security proof. In fact, reducing the tightness loss has the positive effect of lowering the parameter sizes while maintaining the same security confidence.

4.2 Contributions

The research done throughout this thesis resulted in several contributions to the field of isogeny-based cryptography:

- in terms of cryptographic protocols, we provide a CSIDH-based key-exchange protocol and an SIDH-based signature scheme. With the former, we construct the first provably secure CSIDH-based KEX with a special focus on the tightness of the security reduction. With the latter, we design an SIDH-based signature scheme, and apply several optimisations to shorten its signature size and running time. Despite not being novel in the literature, these optimisations are not often taken into account in the design of a digital signature scheme. Morevoer, we extend the unbalanced challenge space analysis to the ternary challenge case, proving results of novel and independent interest in cryptography.
- in terms of generic knowledge on supersingular isogeny graphs, we analyse the occurrence of cycles formed by two same-degree isogenies. After assessing how they affect the security of the SIDH identification protocol, we argue that these cycles are few under some reasonable assumptions. In light of our findings, the original SIDH identification protocol was not secure anymore. Nevertheless, we salvage it without any modifications by simply providing a new knowledge extractor, which never fails despite the existence of (arguably few) cycles.. Our study on the supersingular isogeny graph is of independent interest, even though our applications are tailored to the deceased SIDH protocol.
- in terms of algorithmic improvements, we speed up the computation of the Deuring correspondence in general characteristic, while previous work focused on specially crafted primes. This has two main impacts: we tighten up the security reductions that are based on KLPT-like algorithms, and we provide faster (sage) implementation of algorithms for computational number theorists and cryptographers working with endomorphism rings of supersingular elliptic curves.

In the following we summarise the contributions in the five papers presented in chronological order, while the authors are listed in alphabetical order.

Paper 1 - Practical Isogeny-Based Key-exchange with Optimal Tightness. Bor de Kock, Kristian Gjøsteen, Mattia Veroni. *Selected Areas in Cryptography, 2020.*

Our motivation behind Paper 1 is to be found in RQ1. Our goal in this work was to adapt a construction by Cohn-Gordon et al. [CCG+19] to supersingular elliptic curves over $\mathbb{F}_p$, obtaining an isogeny based authenticated KEX protocol with an optimally tight proof. After studying the problem, we looked at SIDH and CSIDH for random self-reducibility, a crucial point in the tightness of the security proof. We were able to exploit the Diffie-Hellman-like structure of CSIDH to build the post-quantum authenticated KEX scheme we aimed for. Compared to previous isogeny-based authenticated key-exchange protocols, our scheme is rather simple, its security relies only on the CSIDH version of the Strong Diffie-Hellman problem and it has optimal communication complexity for CSIDH-based protocols. Our security proof crucially depends on the re-randomizability of CSIDH-like problems, and carries on in the ROM.

Paper 2 - Collisions in Supersingular Isogeny Graphs and the SIDH-based Identification Protocol. Wissam Ghantous, Shuichi Katsumata, Federico Pintore, Mattia Veroni. *Submitted at Journal of Algebra in March 2022.*

With this paper we addressed RQ 2, testing the reliability of certain assumptions and questioning the security proof of the identification protocol based on SIDH. We analysed the security proofs available in the literature for the SIDH-based identification protocol, together with their effects on the security of the digital signatures obtained via the Fiat-Shamir transform. All such proofs consider the same extraction algorithm when it comes to proving the special-soundness property. The scope of this extractor is to output the witness for a statement that appears in two valid transcripts with equal commitments and different challenges.

Our doubts on the reliability of such extractor were raised after we run some tests on SIDH (instantiated with toy-example parameters), that showed how this was not always true. We were then able to produce a few counterexamples with parameters of cryptographic size that invalidated the property claim. The general argument, in fact, fails due to the existence of some collisions (cyclic isogenies with different kernels and equal domain and codomain curves) in supersingular isogeny graphs.

After detecting the fault, we began to study the existence of such collisions from a theoretical point of view, discussing their impact on the security of the SIDH-based digital signatures. We could argue that such cycles occur in negligible quantity, after making some mild assumptions. We also took a different approach to restore the security of the identification protocol: relying on the Generalised Riemann Hypothesis, we introduced a new extractor for which we rigorously proved the special soundness property.

Paper 3 - Sigh: faster and shorter SIDH signatures. Wissam Ghantous, Federico Pintore, Mattia Veroni. *Cannot be published as is, due to the recent attacks on* SIDH*.*

In this paper we tackled RQ 3 in light of our results from Paper 2. We presented an isogeny-based signature scheme whose security relied on the computational supersingular isogeny problem. The protocol was obtained by applying the Fiat-Shamir transform to the SIDH identification protocol, and then performing a series of optimisations both on the signature size and on the signing algorithm. Compared to other SIDH-based signature schemes, our protocol allowed for faster and smaller signatures, since we relied on the original SIDH identification scheme (proven secure in Paper 2), rather than on modified versions such as in [DDGZ21], that aims to restore special soundness by changing the protocol at the cost of extra computation.

We were working on the implementation of our scheme when the devastating attacks on SIDH appeared in August 2022. Unfortunately, all attacks on SIDH apply to our scheme as well, since we make use of the torsion points images in the same fashion of SIDH. Further investigation on recent masking techniques [Fou22, Mor22] needs to be conducted before we can analyse their effects on our scheme, and the paper would need further polishing and proof-reading.

Paper 4 - Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic. Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, Mattia Veroni. *Available on ePrint, in submission to LuCaNT 2023.*

In this work, we design an algorithm to solve the constructive Deuring correspondence for general primes p, translating an ideal in the quaternion algebra ramified at p and ∞ into an isogeny. The fastest algorithms to compute such correspondence are the ones used in SQISign, but they work only for a very special family of primes. Instead, we apply several optimisations to speed up existing algorithms that work for more general primes than the ones carefully crafted in SQISign. The most significant gain comes from the observation that it is advantageous to allow for higher degree isogenies, lowering in this way the necessary extension field degree. We show the impact of this simple improvement (together with other, but less impactful, ones) by computing the Deuring correspondence for generic primes up to 75 bits. Additionally, our method further exploits the particular structure of the primes typically used in isogeny based cryptography; for example, choosing p such that $p^2 - 1$ has many small prime-power divisors.

The version attached to this thesis is a preliminary draft of the version that we will submit to LuCaNT in January 2023. There we plan on incrementing the size of the prime p where we solve the constructive Deuring correspondence to cryptographic size, and produce figures of different running time as $\log p$ increases.

Paper 5 - Efficiency of SIDH-based signatures (yes, SIDH). Wissam Ghantous, Federico Pintore, Mattia Veroni. *Pre-print version available on ePrint.*

In this note, we try to salvage some work from Paper 3 adapting the same techniques to a new SIDH-like identification protocol. Specifically, we assess the efficiency of a SIDH-based digital signature built on a recent identification protocol proposed by Basso *et al*, which we denote by Σ_{SEC}. Despite the devastating attacks against SIDH, the protocol Σ_{SEC} is secure, since it relies on a different (and more standard) isogeny-finding problem.

We apply some known cryptographic techniques to decrease the signature size, and propose minor optimisations to compute many isogenies from the same starting curve (and their kernels) in parallel. Our assessment confirms that the problem of designing a practical isogeny-based signature scheme remains largely open. In fact, we still get signatures of about 35KB for $\lambda = 128$, and we expect a significantly large signing time (probably in the order of tens of seconds). However, we concretely determine the current state of the art which future optimisations can compare to.

4.3 Attacks on SIDH and their effects on this thesis

Three major attacks [CD22, MM22, Rob22] have appeared over the past months compromising the security of SIDH-based protocols. In this section, we first hint on the Castryck–Decru [CD22] attack, with notation adapted on Section 3.2. We then highlight some countermeasures that have been recently proposed, closing with some final remarks on how this attacks have impacted this thesis.

4.3.1 Castryck-Decru's attack

Let us first recall the notation we introduced when we described the SIDH key-exchange, with the only difference that we consider only one coefficient for the secret keys (since this is the case analysed in the attack).

Let $A = 2^e \approx 3^f = B$ be two coprime integers such that $p = A \cdot B - 1$ is prime. Let $\langle P_A, Q_A \rangle = E_0[A]$ be A-torsion subgroup of E_0 with enclosed basis, and let $E_0[B] = \langle P_B, Q_B \rangle$ be the B-torsion subgroup with enclosed basis. Alice's secret key is computed by sampling n_A uniformly at random from $\mathbb{Z}/A\mathbb{Z}$ and computing the kernel $R_A = P_A + [n_A]Q_A$ corresponding to the isogeny $\varphi_A : E_0 \longrightarrow E_A$ of degree A. Her public key is the triplet (E_A, P'_B, Q'_B), where $P'_B = \varphi_A(P_B)$ and $Q'_B = \varphi_A(Q_B)$. Similarly, let $n_B \xleftarrow{\$} \mathbb{Z}/B\mathbb{Z}$ be the secret coefficient sampled by Bob to construct the kernel kernel $R_B = P_B + [n_B]Q_B$ corresponding to the isogeny $\varphi_B : E_0 \longrightarrow E_B$ of degree B. Let (E_B, P'_A, Q'_A) be Bob's public key, where $P'_A = \varphi_B(P_A)$ and $Q'_A = \varphi_B(Q_A)$.

Suppose that we are given Bob's public key (E_B, P'_A, Q'_A), and our goal is to recover Bob's secret isogeny of degree $B = 3^f$ (equivalently, the secret coefficient n_B). Retrieving such an isogeny completely breaks the key-exchange security, since we can then use φ_B to compute the shared key from Alice's

public key. The attack heavily relies on three pieces of information, two of which are the torsion points images and the fixed degree of Bob's isogeny. The third one is the known endomorphism ring of the initial curve E_0.

The high-level description of the attacks is quite short and easy to understand. Suppose there exists a decision oracle, which can tell right from wrong guesses on steps in Bob's secret walk. The strategy to reconstruct the secret scalar n_B digit-by-digit is simply to guess a step in Bob's path and query the oracle on its correctness. At every iteration, there are only three options for the next step, since there are four isogenies of degree 3 for each vertex but one is precluded by the non-backtracking property of SIDH's isogenies. With at most two oracle calls, one can thus determine the correct step and move onto the next one, with a maximum of $2 \cdot f$ oracle calls to recover the whole path.

The story gets complicated when we dive into the details of the decision oracle, which exploits a **glue-and-split** technique. First of all, one needs to be familiar with $(2,2)$-isogeny graphs of superspecial principally polarised abelian varieties of dimension 2 over $\overline{\mathbb{F}}_p$. We are far from experts in the field, but would like to try giving an overview of the reasoning behind the construction of the decision oracle.

Let $p > 3$ be prime, q be a power of p and fix $\ell = 2$. An **hyperelliptic curve $\mathcal{C}$ of genus 2** over $\mathbb{F}_q$ is defined as the locus of points satisfying $y^2 = f(x)$. If $\deg(f) = 6$, there are 15 possible factorisations of f as product of polynomials of degree 2. To each of these factorisations we associate a **Richelot isogeny**, which maps $\mathcal{C}$ either to another hyperelliptic curve $\mathcal{A}$ of genus 2 or to a product (E, E') of supersingular elliptic curves. Having curves and isogenies, we can construct a graph: the vertices are either products of supersingular elliptic curves or Jacobians of superspecial curves of genus 2 (whatever this means), and the edges are Richelot isogenies.

It turns out that the vertices (defined over $\mathbb{F}_{p^2}$) are not split 50/50 among the two types we have introduced: there are very few (roughly $\frac{p^2}{288}$) products of supersingular elliptic curves compared to the Jacobians of superspecial elliptic curves (roughly $\frac{p^3}{2880}$). A step represented by a Richelot isogeny is said to be **split** if it lands on a vertex of the form (E, E'), or **glued** if it lands on a Jacobian. Notice how, for primes p of cryptographic size, a random walk of polynomial length on the graph will end on a vertex of the form (E, E') with negligible probability.

On the other hand, under certain conditions on the factors of $f(x)$, a Richelot isogeny actually gives a rational map whose kernel is isomorphic to $(\mathbb{Z}/2\mathbb{Z}) \oplus (\mathbb{Z}/2\mathbb{Z})$, and thus can be written as a $(2,2)$-isogeny (think of it as a pair of 2-isogenies). Every Richelot isogeny of the form $(2^e, 2^e)$ has kernel isomorphic to $(\mathbb{Z}/2^e\mathbb{Z}) \oplus (\mathbb{Z}/2^e\mathbb{Z})$ in $(E, E')[2^e]$ and can be written as a chain of e $(2,2)$-isogenies. Kani [Kan97] gives us a way to determine whether the codomain of a Richelot isogeny is a Jacobian or a product of supersingular elliptic curves. First of all, Kani defines an **isogeny diamond configuration of order** N as a triplet (φ, G_1, G_2) of an isogeny $\varphi : E \longrightarrow E'$ and two disjoint (but for the point at infinity) subgroups $G_1, G_2 \subseteq \ker \varphi$ such that

$\deg\varphi = \#G_1 \cdot \#G_2$ and $N = \#G_1 + \#G_2$. Then the theorem: the codomain of a given (N, N)-subgroup of (E, E') is the product of supersingular elliptic curves if and only if it comes from isogeny diamond configuration of order N, i.e. it is generated by $(P, x\varphi(P))$ and $(Q, x\varphi(Q))$ for an N-torsion basis $\langle P, Q\rangle = E[N]$ and a suitable $x \in \mathbb{Z}$.

The authors of [CD22] brilliantly managed to turn these facts into a decision oracle to decide whether an attempted step in the key-recover attack is correct or not. Unfortunately, there is no shortcut to show the exact operation of the algorithm, so the curious reader must first conquer all the necessary mathematical background. A good starting point is the Isogeny-based cryptography school (https://isogenyschool2020.co.uk/schedule/), where many excellent researchers have contributed with several talks on this topic, particularly during week 9.

4.3.2 Possible countermeasures

Two recent pre-prints by Fouotsa [Fou22] and Moriya [Mor22] propose countermeasures to the devastating attack mentioned in Section 4.3. Being still in their preliminary versions at the time of writing, I will only highlight some aspects and techniques thereby contained.

In [Fou22], the countermeasure is to mask (by randomly scaling) the torsion points images, one of the two pieces of information crucial to the Castryck-Decru's attack. The idea is to rescale the torsion point images $\varphi_A(P_B), \varphi_A(Q_B)$ by a uniformly random integer $b \xleftarrow{\$} \mathbb{Z}/B\mathbb{Z}$, thus replacing $\varphi_A(P_B), \varphi_A(Q_B)$ with $[b]\varphi_A(P_B), [b]\varphi_A(Q_B)$ in Alice's public key. The key exchange is preserved: the point $[m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B)$ and the point $[m_B][b]\phi_A(P_B) + [n_B][b]\phi_A(Q_B) = [b]([m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B))$ generate the same subgroup, so the final curves E_{AB} and E_{BA} respectively computed by Bob and Alice are still isomorphic. At the same time, the Castryck-Decru attack loses its efficiency if used to recover the new secret isogeny $[b]\varphi_A$ of degree $p^{3/2}$, and loses it success probability if one tries to recover the original φ_A, due to the large number of small prime factors of B.

The modified SIDH key-exchange, that we hereby call Masked Torsion-Point SIDH ($\mathsf{MTP}_{\mathsf{SIDH}}$) is depicted in Figure 4.1, where

- $p = A \cdot B \cdot f \pm 1$ is a prime, where $A = \prod_{i=1}^{\lambda} p_i$ and $B = \prod_{i=1}^{\lambda} q_i$ are coprime integers, all p_i, q_i are small distinct primes, $A \sim B$ and f is a small cofactor;
- E_0 is a supersingular elliptic curve over $\mathbb{F}_{p^2}$, with torsion bases $\langle P_A, Q_A\rangle = E_0[A]$ and $\langle P_B, Q_B\rangle = E_0[B]$;
- $R_A = [m_A]P_A + [n_A]Q_A$ generates the kernel of Alice's secret isogeny $\varphi_A : E_0 \longrightarrow E_A$, and $R_B = ([m_B]P_B + [n_B]Q_B)$ generates the kernel of Bob's secret isogeny $\varphi_B : E_0 \longrightarrow E_B$;

- $(P'_B, Q'_B) = ([b]\phi_A(P_B), [b]\phi_A(Q_B))$ for a secret integer $b \in \mathbb{Z}/B\mathbb{Z}$ sampled at random by Alice, and $(P'_A, Q'_A) = ([a]\phi_B(P_A), = [a]\phi_B(Q_A))$ for a secret integer $a \in \mathbb{Z}/A\mathbb{Z}$ sampled at random by Bob;
- $R'_B = [m_b]P'_B + [n_B]Q'_B$ generates the kernel of $\phi'_B : E_A \longrightarrow E_{AB}$ computed by Bob, and $R'_A = [m_a]P'_A + [n_A]Q'_A$ generates the kernel of $\phi'_A : E_B \longrightarrow E_{BA}$ computed by Alice.

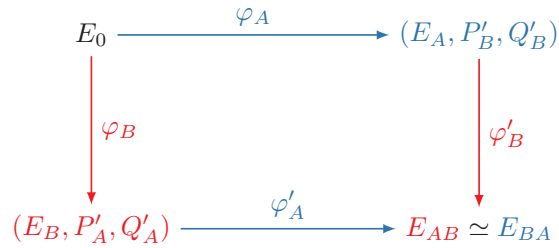

Figure 4.1: The $\mathsf{MTP}_{\mathsf{SIDH}}$ key exchange. Elements in blue are computed by Alice, those in red are computed by Bob.

In [Mor22], the countermeasure is to mask (by varying) the degree of the secret isogenies, in addition to the masking of the torsion point images (here necessary to guarantee the secrecy of the degree). Without getting lost in a forest of indices and superscripts, let us just say that with respect to [Fou22], the coprime integers A and B are still products of many small different primes, but each prime p_i appears with multiplicity e_i which is sampled at random (from values between 0 and 6 as per parameters suggested in the paper). The rest of the protocol, which we call Masked Degree SIDH ($\mathsf{MD}_{\mathsf{SIDH}}$), is basically identical to $\mathsf{MTP}_{\mathsf{SIDH}}$. Let us summarise in Table 4.1 the differences in size between $\mathsf{SIDH}, \mathsf{MTP}_{\mathsf{SIDH}}$ and $\mathsf{MD}_{\mathsf{SIDH}}$.

Conclusive remarks. At the time of submission (16 December 2022), we are in the uneasy situation of concluding this thesis with the unpublishable Paper 3 and one at an unclear state. With the latter we refer to Paper 2, where we study on collisions in supersingular isogeny graphs. The manuscript was submitted in March 2022 to a journal, but we authors have not received any reviews yet despite having urged the editors multiple times. The second part of

[1]Values for SIDH parameters are taken from the Open Quantum Safe library https://openquantumsafe.org/liboqs/algorithms/kem/sike.html

[2]These numbers are missing from the original paper. I have computed them directly from the specific parameters and key-generation algorithm. My computations assume 3 bits per exponent when the maximum value is 6, 1 bit when the maximum value is 1 and a total of p bits for the two secret coefficients (one in $\mathbb{Z}/A\mathbb{Z}$ and the other in $\mathbb{Z}/B\mathbb{Z}$). I would like to remark that one of the two coefficients is not necessary to complete the key exchange and can be removed from the secret key, saving ≈ 425 B and ≈ 700 B.

		SIDH[1] $A = 2^e$ $B = 3^f$	MTP$_{\mathsf{SIDH}}$ $A = \prod_{i=1}^{\lambda} p_i$ $B = \prod_{i=1}^{\lambda} q_i$	MD$_{\mathsf{SIDH}}$ $A = \prod_{i=1}^{\lambda} p_i^{e_i}$ $B = \prod_{i=1}^{\lambda} q_i^{f_i}$
$\lambda = 128$	p	434 b	2308 b	6806 b
	sk	28 B	145 B	$\approx$ 878 B[2]
	pk	330 B	1734 B	5105 B
	pk$_c$	197 B	1013 B	2980 B
$\lambda = 192$	p	610 b	465 b	11191 b
	sk	39 B	233 B	$\approx$ 1473 B[2]
	pk	462 B	2796 B	8394 B
	pk$_c$	274 B	1631 B	6890 B

Table 4.1: Comparing sizes of base primes, secret keys and public keys in SIDH and those in the two masked variants from [Fou22] and [Mor22].

the paper is clearly obsolete now, since the SIDH-based identification protocol is now broken in light of the attacks appeared since August 2022. The first part, addressing general questions on the supersingular isogeny graph and performing analysis of independent interest, contains results that are still valid. We are waiting for the first round of reviews from the journal, but the fate of this paper is undecided at this stage. Despite appearing here as pre-prints, Paper 4 and Paper 5 are valid and we expect them to be accepted for publication in 2023.

Bibliography

[BMSS08] A. Bostan, F. Morain, B. Salvy, and É. Schost. Fast algorithms for computing isogenies between elliptic curves. *Mathematics of Computation*, 77(263):1755–1778, sep 2008.

[BR94] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer, Heidelberg, August 1994.

[CCG+19] Katriel Cohn-Gordon, Cas Cremers, Kristian Gjøsteen, Håkon Jacobsen, and Tibor Jager. Highly efficient key exchange protocols with optimal tightness. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 767–797. Springer, Heidelberg, August 2019.

[CD22] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Paper 2022/975, 2022. https://eprint.iacr.org/2022/975.

[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.

[CLG09] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, January 2009.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, Heidelberg, December 2018.

[Cor08] Giuseppe Cornacchia. Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. *Giornale di Matematiche di Battaglini*, page 46:33–90, 1908.

[Cos19] Craig Costello. *Supersingular Isogeny Key Exchange for Beginners*, page 21–50. Springer-Verlag, Berlin, Heidelberg, 2019.

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291.

[DdF+21] Luca De Feo, Cyprien de Saint Guilhem, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, and Benjamin Wesolowski. Séta: Supersingular encryption from torsion attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 249–278. Springer, Heidelberg, December 2021.

[DDGZ21] Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. Cryptology ePrint Archive, Report 2021/1023, 2021. https://eprint.iacr.org/2021/1023.

[Deu41] Max Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DKL+20a] 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, *Advances in Cryptology – ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, Heidelberg, December 2020.

[DKL+20b] 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, *Advances in Cryptology – ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, Heidelberg, December 2020.

[DKS18] Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 365–394. Springer, Heidelberg, December 2018.

[DLW22] Luca De Feo, Antonin Leroux, and Benjamin Wesolowski. New algorithms for the deuring correspondence: SQISign twice as fast. Cryptology ePrint Archive, Report 2022/234, 2022. https://eprint.iacr.org/2022/234.

[EHL+18] Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 329–368. Springer, Heidelberg, April / May 2018.

[FLW22] Luca De Feo, Antonin Leroux, and Benjamin Wesolowski. New algorithms for the Deuring correspondence: SQISign twice as fast. *IACR Cryptol. ePrint Arch.*, page 234, 2022.

[Fou22] Tako Boris Fouotsa. SIDH with masked torsion point images. Cryptology ePrint Archive, Paper 2022/1054, 2022. https://eprint.iacr.org/2022/1054.

[GPS20] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, 33(1):130–175, January 2020.

[GPV21] Wissam Ghantous, Federico Pintore, and Mattia Veroni. Collisions in supersingular isogeny graphs and the SIDH-based identification protocol. Cryptology ePrint Archive, Report 2021/1051, 2021. https://eprint.iacr.org/2021/1051.

[HBD+17] Andreas Hulsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kolbl, Tanja Lange, Martin M Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, and Peter Schwabe. SPHINCS+. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, Heidelberg, November 2017.

[HM89] James Lee Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of The American Mathematical Society*, 1989.

[JAC+17] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes,

Vladimir Soukharev, and David Urbanik. SIKE. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[JD11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011.

[Kan97] Ernst Kani. The existence of curves of genus two with elliptic differentials. *Journal of Number Theory*, 64(1):130–161, 1997.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. 2014. https://eprint.iacr.org/2014/505.

[Kup03] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. 2003.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979.

[LDK+17] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[Ler21] Antonin Leroux. A new isogeny representation and applications to cryptography. Cryptology ePrint Archive, Report 2021/1600, 2021. https://eprint.iacr.org/2021/1600.

[LS08] Reynald Lercier and Thomas Sirvent. On Elkies subgroups of ℓ-torsion points in elliptic curves defined over a finite field, 2008.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. The deep space network progress report 42-44, Jet Propulsion Laboratory, California Institute of Technology, January/February 1978. https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[Mes86] Jean-Francois Mestre. La méthode des graphes. Exemples et applications. In *Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata)*, pages 217–242. Citeseer, 1986.

[MM22] Luciano Maino and Chloe Martindale. An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Paper 2022/1026, 2022. https://eprint.iacr.org/2022/1026.

[Mor22] Tomoki Moriya. Masked-degree SIDH. Cryptology ePrint Archive, Paper 2022/1019, 2022. https://eprint.iacr.org/2022/1019.

[PFH+17] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[Piz90] Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bulletin (New Series) of the American Mathematical Society*, 23(1):127 – 137, 1990.

[PL17] Christophe Petit and Kristin Lauter. Hard and easy problems for supersingular isogeny graphs. Cryptology ePrint Archive, Report 2017/962, 2017. https://eprint.iacr.org/2017/962.

[Reg04] Oded Regev. A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space. *arXiv e-prints*, pages quant–ph/0406151, June 2004.

[Rob22] Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, Paper 2022/1038, 2022. https://eprint.iacr.org/2022/1038.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. https://eprint.iacr.org/2006/145.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978.

[SAB+17] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[Sch85] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44:483–494, 1985.

[Sch87] René Schoof. Nonsingular plane cubic curves over finite fields. *J. Comb. Theory Ser. A*, 46(2):183–211, nov 1987.

[Sch95] René Schoof. Counting points on elliptic curves over finite fields. *Journal de Theorie des Nombres de Bordeaux*, 7:219–254, 1995.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct 1997.

[Sil09] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106. 2009.

[Sut13] Andrew Sutherland. Isogeny volcanoes. *The Open Book Series*, 1(1):507–530, nov 2013.

[Tat66] J. Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathematicae*, 2:134–144, 1966.

[Vél71] J. Vélu. Isogénies entre courbes elliptiques. *Comptes-Rendus de l'Académie des Sciences, Série I*, 273:238–241, 1971.

[Voi21] John Voight. *Quaternion Algebras*. 01 2021.

[Was08] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman & Hall/CRC, 2 edition, 2008.

[Wes22] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1100–1111, 2022.

[YM21] Borissov Yuri and Markov Miroslav. An efficient approach to point-counting on elliptic curves from a prominent family over the prime field $\mathbb{F}_p$. *Mathematics*, 9(12), 2021.

Part II
Included papers

Paper 1

Practical Isogeny-Based Key-exchange with Optimal Tightness

Bor de Kock, Kristian Gjøsteen, Mattia Veroni.

Selected Areas in Cryptography, 2020.

Practical Isogeny-Based Key-exchange with Optimal Tightness

Bor de Kock, Kristian Gjøsteen, and Mattia Veroni

NTNU – Norwegian University of Science and Technology, Trondheim, Norway.
{bor.dekock,kristian.gjosteen,mattia.veroni}@ntnu.no

Abstract. We exploit the Diffie-Hellman-like structure of CSIDH to build a quantum-resistant authenticated key-exchange algorithm. Our security proof has optimal tightness, which means that the protocol is efficient even when instantiated with theoretically-sound security parameters. Compared to previous isogeny-based authenticated key-exchange protocols, our scheme is extremely simple, its security relies only on the underlying CSIDH-problem and it has optimal communication complexity for CSIDH-based protocols. Our security proof relies heavily on the re-randomizability of CSIDH-like problems and carries on in the ROM.

Keywords: Post-quantum, isogenies, key-exchange, provable-security, tightness, re-randomization.

1 Introduction

Authenticated key-exchange protocols allow two parties to collaborate in order to create a shared secret key, providing each of them with some assurance on the identity of the partner. Authentication can be achieved in two ways: *implicitly*, if the algebraic properties of the scheme imply that the only user who can compute the shared key is the intended one, or *explicitly*, by receiving a confirmation that the interlocutor has actually computed the key. The latter implies the use of a second mechanism which provides authentication, like a signature scheme, a KEM or a MAC. Even if explicit authentication might seem a stronger and preferable feature, in the real world it does not add much to the security of the protocol. First of all, it does not guarantee that the partner holds the shared key for all the time between the key confirmation and the use of the key. Moreover, the generation of signatures or the use of KEMs and MACs produces evidence of participation to a key-exchange, while implicit authentication does not. Finally, the schemes relying on implicit authentication typically require less computations and message exchanges compared to those involving an explicit authentication mechanism, with a significant profit in computational cost and communication efficiency.

The security proof limits the advantage of an adversary in breaking the scheme to the probability of solving some mathematical hard problem. Deploying a cryptographic algorithm should always be done in a *theoretically sound* way: the size of the concrete parameters must be large enough to guarantee the

required λ bits of security. If on one hand any security proof asymptotically guarantees the desired security level, on the other hand we want to use the smallest parameters possible, in order to obtain the most efficient implementation under the given security constraints. It is therefore extremely relevant to measure the so-called *tightness* of the proof by computing its security loss $L(\lambda)$, which should be as small as possible. The parameters on which we focus are, in particular, the number of users running the protocol and the number of sessions per user; both quantities are typically approximated to 2^{16}. Note that, nowadays, security proofs [JKSS12,KPW13,BFK+14] for a widely deployed protocol such as TLS have a quadratic loss in the number of sessions, fact that is not taken into account for the implementation.

In 2019 Cohn-Gordon et al. [CCG+19] developed a key-exchange protocol with an nearly (but optimally) tight security proof. In particular, the security loss is linear in the number of users and constant in the number of sessions per user. The schemes in the latter paper base their security on the Strong-DH assumption and its variants, defined over cyclic groups of prime order. The re-randomization of Diffie-Hellman problems plays a fundamental role in achieving the optimal tightness of the proofs, and thus it is a desirable feature that we cannot disregard. The tightness and practicality of these schemes raise an interesting question: is it possible to adapt the protocols (together with their security proofs) in order to make them quantum-safe?

In 1997, Peter Shor [Sho97] published a quantum algorithm for integer factorization and one for computing discrete logarithms, both running in polynomial time. As soon as a large-scale quantum computer will become available, the information security based on primitives like the RSA cryptosystem and the Diffie-Hellman key-exchange will be breached. In order to address this quantum threat, many researchers have focused their attention on post-quantum cryptography. The goal is to find new cryptographic primitives which can be implemented on classical computers, still guaranteeing security against both classical and quantum adversaries. In 2016, NIST announced a world-wide competition for new post-quantum standards in public-key encryption and digital signature algorithms. 69 submissions were accepted in the first round, 26 made it to the second step, and 7 finalists were announced on July 22, 2020. The search for new post-quantum cryptographic standards is still ongoing.

Supersingular-Isogeny based Diffie-Hellman (SIDH) [JD11] is one of the promising candidates in the search for post-quantum cryptographic protocols. Key-exchange protocols based on isogenies are unique in the sense that they provide key-sizes roughly similar to those of pre-quantum alternatives, but they are also known for being more complex (algebraically) compared to some of the post-quantum alternatives. An example of a scheme that is based on SIDH is SIKE [JAC+19], which is one of the 26 candidates in the second round of NIST's 2016 competition for post-quantum cryptographic protocols. Even if SIKE is not among the finalists announced in July 2020, NIST has shown high interest on isogeny-based cryptography, encouraging further research on this field [AASA+].

Although SIDH-based schemes have been around for a few years now, there are still open questions about the security behind them. In particular, random self-reducibility of SIDH problems seems very hard to achieve. A different isogeny-based scheme is CSIDH [CLM+18]: introduced in 2018, it offers a much more flexible and adaptable algebraic structure. In this paper we show how to obtain an optimally tight security proof for a CSIDH-based key-exchange protocol, making use of random self-reducibility. This kind of re-randomization plays a fundamental role in the tight proofs of, for instance, the classical Diffie-Hellman key-exchange, but is also used in modern schemes: Cohn-Gordon et al. [CCG+19] exploit this property to construct a tightly-secure AKE protocol.

The protocol we introduce is, to our knowledge, the best proven-secure result for isogeny-based key-exchange protocols. The proofs presented here draw on the proofs from Cohn-Gordon et al. [CCG+19], but with changes to the re-randomization strategy, since re-randomization in the isogeny case is different from the one in the cyclic group case. Both efficiency and tightness are a significant improvement over the state of the art, and can lead to the deployment of schemes with more efficient parameter choices obtaining high security at computational costs which are as low as possible.

1.1 Our contributions

In section 3.2 we adapt protocol Π by Cohn-Gordon et al. [CCG+19] to the isogeny setting, obtaining the first implicitly authenticated CSIDH-like protocol with weak forward secrecy, under only the Strong-CSIDH assumption. This is the first scheme with a security proof (moreover with optimal tightness) in the same setting as CSIDH. The protocol requires each user to perform 4 ideal-class evaluations, and its security proof, shown in Appendix B, has a tightness loss which is linear in the number of sessions performed by a single user.

The adaptation we perform is, however, not entirely straightforward. In the new setting we have only one operation, namely the multiplication of ideal classes, while in the original protocol re-randomization is achieved via two operations (addition and multiplication of exponents). This leads to a different re-randomization technique which relies one the random self-reducibility of the computational CSIDH problem shown in appendix 4.1.

We obtain a significant improvement over the state of the art of isogeny-based key-exchange protocols. Compared to one of the latest scheme, from "Strongly Secure Authenticated Key Exchange from Supersingular Isogenies" [XXW+19], we obtain better efficiency and tightness. Moreover, unlike this latter scheme, our protocol does not require any authentication mechanism. This allows us to rely on the same class (and a smaller number) of hardness assumptions, and to avoid the use of signatures, which are tricky and expensive [DG19] to produce in the isogeny setting. Compared to the CSIDH protocol, which lacks a security proof and for which authentication seems hard to achieve, our Π-SIDE protocol has implicit authentication at the cost of a few more ideal-class evaluations. As shown in section 6, our Π-SIDE protocol is competitive with other post-quantum candidates, once instantiated with theoretically-sound parameters.

1.2 Related work

In the last years, a lot of research has been conducted on SIDH-based schemes. For example, Galbraith [Gal18] has shown how to adapt generic constructions to the SIDH setting, and he introduced two new SIDH-AKE protocols. Similar results were achieved by Longa [Lon18], except for the introduction of the two new schemes. Assuming a straightforward adaptation, a few other protocols have a non-quadratic tightness loss. For example KEA+ [LM06] has a linear loss in the number of participants multiplied by the number of sessions, assuming the hardness of the Gap-DH problem. Although, it does not achieve wPFS and takes $O(t \log t)$ time only when instantiated on pairing-friendly curves.

In their recent paper, Xu et al. [XXW+19] propose SIAKE_2 and SIAKE_3, a two-pass and a three-pass AKE respectively. SIAKE_2, whose security relies on the decisional SIDH assumption, has a rather convoluted construction: they design a strong One-Way CPA secure PKE scheme, which is then turned into a One-Way CCA KEM through the modified FO-transform and finally used as a building block for the AKE scheme. The three-pass AKE SIAKE_3 is obtained by modifying the previously designed KEM, once a new assumption (the 1-Oracle SI-DH, an analogue of the Oracle Diffie-Hellman assumption in which only one query is allowed) is made. Compared to this scheme, our result is simpler and it has a tighter security proof, smaller communication complexity and improved overall efficiency.

2 Preliminaries

In this section, we first recall the definition of tightness for security reductions. Then we provide the reader with key-concepts and results which are indispensable to understand the constructions of SIDH and CSIDH. Good references regarding elliptic curves and isogenies are Silverman [HS09], Washington [Was08] and De Feo [Feo17]; the original papers introducing SIDH and CSIDH are Jao-De Feo [JD11] and Castryck et al. [CLM+18], respectively.

2.1 Tight reductions

When comparing schemes, one should always consider protocols once they have been instantiated with theoretically-sound parameters, which guarantee the desired level of security. These parameters (such as the bit-length of the prime defining a base field or the key size) strongly depend on the security proof correlated with the protocol. A security proof usually consists of

- a security model, in which we describe an adversary by listing a set of queries that it can make (and therefore specifying what it is allowed to do);
- a sequence of games leading to a *reduction*, in which an adversary $\mathcal{A}$ against the protocol is turned into a solver $\mathcal{B}$ for an allegedly hard problem.

The "quality" of a reduction can be measured by computing its security loss: if $t_{\mathcal{A}}$ and $\epsilon_{\mathcal{A}}$ are the running time and the success probability of $\mathcal{A}$ respectively, and $t_{\mathcal{B}}$ and $\epsilon_{\mathcal{B}}$ respectively are the running time and the success probability of $\mathcal{B}$, then we define the *security loss* L as

$$\frac{t_{\mathcal{A}}}{\epsilon_{\mathcal{A}}} = L\,\frac{t_{\mathcal{B}}}{\epsilon_{\mathcal{B}}}. \tag{1}$$

If L is constant, then we say that the reduction is *tight*. Having a tight proof is as relevant as building an efficient protocol, because this leads to deploy the smallest possible parameters when concretely instantiating a protocol.

In some cases, however, it is impossible to obtain a tight reduction. In a *simple scheme* the adversary is run only once, in comparison to other protocols which use the Forking Lemma in order to run multiple copies of the adversary. A linear loss in the number of participants to the protocol is unavoidable for simple schemes, while applying the Forking Lemma leads to a non-tight proof. We therefore focus on *optimal tightness* whenever tightness is unachievable: the L in Equation (1) turns out to be not constant, but one proves that it is impossible to decrease its order. We rely on the same strategies adopted in the paper by Cohn-Gordon et al. [CCG+19] to prove the lower bound on the tightness loss, applying their variant of the meta-reduction techniques by Bader et al. [BJLS16].

Many available schemes, which are actually taken into account for standardization processes, have quite non-tight security reductions. Let μ be the number of users running the protocol and let k be the number of sessions per user. HMQV [Kra05], a classically secure protocol in the random-oracle model under the CDH assumption, has security loss $O\left(\mu^2 k^2\right)$. If we consider a generic signed KEM approach, we get a $O\left(\mu^2 k^2\right)$ loss in addition to the signature scheme loss. In many cases, parameters are chosen in a non theoretically-sound way, while tightness loss should always be considered when comparing protocols.

2.2 Elliptic curves, isogenies and endomorphism rings

Let $\mathbb{F}_p$ be a finite field for a large prime p and let E be an elliptic curve over $\mathbb{F}_p$. We say that E is *supersingular* if and only if it has order $\#E(\mathbb{F}_p) = p + 1$. Consider the isomorphisms of elliptic curves, i.e. all the invertible algebraic maps. Any two elliptic curves over the algebraic closure $\overline{\mathbb{F}}_p$ are *isomorphic* if and only if they have the same j-invariant. Thus we can use isomorphisms to define an equivalence relation between elliptic curves and identify an equivalence class by the j-invariant of the curves in the class.

Let E_1 and E_2 be two elliptic curves defined over $\mathbb{F}_p$ and let $0_{E_1}, 0_{E_2}$ denote the respective points at infinity. An *isogeny* from E_1 to E_2 is a morphism $\phi : E_1 \to E_2$ such that $\phi(0_{E_1}) = 0_{E_2}$. For any isogeny $\phi : E_1 \to E_2$ there exists a *dual isogeny* $\hat{\phi} : E_2 \to E_1$ such that $\hat{\phi} \circ \phi = [deg(\phi)]_{E_1}$ and $\phi \circ \hat{\phi} = [deg(\phi)]_{E_2}$. An isogeny is essentially determined by its kernel: given a finite subgroup $G \subset E(\overline{\mathbb{F}}_p)$ there exist a unique (up to isomorphisms) elliptic curve $E_2 \simeq E_1/G$ and a separable isogeny $\phi : E_1 \to E_2$ such that $ker(\phi) = G$. The isogeny ϕ has *degree*

ℓ equal to the cardinality of its kernel, and we call it an ℓ-isogeny. Given the kernel of an isogeny, we can exploit Vélu's formulae [Vél71] to compute the isogeny ϕ together with the codomain curve E_2 in $O(\ell\ log(p)^2)$ bit operations. This is the best approach when ℓ is small enough and p is shorter than a few thousand bits. Any separable isogeny defined over $\mathbb{F}_p$ can be written as the composition of isogenies of prime degrees.

An *endomorphism* is an isogeny from E to itself; the set of endomorphisms of E, together with the zero map and equipped with pointwise addition and composition, forms the *endomorphism ring* $End(E)$. We denote by $End_p(E)$ the ring of endomorphisms defined over $\mathbb{F}_p$. For ordinary curves $End_p(E) = End(E)$, while for supersingular curves $End_p(E) \subset End(E)$. In particular, $End(E)$ is an order in a quaternion algebra, whilst $End_p(E)$ is an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{p})$. A classical result by Deuring [Deu41] reveals that $End(E)$ is a maximal order in $B_{p,\infty}$, the quaternion algebra ramified at p and at ∞.

2.3 The ideal class group action

We hereafter provide the reader with the basic definitions and known results regarding ideal class group action. In particular, this section gravitates around a recurring sentence in isogeny-based cryptography:

> "The ideal class group of an imaginary quadratic order $\mathcal{O}$ acts freely via isogenies on the set of elliptic curves with $End_p(E) \simeq \mathcal{O}$."

We will then focus on the computational aspects, essential to understand CSIDH.

Algebraic foundations. An *algebra* A is a vector space over a field $\mathbb{K}$ equipped with a bilinear operation. If the bilinear operation is associative, then we say that A is an associative algebra. Given a unitary ring R, a left *R-module* ${}_RM$ consists of an abelian group $(M, +)$ and a scalar multiplication $R \times_R M \longrightarrow_R M$ which satisfies left/right distributivity, associativity and neutrality of ring's unit. Let R be an integral domain (a commutative unitary ring without zero-divisors) and let $\mathbb{K}$ be its field of fractions; a left R-module ${}_RM$ is a *lattice* in the vector space V over $\mathbb{K}$ if ${}_RM$ is finitely generated, R-torsion free and an R-submodule of V. An *order* is a subring $\mathcal{O}$ of a ring A such that 1) A is a finite dimensional algebra over $\mathbb{Q}$, 2) $\mathcal{O}$ spans A over $\mathbb{Q}$ (i.e. $\mathbb{Q}\mathcal{O} = A$), 3) $\mathcal{O}$ is an integer lattice in A.

The ideal class group. Let $\mathbb{K}$ be a finite extension of $\mathbb{Q}$ of degree 2, which is called a *quadratic number field*, and let $\mathcal{O} \subseteq \mathbb{K}$ be an order. The *norm* of an $\mathcal{O}$-ideal $\mathfrak{a} \subseteq \mathcal{O}$ is defined as $N(\mathfrak{a}) = |\mathcal{O}/\mathfrak{a}|$, which is equal to $\gcd(\{N(\alpha) \mid \alpha \in \mathfrak{a}\})$. Norms are multiplicative: $N(\mathfrak{ab}) = N(\mathfrak{a})\,N(\mathfrak{b})$. A *fractional ideal* of $\mathcal{O}$ is an $\mathcal{O}$-submodule of $\mathbb{K}$ of the form $\alpha\mathfrak{a}$, where $\alpha \in \mathbb{K}^*$ and $\mathfrak{a}$ is an $\mathcal{O}$-ideal. Fractional ideals can be multiplied and conjugated in the obvious way, and the norm extends multiplicatively to fractional ideals. A fractional $\mathcal{O}$-ideal is *invertible* if there exists a fractional $\mathcal{O}$-ideal $\mathfrak{b}$ such that $\mathfrak{ab} = \mathcal{O}$. If such $\mathfrak{b}$ exists, we denote $\mathfrak{a}^{-1} = \mathfrak{b}$. All the principal fractional ideals $\alpha\mathcal{O}$ where $\alpha \in \mathbb{K}^*$ are invertible.

The *ideal class group* of $\mathcal{O}$, defined as $cl(\mathcal{O}) := I(\mathcal{O})/P(\mathcal{O})$, is the quotient of the set of invertible fractional ideals $I(\mathcal{O})$ by the set of principal invertible fractional ideals $P(\mathcal{O})$: For any $M \in \mathbb{Z} \setminus \{0\}$, every ideal class $[\mathfrak{a}]$ has an integral representative of norm coprime to M. There is a unique *maximal order* of $\mathbb{K}$ with respect to inclusion, which is called the *ring of integers* and is denoted by $\mathcal{O}_{\mathbb{K}}$. The *conductor* of $\mathcal{O}$ in $\mathcal{O}_{\mathbb{K}}$ is the index $f = [\mathcal{O}_{\mathbb{K}}/\mathcal{O}]$. Every $\mathcal{O}$-ideal of norm coprime to the conductor is invertible and factors uniquely into prime ideals.

The class group action. Let $\mathcal{E}\ell\ell_p(\mathcal{O})$ be the set of supersingular elliptic curves over $\mathbb{F}_p$ with $End_p(E)$ isomorphic to an order $\mathcal{O}$ in an imaginary quadratic field and let $E \in \mathcal{E}\ell\ell_p(\mathcal{O})$. Given an $\mathcal{O}$-ideal $\mathfrak{a}$, we define the *action* of $\mathfrak{a}$ on E as follows:

1. we consider all the endomorphisms α in $\mathfrak{a}$,
2. we compute the $\mathfrak{a}$-torsion subgroup $E[\mathfrak{a}] = \cap_{\alpha\in\mathfrak{a}} ker(\alpha) = \{P \in E(\overline{\mathbb{F}}_p) : \alpha P = 0_E \ \forall \alpha \in \mathfrak{a}\}$,
3. we compute the isogeny $\phi_{\mathfrak{a}} : E \to E_{\mathfrak{a}} \simeq E/E[\mathfrak{a}]$.

It is common practice to denote the action of $\mathfrak{a}$ on E by $\mathfrak{a} * E$.

A fundamental result in isogeny-based protocols is the *Deuring correspondence* between the set of maximal orders in $B_{p,\infty}$ and the set of elliptic curves: fixing a supersingular elliptic curve E_0, every ℓ-isogeny $\alpha : E_0 \to E$ corresponds to an ideal $\mathfrak{a}$ of norm ℓ, and vice-versa. Since $E_{\mathfrak{a}}$ is determined (up to isomorphism) by the ideal class of $\mathfrak{a}$, finding different representatives of an ideal class corresponds to finding different isogenies between two fixed curves.

We can rewrite any ideal $\mathfrak{a}$ of $\mathcal{O}$ as the product of $\mathcal{O}$-ideals $\mathfrak{a} = (\pi_p\mathcal{O})^r\mathfrak{a}_s$, where π_p is the p-th Frobenius endomorphism and $\mathfrak{a}_s \not\subseteq \pi_p\mathcal{O}$. This defines an elliptic curve $\mathfrak{a} * E$ and an isogeny $\phi_{\mathfrak{a}} : E \longrightarrow \mathfrak{a} * E$ of degree $N(\mathfrak{a})$ as follows:

- the separable part of $\phi_{\mathfrak{a}}$ has kernel $\cap_{\alpha\in\mathfrak{a}_s} ker(\alpha)$;
- the purely inseparable part consists of r iterations of Frobenius.

The isogeny $\phi_{\mathfrak{a}}$ and the codomain $\mathfrak{a} * E$ are both defined over $\mathbb{F}_p$ and are unique up to $\mathbb{F}_p$-isomorphism. Directly from this construction it is clear that multiplying ideals and composing isogenies are equivalent operations.

Let $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ be the set of elliptic curves defined over $\mathbb{F}_p$ whose endomorphism ring is isomorphic to $\mathcal{O}$ such that the Frobenius endomorphism π_p corresponds to π. As explained by Castryck et al. [CLM+18], we get the following fundamental result:

Theorem 1. *Let $\mathcal{O}$ be an order in an imaginary quadratic field and $\pi \in \mathcal{O}$ such that $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ is non-empty. Then the ideal class group $cl(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ via the map*

$$\begin{aligned} cl(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) &\longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) \\ ([\mathfrak{a}], E) &\longrightarrow [\mathfrak{a}] * E. \end{aligned}$$

From now on, we drop the class notation"$[\mathfrak{a}]$" in favor of a simpler "$\mathfrak{a}$" by considering any integral representative in the class.

The structure of the class group. The class group $cl(\mathcal{O})$ is a finite abelian group whose cardinality is asymptotically $\#cl(\mathcal{O}) \sim \sqrt{|\Delta|}$. As argued by CSIDH's authors [CLM+18], computing the exact structure of the class group requires a lot of computational effort. The best known algorithm (by Hafner and McCurley [HM89]) for computing the structure of the class group is subexponential in Δ, which is typically very large for CSIDH (about the size of p). Therefore, the authors opt for heuristics which allow to find a very good approximation.

We are interested in the primes for which there exist distinct prime ideals $\mathfrak{l}, \bar{\mathfrak{l}}$ of $\mathcal{O}$ such that $\ell\mathcal{O} = \mathfrak{l}\bar{\mathfrak{l}}$. If ℓ is such a prime, we say that it splits in $\mathcal{O}$; ℓ is called an *Elkies primes* in the point-counting setting. The ideal $\mathfrak{l}$ is generated as $(\ell, \pi - \lambda)$, where $\lambda \in \mathbb{Z}/\ell\mathbb{Z}$ is an eigenvalue of π_p on the ℓ-torsion, and its conjugate is $\bar{\mathfrak{l}} = (\ell, \pi - \pi/\lambda)$, where p/λ is any integral representative of that quotient modulo ℓ. The prime ℓ splits in $\mathcal{O}$ if and only if Δ is a non-zero square modulo ℓ. The CSIDH protocol is carefully designed such that a long list of primes (74 in the 512-bit implementation) are Elkies primes.

Computing the group action. According to the heuristics which are assumed in CSIDH, any element of the group can be represented as the product of small primes ideals. We can compute $\mathfrak{l} * E$, the action of a prime ideal $\mathfrak{l} = (\ell, \pi - \lambda)$ on E, in three different ways:

(a) by using the modular polynomials [Sut13]:
   1. find $\mathbb{F}_p$-rational roots of the modular polynomial $\Phi_l(X, j(E))$, which are the j-invariants of the two possible codomains;
   2. compute the kernel polynomials $\chi(x) \in \mathbb{F}_p[x]$ for the corresponding isogenies;
   3. determine which of the options is the correct one by checking if $\pi_p(x, y) = [\lambda](x, y)$ modulo $\chi(x)$ over the curve;

(b) by using the division polynomials [Was08, XI.3]:
   1. factor the ℓ-th division polynomial $\psi_l(E)$ over $\mathbb{F}_p$;
   2. match the irreducible factors with the right Frobenius eigenvalues;
   3. use Kohel's formulae to compute the codomain;

(c) by using Vélu's formulae:
   1. find a basis of the ℓ-torsion points and compute the eigenspaces of π_p;
   2. apply Vélu's formulae to a basis point of the correct eigenspace to compute the codomain.

In CSIDH, the authors opt for the last method, which is the fastest when the necessary extension fields (in which the basis points lie) are small.

When $\lambda = 1$ the curve has a rational point defined over the base field $\mathbb{F}_p$. If we also have that $p/\lambda = -1$, the other eigenspace of Frobenius endomorphism modulo ℓ is defined over $\mathbb{F}_{p^2}$, so both codomains can be easily computed using Vélu's formulae over the base field, switching from a curve to its quadratic twist if necessary. The parameters of the implementation are decided such that $p \equiv -1 \pmod{\ell}$ for many different primes ℓ: in this case, $\lambda = 1$ automatically implies $p/\lambda = -1$.

3 Isogeny-based key-exchange protocols

Isogeny-based cryptography is a class of allegedly quantum-resistant schemes resulting from NIST's competition. Two of the most peculiar features that distinguish them from the other candidates are the use of shorter keys and the deployment of more sophisticated algebraic structures. In this section, we first provide an overview of CSIDH (pronounced "seaside") [CLM+18], a key-exchange protocol which does not take part in NIST's competition but is extremely interesting and promising. Then we introduce our new protocol Π-SIDE (pronounced "pie-side"), a translation if the protocol Π [CCG+19] in the CSIDH setting.

3.1 CSIDH

What follows is an outline of the CSIDH protocol, whose underlying algebraic structures are briefly explained in section 2.3. We dwell in particular on the aspects which are relevant to our results.

Parameters. Fix a large prime $p = 4 \cdot \ell_1 \cdot \ell_2 \cdot \cdots \ell_n - 1$ where ℓ_i are small distinct odd primes. p is designed such that $p \equiv 3 \pmod 4$, in order to

- easily write down supersingular elliptic curves over $\mathbb{F}_p$;
- make use of the Montgomery form of elliptic curves in the implementation.

The starting curve for each execution of the protocol is the supersingular elliptic curve in Montgomery form $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$. In this case the characteristic equation of the Frobenius endomorphism is $\pi_p^2 = -p$, which implies that the $\mathbb{F}_p$-rational endomorphism ring $End_p(E_0)$ is an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$; in particular, $End_p(E_0) = \mathbb{Z}[\pi]$. The resulting ℓ_i-isogeny graph is a disjoint union of cycles. Moreover, since $\pi^2 - 1 \equiv 0 \pmod{\ell_i}$ for each $i = 1, \ldots, n$, the ideals $\ell_i\mathcal{O}$ split as $\ell_i\mathcal{O} = \mathfrak{l}_i\overline{\mathfrak{l}_i} = (\ell_i, \pi - 1)(\ell_i, \pi + 1)$ (so all the ℓ_i are Elkies primes). Furthermore, the kernel of $\phi_{\mathfrak{l}_i}$ is the subgroup generated by a point P of order ℓ_i which lies in the kernel of $\pi - 1$. Analogously, the kernel of $\phi_{\overline{\mathfrak{l}_i}}$ is generated by a point Q of order ℓ_i that is defined over $\mathbb{F}_{p^2}$ but not in $\mathbb{F}_p$ and such that $\pi(Q) = -Q$.

Sampling ideals and computing their action. Although we want to sample uniformly at random from the ideal class group $cl(\mathcal{O})$, it is preferable not to compute its exact structure because of the large size of the discriminant Δ. By heuristically assuming that

- the ideals $\mathfrak{l}_i$ do not have very small order,
- the ideals $\mathfrak{l}_i$ are evenly distributed in the class group,

two ideals $\mathfrak{l}_1^{e_1}\mathfrak{l}_2^{e_2} \cdots \mathfrak{l}_n^{e_n}$ for small e_i will rarely lie in the same class. The e_i are sampled from a short range $\{-m, \ldots m\}$ for some integer m such that $2m + 1 \geq \sqrt[n]{\#cl(\mathcal{O})}$. Since the prime ideals $\mathfrak{l}_i$ are fixed, we represent any ideal $\prod_i \mathfrak{l}_i^{e_i}$ (which will be the user's secret key) as a vector $(e_1, e_2, \ldots, e_n) \in [-m, m]^n$.

Since $\pi^2 \equiv -p \equiv 1 \pmod{\ell_i}$, the eigenvalues of all ℓ_i-torsion subgroups are $+1$ and -1. This allows us to efficiently compute the action of $\mathfrak{l}_i$ by using method 3. in section 2.3.

Representing and validating $\mathbb{F}_p$-isomorphism classes. SIDH misses a key-validation protocol, and countermeasures are expensive. We recall how the authors of CSIDH solve the problem for their protocol. First of all, they provide a result [CLM+18, Proposition 8]) which states that, for the chosen p and supersingular elliptic curve, the Montgomery coefficient uniquely represents the class of elliptic curves resulting from the evaluation of an ideal. Secondly, to prove that an elliptic curve is supersingular (and thus $\#E(\mathbb{F}_p) = p+1$), it is enough to find a point $Q \in E$ whose order is a divisor of $p+1$ greater than $4\sqrt{p}$ (by Hasse's theorem, we have only one multiple of that divisor in the interval $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$, which must be the group order by Lagrange's theorem). They therefore provide an algorithm which takes a point at random and computes its order. With high probability (increasing with ℓ_i), this will tell in only one step if the curve is supersingular or not. If x-only Montgomery arithmetic is used, a random point P is obtained by randomly picking $x \in \mathbb{F}_p$, and there is no need to differentiate points in $\mathbb{F}_p$ and in $\mathbb{F}_{p^2}$ (in the second case, the point will correspond to an $\mathbb{F}_p$-rational point in the quadratic twist, which is supersingular if and only if the original curve is supersingular).

The CSIDH protocol. We first describe how to perform the Setup and the key-generation, then we schematise the simple structure of key-exchange protocol.

Setup. In this phase we set up the global parameters of the key-exchange protocol. In particular, we fix:

- n distinct odd primes ℓ_i, corresponding to n isogeny-degrees;
- a large prime $p = 4 \cdot \ell_1 \cdot \ell_2 \cdots \ell_n - 1$;
- the supersingular elliptic curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$ with endomorphism ring $\mathcal{O} = \mathbb{Z}[\pi]$.

Key generation. The *private key* is an n-tuple $(e_1, \dots, e_n)$ of integers, randomly sampled from a range $\{-m, \dots, m\}$ such that $2m + 1 \geq \sqrt[n]{\#cl(\mathcal{O})}$, representing the ideal class $\mathfrak{a} = \mathfrak{l}_1^{e_1}\mathfrak{l}_2^{e_2} \dots \mathfrak{l}_n^{e_n} \in cl(\mathcal{O})$. The *public key* is the Montgomery coefficient $A \in \mathbb{F}_p$ of the elliptic curve $\mathfrak{a} * E_0 : y^2 = x^3 + Ax^2 + x$, obtained by applying the action of $\mathfrak{a}$ to the curve E_0.

Algorithm 2: CSIDH, the non-interactive key-exchange protocol.

Alice	**Bob**
$ssk_A : \mathfrak{a} \in cl(\mathcal{O})$	$ssk_B : \mathfrak{b} \in cl(\mathcal{O})$
$spk_A : E_A = \mathfrak{a} * E_0$	$spk_B : E_B = \mathfrak{b} * E_0$
retrieve E_B and check its supersingularity;	retrieve E_A and check its supersingularity;
$K_A = \mathfrak{a} * E_B$	$K_B = \mathfrak{b} * E_A$

$$K_A = \mathfrak{a}\mathfrak{b} * E_0 = K_B$$

3.2 Our protocol: Π-SIDE

Algorithm 3: Π-SIDE protocol.

Alice: $\mathcal{P}_A \in \mathbb{F}_p$		**Bob**: $\mathcal{P}_B \in \mathbb{F}_p$
$ssk_A : \mathfrak{a} \in cl(\mathcal{O})$		$ssk_B : \mathfrak{b} \in cl(\mathcal{O})$
$spk_A : E_A = \mathfrak{a} * E_0$		$spk_B : E_B = \mathfrak{b} * E_0$
retrieve E_B and check its supersingularity;		
$esk_A : \mathfrak{f} \xleftarrow{\$} cl(\mathcal{O})$		
$epk_A : E_F = \mathfrak{f} * E_0$		retrieve E_A and check its supersingularity;
	$\xrightarrow{E_F}$	$esk_B : \mathfrak{g} \xleftarrow{\$} cl(\mathcal{O})$
		$epk_B : E_G = \mathfrak{g} * E_0$
	$\xleftarrow{E_G}$	

$$\texttt{ctxt} = \mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G$$

$$K_B = H(\texttt{ctxt} \,\|\, \mathfrak{g} * E_A \,\|\, \mathfrak{b} * E_F \,\|\, \mathfrak{g} * E_F)$$

$$K_A = H(\texttt{ctxt} \,\|\, \mathfrak{a} * E_G \,\|\, \mathfrak{f} * E_B \,\|\, \mathfrak{f} * E_G)$$

Just like in CSIDH, we fix a large prime $p = 4 \cdot \ell_1 \cdot \ell_2 \cdots \ell_n - 1$ for odd and distinct primes ℓ_i. Then we consider the supersingular elliptic curve $E_0 : y^2 = x^3 + x$ defined over $\mathbb{F}_p$, with endomorphism ring $\mathcal{O} = \mathbb{Z}[\pi]$. We recall that a key-pair $(\mathfrak{a}, E_A)$ can be correctly (with heuristic assumptions) formed as follows:

1. for $i = 1, 2, \ldots, n$, sample the exponent $a_i \xleftarrow{\$} \{-m, \ldots m\}$, where m is the smallest integer such that $2m + 1 \geq \sqrt[n]{\#cl(\mathcal{O})}$;
2. construct the fractional ideal $\mathfrak{a} = \mathfrak{l}_1^{a_1} \cdot \mathfrak{l}_2^{a_2} \cdots \mathfrak{l}_n^{a_n}$. The ideal class $\mathfrak{a}$ will play the role of secret key;
3. evaluate the action of the ideal class $\mathfrak{a}$ on the elliptic curve E_0, obtaining the curve $E_A = \mathfrak{a} * E_0$; E_A is the Montgomery curve defined by the equation $y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ and E_A will be the public part of the key pair.

The implementation-oriented reader should always remember that each elliptic curve should be represented using its Montgomery coefficient. For the sake of notation we will refer to the curve instead.

Let $\mathcal{P}$ be the set of participants to the key-exchange protocol. We assume that each party in $\mathcal{P}$ holds a *static secret key ssk* and a *static public key spk*, the latter registered at a certificate authority $\mathcal{CA}$. The certificate authority, upon

registering a public key, does not require a proof of knowledge on the corresponding secret key. We do not demand that public keys differ from party to party, but we allow each party to register only one public key.

Suppose now that two parties Alice and Bob (uniquely identified as $\mathcal{P}_A$ and $\mathcal{P}_B$) in the set $\mathcal{P}$ want to establish a shared key. Here we have to distinguish between the initiator of the protocol (in our example Alice) and the responder. At the beginning of the session, upon retrieving Bob's public key, Alice samples an *ephemeral secret key* $esk_A = \mathfrak{f}$, computes the *ephemeral public key* $epk_A = E_F$ and sends the result to $\mathcal{P}_B$. Upon receiving E_F, Bob first checks that it is supersingular and that its Montgomery coefficient is not in $\{\pm 2\}$; if so, he in turn samples an ephemeral secret key $esk_B = \mathfrak{g}$, computes the ephemeral public key E_G and sends it to Alice. Alice herself verifies the validity of E_G. Each of them can now obtain the *session key* K: given access to an hash function H, they can locally compute

$$K = H(\mathcal{P}_A \parallel \mathcal{P}_B \parallel E_A \parallel E_B \parallel E_F \parallel E_G \parallel \mathfrak{ag} * E_0 \parallel \mathfrak{bf} * E_0 \parallel \mathfrak{fg} * E_0).$$

3.3 The SIDH case

A question naturally arises: if Π can be adapted to the CSIDH setting, why can't we do the same in the SIDH setting? On one hand, it is surely possible to translate the protocol itself, since SIDH has a Diffie-Hellman-like structure too. The adaptation would require a different parameter choice, allowing two extra sets of basis points, and the exchange of four extra image points (the images of the peer's basis points via the ephemeral isogeny) in order to allow the two parties to compute the common key.

On the other hand, in this case the security proof wouldn't hit the optimality bound in the tightness loss. As it will be clarified in the next section, a property that plays a fundamental role in this sense is the random self-reducibility of the computational problem. In the next section we provide a formal proof of this feature in the CSIDH case. At our knowledge, there exists no evidence that SIDH shares this property, and it is rather unlikely to find a way to prove it.

4 Random self-reducibility

According to a fundamental definition by Blum and Micali, later rephrased by Naor [NR97], a problem f is *random self-reducible* if solving it at any given instance x can be reduced in polynomial time to the solution of f at one or more random instances y_i. In order to achieve random self-reducibility, there are two conditions that have to be satisfied:

- the generation of the random instances $y_1, \dots y_n$ has to be performed non-adaptively;
- the instances $y_1, \dots y_n$ must be uniformly distributed.

Random self-reducible problems are extremely relevant for cryptographic purposes. First of all, they are used in *worst-case to average-case reductions*: a worst-case instance of the problem can be used to generate a set of random instances, so that solving f on the random instances allows us to solve f at the worst-case instance in polynomial time. In the early '80s, Goldwasser and Micali exploited random self-reducibility of mathematical problems to construct cryptographic algorithms for probabilistic encryption [GM82] and pseudorandom generation [BM82]. Even more, if the group G and its generator g are properly chosen, the random self-reducibility of the discrete logarithm problem guarantees passive security of the plain Diffie-Hellman key-exchange protocol.

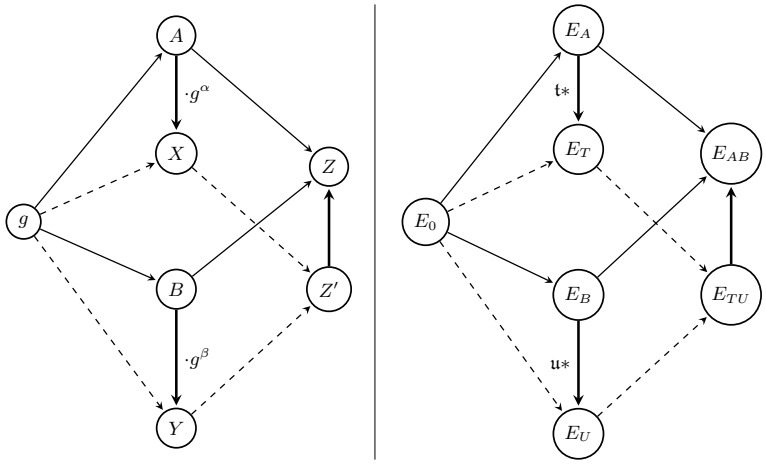

Fig. 1: Rerandomization graphs for Computational Diffie-Hellman and Computational-CSIDH problems.

4.1 Random self-reducibility on CSIDH

It is folklore that the key-recovery problem in CSIDH is random self-reducible, while SIDH-based problems are not. De Feo and Galbraith [DG19] provide a short proof of random self-reducibility of CSIDH; hereafter, we prove this property more verbosely, in a fashion that resembles the classical proof of rerandomizability for the Computational Diffie-Hellman problem. A fundamental role is played by the commutative action of $cl(\mathcal{O})$ on the set of elliptic curves with endomorphism ring isomorphic to $\mathcal{O}$. The presence of a commutative action is a very strong element of resemblance with the Diffie-Hellman protocol.

Let us start with the definition of the Computational CSIDH problem. Let $\mathbb{G}$ be the set of elliptic curves defined over $\mathbb{F}_p$.

Problem 1 (Computational-CSIDH problem)**.** *Given n distinct odd primes ℓ_i and a large prime $p = 4 \cdot \ell_1 \cdot \ell_2 \cdots \ell_n - 1$, let $E_0 \in \mathbb{G}$ be the supersingular elliptic curve in Montgomery form $y^2 = x^3 + x$. Given two valid CSIDH public keys $A, B \in \mathbb{F}_p$, where A is the Montgomery coefficient of the elliptic curve $E_A = \mathfrak{a} * E_0$ and B is the one of $E_B = \mathfrak{b} * E_0$, find the Montgomery coefficient $Z \in \mathbb{F}_p$ of the elliptic curve $E_{A,B} = \mathfrak{ab} * E_0$.*

Theorem 2. *The computational-CSIDH problem is random self-reducible. In other words, given any two random elliptic curves $E_T = \mathfrak{t} * E_0$ and $E_U = \mathfrak{u} * E_0$, for any algorithm $\mathcal{B}$ which solves the computational-CSIDH problem with advantage*

$$\mathrm{Adv}^{\mathrm{Comp-CSIDH}}_{\mathbb{G}}(\mathcal{B}) = \mathrm{Prob}\big[\, \mathcal{B}(E_T, E_U) = Z' \mid E_T \xleftarrow{\$} \mathbb{G}, E_U \xleftarrow{\$} \mathbb{G} \big]$$

there exists an oracle algorithm $\mathcal{A}^{\mathcal{B}}$ that, for any input $E_A, E_B \in \mathbb{G}$, outputs the correct solution to the corresponding computational-CSIDH problem with advantage $\mathrm{Adv}^{\mathrm{Comp-CSIDH}}_{\mathbb{G}}(\mathcal{B})$, and has roughly the same running time.

Proof. Let $E_A = \mathfrak{a} * E_0$ and $E_B = \mathfrak{b} * E_0$ be the two elliptic curves corresponding to the Montgomery coefficients A and B; we can construct the following algorithm:

$$\begin{aligned}
&\mathcal{A}^{\mathcal{B}}(E_A, E_B) \\
&\mathfrak{t}, \mathfrak{u} \xleftarrow{\$} cl(\mathcal{O}) \\
&E_T \leftarrow \mathfrak{t} * E_A = \mathfrak{t}' * E_0, \;\; E_U \leftarrow \mathfrak{u} * E_B = \mathfrak{u}' * E_0 \\
&Z' \leftarrow \mathcal{B}(E_T, E_U) \\
&\textbf{return } Z \text{ of } [\mathfrak{t}^{-1}\mathfrak{u}^{-1}] * E_{Z'}
\end{aligned}$$

In other words, the algorithm proceeds as follows. First of all, we pick uniformly at random two isogeny classes $\mathfrak{t}, \mathfrak{u} \in cl(\mathcal{O})$: they are defined as $\mathfrak{t} = \mathfrak{l}_1^{t_1}\mathfrak{l}_2^{t_2}\ldots\mathfrak{l}_n^{t_n} \in cl(\mathcal{O})$ and $\mathfrak{u} = \mathfrak{l}_1^{u_1}\mathfrak{l}_2^{u_2}\ldots\mathfrak{l}_n^{u_n} \in cl(\mathcal{O})$ where each exponent t_i, u_j is picked uniformly at random from the set $\{-m, \ldots, m\}$. Then we evaluate the action of $\mathfrak{t}$ on E_A and the action $\mathfrak{u}$ on E_B, obtaining two random elliptic curves $E_T, E_U \in \mathbb{G}$. Finally, we submit the new random instance to the algorithm $\mathcal{B}$, which outputs Z', the Montgomery coefficient of the elliptic curve $E_{Z'}$. Since

$$\begin{aligned}
E_{Z'} &= \mathfrak{t}'\mathfrak{u}' * E_0 \\
&= (\mathfrak{ta})(\mathfrak{ub}) * E_0 \\
&= (\mathfrak{tu})(\mathfrak{ab}) * E_0 \\
&= (\mathfrak{tu}) * E_{A,B},
\end{aligned}$$

we can easily retrieve the Montgomery coefficient Z of the elliptic curve $E_{A,B} = \mathfrak{t}^{-1}\mathfrak{u}^{-1} * E_{Z'}$. The advantage of the algorithm $\mathcal{A}^{\mathcal{B}}$ can be calculated as follows:

$$\mathrm{Prob}[\mathcal{A}^{\mathcal{B}}(E_A, E_B) = Z] = \mathrm{Prob}\left[\mathfrak{t}, \mathfrak{u} \xleftarrow{\$} cl(\mathcal{O}) : \mathcal{B}(\mathfrak{t} * E_A, \mathfrak{u} * E_B) = (\mathfrak{ta})(\mathfrak{ub}) * E_0\right].$$

By construction, the ideal classes $\mathfrak{t}$ and $\mathfrak{u}$ can be considered as sampled uniformly at random from $cl(\mathcal{O})$ (for the heuristics assumed in CSIDH), and therefore the elliptic curves $E_T = \mathfrak{t} * E_A$ and $E_U = \mathfrak{u} * E_B$ are independent and uniformly distributed on $\mathbb{G}$. Therefore, the oracle consulted by $\mathcal{A}^{\mathcal{B}}$ receives a well formed instance, so we can conclude that

$$\begin{aligned}\mathrm{Prob}[\mathcal{A}^{\mathcal{B}}(E_A, E_B) = Z] &= \mathrm{Prob}\left[\mathcal{B}(E_T, E_U) = \mathfrak{tau}\mathfrak{b} * E_0 \;\middle|\; \mathfrak{t}, \mathfrak{u} \xleftarrow{\$} cl(\mathcal{O})\right] \\ &= \mathrm{Adv}_{\mathbb{G}}^{\mathrm{Comp-CSIDH}}(\mathcal{B}).\end{aligned}$$

As pointed out in section 2.3, we can efficiently compute the action of the ideal classes $\mathfrak{l}$ and $\mathfrak{l}^{-1}$ by using Vélu-type formulae. Therefore we can conclude that, if $\mathcal{B}$ runs in t-time, then the algorithm $\mathcal{A}^{\mathcal{B}}$ runs in $(t + \delta)$-time, where δ is the small running time required to sample elements and evaluate the action of ideal classes.

□

5 Security of Π-SIDE

In this section, we define some allegedly hard problems in the CSIDH setting. The definition of our security model and the full proof can be found in Appendix B. The structure of the proof is similar to the one for protocol Π[CCG+19], but we have made a number of changes, mostly related to the new re-randomization technique. A straightforward adaption would have not been possible by simply substituting exponentiations with class group evaluations.

5.1 Hard problems

In section 4.1, we have seen that the Comp-CSIDH problem consists in finding the Montgomery coefficient $Z \in \mathbb{F}_p$ of the elliptic curve $\mathfrak{ab} * E_0$ given the Montgomery coefficients of the curves $E_A = \mathfrak{a} * E_0$ and $E_B = \mathfrak{b} * E_0$. In order to keep the notation as simple as possible, we will formulate the next problems referring to the elliptic curve itself, instead of its Montgomery coefficient. The reader should always keep in mind that, when it comes to the implementation, each elliptic curve will be represented by its Montgomery coefficient, which lies in $\mathbb{F}_p$. We start with defining a decisional problem:

Problem 2 (Decisional-CSIDH problem)**.** *In the CSIDH setting, let* $\mathfrak{a}, \mathfrak{b}, \mathfrak{r} \xleftarrow{\$} cl(\mathcal{O})$ *be three elements randomly sampled from* $cl(\mathcal{O})$ *and let* $b \xleftarrow{\$} \{0, 1\}$ *be the result of a fairly tossed coin. If* $b = 0$ *set* $E_Z = \mathfrak{r} * E_0$*, otherwise set* $E_Z = \mathfrak{ab} * E_0$ *and run the adversary on input* $(E_A = \mathfrak{a} * E_0, E_B = \mathfrak{b} * E_0, E_Z)$*. We define the advantage of* $\mathcal{A}$ *in solving the decisional CSIDH problem over* $cl(\mathcal{O})$ *as*

$$Adv_{cl(\mathcal{O})}^{Dec-CSIDH}(\mathcal{A}) := \left| Prob\big[\mathcal{A}(E_A, E_B, E_Z) = b\big] - \frac{1}{2} \right|.$$

In other words, the decisional problem is hard if the adversary succeeds with a negligible probability in distinguishing among a properly computed session key and a random key. Trivially, if we can solve the computational variant of problem then we can also solve its decisional variant. But does the opposite hold?

Problem 3 (Gap-CSIDH problem)**.** *In the CSIDH setting, let* $\mathfrak{a}, \mathfrak{b} \xleftarrow{\$} cl(\mathcal{O})$ *be two elements randomly sampled from* $cl(\mathcal{O})$*, corresponding to the curves* $E_A = \mathfrak{a} * E_0$ *and* $E_B = \mathfrak{b} * E_0$*. Suppose that the adversary* $\mathcal{A}$ *is given access to a Dec-CSIDH oracle* $\mathcal{D}(\cdot,\cdot,\cdot)$*, which outputs 1 if queried on a valid CSIDH triplet* (E_A, E_B, E_{AB}) *and 0 otherwise. We define the advantage of* $\mathcal{A}$ *in solving the Gap-CSIDH problem over* $cl(\mathcal{O})$ *as*

$$Adv^{Gap-CSIDH}_{cl(\mathcal{O})}(\mathcal{A}) := Prob\big[\mathcal{A}(E_A, E_B) = E_{A,B}\text{, providing } \mathcal{A} \text{ access to } \mathcal{D}(\cdot,\cdot,\cdot)\big]$$

The security of protocol Π [CCG+19] relies on the Strong-DH problem [ABR01], a variant of the Gap problem in which the adversary is granted access to a more limited decisional oracle.

Problem 4 (Strong-CSIDH problem)**.** *In the CSIDH setting, let* $\mathfrak{a}, \mathfrak{b} \xleftarrow{\$} cl(\mathcal{O})$ *be two elements randomly sampled from* $cl(\mathcal{O})$*, corresponding to the curves* $E_A = \mathfrak{a} * E_0$ *and* $E_B = \mathfrak{b} * E_0$*. Let* $\mathcal{D}$ *be an oracle for the decisional CSIDH problem. Suppose that the adversary* $\mathcal{A}$ *is given access to a decisional oracle with fixed first input* $\mathcal{D}_X(\cdot,\cdot) := \mathcal{D}(E_X, \cdot, \cdot)$*, which outputs 1 if queried on a valid CSIDH triplet* (E_X, E_Y, E_{XY}) *and 0 otherwise. We define the advantage of* $\mathcal{A}$ *in solving the Strong-CSIDH problem over* $cl(\mathcal{O})$ *as*

$$Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{A}) := Prob\big[\mathcal{A}(E_A, E_B) = E_{A,B}\text{, providing } \mathcal{A} \text{ access to } \mathcal{D}_X(\cdot,\cdot)\big]$$

Rerandomizability of the Gap-CSIDH and the Strong-CSIDH problems follows directly from Theorem 4.1. The full security proof, which strongly relies on these problems, is provided in Appendix B. Based on the current state of the art, there is no reason to believe that the above problems can be easily solved.

6 Comparison

Comparing the efficiency of our scheme with other post-quantum schemes is hard. First of all, many schemes do not have a security proof [Ber19] (and thus we cannot define theoretically-sound parameters); secondly, it is highly non-trivial to convert the concrete analysis into security parameters for many schemes.

Castryck et al. [CLM+18] describe an implementation for a 128-bit security level that requires about $106 \cdot 10^6$ clock cycles to compute the group action. Since our protocol Π-SIDE requires four group action computations, we have a total cost of about $400 \cdot 10^6$ clock cycles, ignoring hashing and other cheap operations.

The most natural target for comparison is SIKE [JAC+19]. The original Π-protocol can also be generalized to SIKE, but one would probably not attempt to build it on top of the defined KEM, but use the underlying isogeny instead.

Table 2.1 from SIKE [JAC+19] suggests that an isogeny computation using the optimized implementation (which probably matches the CSIDH implementation best) requires roughly $50 \cdot 10^6$ clock cycles for the 128 bit security level (`SIKEp434`), which becomes roughly $200 \cdot 10^6$ clock cycles for the generalized Π-protocol, significantly faster than the CSIDH-based version.

Now suppose we instantiate the protocol with 2^{16} users and 2^{16} sessions per user. In this case, the apparent security level of our protocol falls to about 110 bits. The SIKE-based protocol with the standard security proof will have a quadratic security loss. This means that in order to get a similar theoretically-sound security level from the SIKE-based protocol, we need to switch to `SIKEp610`. Again, Table 2.1 from SIKE [JAC+19] suggests that an isogeny computation using the optimized implementation requires roughly $160 \cdot 10^6$ clock cycles. The generalized Π-protocol then requires roughly $640 \cdot 10^6$ clock cycles, which is significantly slower than the CSIDH-based version. According to this approximate analysis, the CSIDH-based version is faster than the corresponding SIKE-based protocol when instantiated with theoretically-sound parameters. However, to properly determine which is faster, comparable optimized implementations would be needed.

Another natural comparison target is the Strongly secure AKE from Supersingular Isogenies by Xu et al. [XXW+19] referred to in section 1.2. For their two-pass protocol SIAKE_2 and their three-pass protocol SIAKE_3, the numbers of cycles are approximately $7 \cdot 10^9$ and $6 \cdot 10^9$, respectively [XXW+19, Table 6]. Our protocol is significantly faster, by about an order of magnitude.

7 Conclusions and open problems

In this paper we have shown that it is possible to construct post-quantum isogeny-based key-exchange protocols with optimal tightness, without compromising efficiency and key-size. The protocol is an easy adaptation of protocol Π [CCG+19], where we substitute exponentiations in cyclic groups with actions of ideal classes on elliptic curves. The adaptation of the proof, which requires random self-reducibility of the computational-CSIDH problem, could not be done trivially. Indeed, we have had to exploit a different re-randomization technique for the computational challenge, since we only have one group operation on ideal classes against two operations (addition and multiplication) on exponents. We have shown that the resulting scheme is competitive with other isogeny-based protocols, which lack a security proof or have a larger tightness loss.

Our protocol is proven secure in the Random Oracle Model. In a crucial step we use the Strong-CSIDH oracle to detect if the adversary queries the hashing oracle on an input which contains the solution to a given computational-CSIDH challenge. If we allow the adversary to make quantum queries, the target solution might be hidden in the superposition of states. We believe that collapsing the input state after the oracle's answer is not invalidating our security proof, since we do not need to reprogram the oracle. We leave the proof of security in the QROM as future work.

A stronger security notion can be achieved by adding the static-static term in the session-key computation, or by applying the NAXOS trick [LLM07]. But security against state compromise (ephemeral key reveal) increases the tightness loss, since we cannot tightly deal with state reveal queries. How to move to a stronger security model without losing in tightness is still an open problem.

We have seen how the flexible algebraic structure at the basis of CSIDH can be exploited to remodel protocol Π in the isogeny setting. Nevertheless, the simplicity of this scheme might be further exploited. Other quantum-hard problems might be used to translate the scheme in other algebraic contexts. Adaptions in this direction are left for further research.

As a last remark, we would like to clarify that our scheme is not affected by the algorithm recently published by Castryck et al. [CSV20]. This attack, which breaks some instances of the Decisional CSIDH problem, does not work when $p \equiv 3 \pmod 4$, as per our protocol.

References

AASA$^+$. Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Nistir 8309. `https://doi.org/10.6028/NIST.IR.8309`.

ABR01. Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle diffie-hellman assumptions and an analysis of dhies. volume 2020, pages 143–158, 04 2001.

Ber19. Daniel J. Bernstein. Comparing proofs of security for lattice-based encryption. *IACR Cryptology ePrint Archive*, 2019:691, 2019.

BFK$^+$14. Karthikeyan Bhargavan, Cédric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub, and Santiago Zanella Béguelin. Proving the TLS handshake secure (as it is). In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 235–255, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

BJLS16. Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 273–304, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

BM82. Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd FOCS*, pages 112–117, Chicago, Illinois, November 3–5, 1982. IEEE Computer Society Press.

CCG$^+$19. Katriel Cohn-Gordon, Cas Cremers, Kristian Gjøsteen, Håkon Jacobsen, and Tibor Jager. Highly efficient key exchange protocols with optimal tightness. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 767–797, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

CLM$^+$18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*,

volume 11274 of *LNCS*, pages 395–427, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany.

CSV20. Wouter Castryck, Jana Sotáková, and Frederik Vercauteren. Breaking the decisional diffie-hellman problem for class group actions using genus theory. Cryptology ePrint Archive, Report 2020/151, 2020. https://eprint.iacr.org/2020/151.

Deu41. Max Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941.

DG19. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

Feo17. Luca De Feo. Mathematics of isogeny based cryptography. *CoRR*, abs/1711.04062, 2017. http://arxiv.org/abs/1711.04062.

Gal18. Steven D. Galbraith. Authenticated key exchange for SIDH. Cryptology ePrint Archive, Report 2018/266, 2018. https://eprint.iacr.org/2018/266.

GM82. Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377, San Francisco, CA, USA, May 5–7, 1982. ACM Press.

HM89. James Lee Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of The American Mathematical Society*, 1989.

HS09. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106. 2009.

JAC+19. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, and Geovandro Pereira. SIKE. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

JD11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34, Tapei, Taiwan, November 29 – December 2 2011. Springer, Heidelberg, Germany.

JKSS12. Tibor Jager, Florian Kohlar, Sven Schäge, and Jörg Schwenk. On the security of TLS-DHE in the standard model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 273–293, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

KPW13. Hugo Krawczyk, Kenneth G. Paterson, and Hoeteck Wee. On the security of the TLS protocol: A systematic analysis. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 429–448, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

Kra05. Hugo Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 546–566, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Heidelberg, Germany.

LLM07. Brian A. LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *ProvSec 2007*, volume 4784 of *LNCS*, pages 1–16, Wollongong, Australia, November 1–2, 2007. Springer, Heidelberg, Germany.

LM06. Kristin Lauter and Anton Mityagin. Security analysis of KEA authenticated key exchange protocol. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006*, volume 3958 of *LNCS*, pages 378–394, New York, NY, USA, April 24–26, 2006. Springer, Heidelberg, Germany.

Lon18. Patrick Longa. A note on post-quantum authenticated key exchange from supersingular isogenies. Cryptology ePrint Archive, Report 2018/267, 2018. `https://eprint.iacr.org/2018/267`.

NR97. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.

Sho97. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct 1997.

Sut13. Andrew V. Sutherland. On the evaluation of modular polynomials. Cryptology ePrint Archive, Report 2013/181, 2013. `https://eprint.iacr.org/2013/181`.

Vél71. J. Vélu. Isogénies entre courbes elliptiques. *Comptes-Rendus de l'Académie des Sciences, Série I*, 273:238–241, 1971.

Was08. Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman & Hall/CRC, 2 edition, 2008.

XXW+19. Xiu Xu, Haiyang Xue, Kunpeng Wang, Man Ho Au, and Song Tian. Strongly secure authenticated key exchange from supersingular isogenies. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 278–308, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

Appendix

A Security model

Suppose that we have a certificate authority $\mathcal{CA}$, a set of parties $\mathcal{P} := \{\mathcal{P}_i\}_{i=1}^{\mu}$ and an adversary $\mathcal{M}$. The parties can communicate with each other and with $\mathcal{CA}$ by using an unauthenticated network. $\mathcal{CA}$ can be seen as a globally trusted party, or register, who holds and distributes the static public keys of the parties in $\mathcal{P}$. At any time, a new player can join in $\mathcal{P}$ by communicating his static public key to the $\mathcal{CA}$, and the register can grow indefinitely. As we mentioned before, we do not require different parties to hold different public keys, and neither we demand any proof of knowledge of the related secret key. Our protocol is implicitly authenticated and, as such, no identification or proof of knowledge of any secret information is required. The only constraint we impose is that each member can commit to only one static public key at a time.

Each $\mathcal{P}_i$ is represented as a set of oracles $\{\pi_i^1, \pi_i^2, \ldots, \pi_i^k\}$, one for each of the k sessions the user can participate to. Each oracle $\pi_i^s = (P_i^s, \psi_i^s, K_i^s, \mathsf{sent}_i^s, \mathsf{recv}_i^s, role_i^s)$ maintains an internal state consisting of:

- the identity of the intended peer P_i^s which is supposedly taking part to the key-exchange session;
- $\psi_i^s \in \{\emptyset, accept, reject\}$, which indicates whether the session key has not been computed yet, or if it has been accepted or rejected;
- the session key K_i^s, which is not empty if and only if $\psi_i^s = accept$;
- sent_i^s, the collection of all the messages sent by the oracle;
- recv_i^s, the collection of all the messages received by the oracle;
- the role $role_i^s$ of the oracle (`init` or `resp`).

sent_i^s and recv_i^s together form the view view_i^s of $\mathcal{P}_i$ of the session s.

We now define the attribute for indicating two oracles that allegedly participated to the same key-exchange session. Two oracles π_i^s and π_j^t are called *partner oracles* if

1. $P_i^s = P_j$ and $P_j^t = P_i$, i.e. if they are the intended peer of each other;
2. $\psi_i^s = \psi_j^t = accept$, i.e. they both accepted the session key;
3. $\mathsf{view}_i^s = \mathsf{view}_j^t$, i.e. the messages sent and received by P_i match with the ones respectively received and sent by P_j during the key-exchange session;
4. they have specular roles.

Slightly simplifying the definition, an oracle is *fresh* if and only if its session key has not been revealed, its partner oracle has not been corrupted or tested and the partner's session key has not been revealed. We will later constrain the adversary to test only fresh oracles. A party is *honest* if all its oracles are fresh, i.e. if it has not been corrupted yet.

In this model, the adversary $\mathcal{A}$ has full control over the network and interacts with the oracles through queries that allow it to

- *activate* an oracle π_i^s and assign a role by sending it a message on behalf of a peer P_j;
- *reveal the long-term secret key* of a user P_i. This query provides the target user with the attribute of *corrupted* and all its oracles will answer $\perp$ to each later query;
- *register the long-term public key* for a new user. No knowledge of the corresponding secret key is required and the public key is distributed to all other users;
- *reveal the session key* k_i^s stored in the internal state of any oracle π_i^s. The target oracle is now said to be *revealed*.
- *test an oracle* π_i^s, which outputs $\perp$ if $\psi_i^s \neq accept$. If $\psi_i^s = accept$ it then outputs a key, which is either the session-key or one picked at random, according to a previously defined random bit. The key, may it be real or the random, is consistently issued in case of further tests.

Note that the adversary cannot query on the ephemeral key of any session.

We work in the *Real-or-Random* model: when tested, each oracle will output a real session key or a random key, according to a bit sampled at the beginning of the security game. If $b = 0$ each oracle tested during the game will output a random key, while if $b = 1$ each tested oracle will output the real session key.

Once the environment has been set up, we run the following *AKE security game* $G_{\Pi}(\mu, k)$, with μ honest parties and at most k sessions per user:

1. at first we toss a coin $b \xleftarrow{\$} \{0, 1\}$. We also set up μ parties, providing each of them with a long-term key pair (sk_i, pk_i) and with k oracles;
2. we then run the adversary $\mathcal{A}$, which knows all the public keys and can make any number of the previously defined queries. The only restriction is that an oracle must be fresh when it is tested;
3. at some point, $\mathcal{A}$ will eventually output b', its guess on the initial bit b. If the tested oracles are fresh and $b' = b$, then $\mathcal{A}$ wins the security game.

An adversary can try to break the system in three different ways: it can trick two oracles into computing different session keys (event $\texttt{break}_{\texttt{Sound}}$), break the unicity of the partnership relation between two oracles (event $\texttt{break}_{\texttt{Unique}}$) or successfully guess $b' = b$ (event $\texttt{break}_{\texttt{KE}}$). We formalise these ideas in the following definition.

Definition 5. *In this security model, a protocol Π fails if at least one of* $\texttt{break}_{\texttt{Sound}}$, $\texttt{break}_{\texttt{Unique}}$ *and* $\texttt{break}_{\texttt{KE}}$ *occurs while running game $G(\mu, k)$. Given an adversary $\mathcal{A}$, we define its advantage against the AKE security of protocol Π as*

$$Adv_{\Pi}^{AKE}(\mathcal{A}) := \max\left\{Prob[\texttt{break}_{\texttt{Sound}}], Prob[\texttt{break}_{\texttt{Unique}}], Prob[\texttt{break}_{\texttt{KE}}] - \frac{1}{2}\right\}$$

and we say that it $(t, \epsilon_{\mathcal{A}}, \mu, k)$-breaks the AKE security of Π if it runs in time t and has advantage $Adv_{\Pi-SIDE}^{AKE}(\mathcal{A}) \geq \epsilon_{\mathcal{A}}$.

B The security proof

As in the proof by Cohn-Gordon et al. [CCG+19], we prove the following:

Theorem 3. *Consider an environment running Π-SIDE together with an adversary $\mathcal{A}$ against AKE security of Π-SIDE. Then there exist 3 Strong-CSIDH adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that $\mathcal{A}$'s advantage $Adv^{AKE}_{\Pi-SIDE}(\mathcal{A})$ is at most*

$$\mu \cdot Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_1) + Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_2) + \mu \cdot Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_3) + \frac{\mu k^2}{N}$$

where $\mu = |\mathcal{P}|$ is the number of parties, k is the maximal number of AKE-sessions per party and N is the order of $cl(\mathcal{O})$. The run-time of adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ is almost the same as $\mathcal{A}$ and they make at most as many queries to the Strong-CSIDH oracle as $\mathcal{A}$ does to the hash oracle H.

The proof structure is analogous to the one of Π, rephrased and adapted to our setting. It consists of six different games: Game 0 is the AKE experiment, while the other five games involve the following oracle types:

- type I: an initiator oracle which has received the response from a responder oracle (honest when the response is received) and with which it agrees on the transcript `ctxt`;
- type II: an initiator oracle whose intended peer is honest until the oracle accepts;
- type III: a responder oracle triggered by an honest initiator, with which it agrees on `ctxt` and which is still honest when it receives the response;
- type IV: a responder oracle whose intended peer is honest until the oracle accepts;
- type V: an oracle (whether initiator or responder) whose intended peer is corrupted.

Oracle	Init.	Resp.	Honest partner (before acceptance)	Honest partner (after acceptance)	Corrupted partner	Agreement on `ctxt`
Type I	✓		✓	✓		✓
Type II	✓		✓			
Type III		✓	✓	✓		✓
Type IV		✓	✓			
Type V	✓	✓			✓	

Table 1: Oracle types, defined by role, partner's honesty and agreement on `ctxt`.

At the time of starting an AKE session, an initiator oracle cannot be entirely sure about the intended peer's honesty: we cannot tell if it is of type I or type II. This uncertainty vanishes when it receives the response and it comes the

time to compute the session key. This aspect will be taken in account during the definition of the security games.

We now define six different security games, which will lead to the definition of the three adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ in Theorem 3. In each game we will have to look at the input to the hash function; for future references, we indicate the general form of the input to the hash oracle involving a key-exchange session between parties $\mathcal{P}_A, \mathcal{P}_B$ as

$$\mathcal{P}_A \parallel \mathcal{P}_B \parallel E_A \parallel E_B \parallel E_F \parallel E_G \parallel W_1 \parallel W_2 \parallel W_3 \tag{2}$$

For $i = 0, 1, \ldots, 5$ we denote with S_j the event "Game i outputs 1", which will indicate a success for the adversary in breaking protocol Π-SIDE (i.e. at least one of the events $\texttt{break}_{\texttt{Sound}}$, $\texttt{break}_{\texttt{Unique}}$ and $\texttt{break}_{\texttt{KE}}$ happens during Game i).

Game 0. In this game, we simply run the usual AKE security game: the adversary can corrupt some players, reveal some session keys (but not any ephemeral secret key) and delay/redirect messages. When it will be ready, it will pick a fresh oracle and make a query test on its session key. Game 0 will output 1 whenever the adversary breaks the AKE security of protocol Π-SIDE:

$$Prob[S_0] = Prob[\texttt{break}_{\texttt{KE}}].$$

Game 1. In this game we abort if the same `ctxt` is computed by two non-partnered oracles. We can upper-bound the probability of this event with the probability that the following conditions are simultaneously verified:

1. two oracles π_i^s, π_i^t belong to the same user P_i;
2. they pick the same ephemeral secret key during their respective sessions;
3. they are involved in two key-exchange sessions with the same user P_j (since the identity of the intended peer is part of the `ctxt`).

Recalling that we have μ users engaging in at most k sessions, we get that

$$|Prob[S_1] - Prob[S_0]| \leq \frac{\mu k^2}{N}$$

and thus, since in this game the unicity of the partner oracle cannot be broken, we can conclude that

$$Prob[\texttt{break}_{\text{Unique}}] \leq \frac{\mu k^2}{N}.$$

Game 2. In this game we modify how each oracle computes the session key: instead of computing the input to the hash oracle H, it checks if the adversary has queried the oracle on that same input, and behaves consequently: if the answer is yes, then it stores that hash value as the session key (i.e. it properly computes the key), otherwise it picks a key at random and stores that one instead. Note that, when it comes the time for an initiator oracle to compute the session key, the oracle type is fully determined.

A type I oracle (an initiator oracle with a definitely honest partner oracle with which it agrees on the `ctxt`) will store the key computed by the corresponding responder oracle.

Each type II or type V initiator oracles of party $\mathcal{P}_A$ has to check if the input

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{a} * E_G \,\|\, \mathfrak{f} * E_B \,\|\, \mathfrak{f} * E_G$$

has been object of any oracle query. If so, it sets its session key to the corresponding hash value (previously stored by the responder oracle), otherwise it picks a session key at random (answering consistently to any following hash query on that same input).

Each type III, IV or V responder oracle of a party $\mathcal{P}_B$ in a session with $\mathcal{P}_A$ will check if any queries have been made on input

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{g} * E_A \,\|\, \mathfrak{b} * E_F \,\|\, \mathfrak{g} * E_F.$$

If so, it stores the same result; otherwise, it stores a random key. In any case, each later hash query is consistently answered with the stored session key.

We cannot observe the exact time in which the key derivation oracle is queried for the first time, thus Game 2 outputs 1 whenever Game 1 outputs 1, and vice versa. We can therefore conclude that

$$Prob[S_2] = Prob[S_1].$$

Game 3. In this game (which is a variant of Game 2) we modify how a type IV oracle (a responder oracle whose intended peer is honest until the oracle accepts) chooses the session key. What it does is 1) to pick a random key; 2) to wait for the adversary to possibly corrupt the intended peer $\mathcal{P}_A$; 3) only then modify the hash oracle with the random key k.

We can now define the following events:

- L (for Long-term key), in which the adversary queries the hash oracle on input
$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{g} * E_A \,\|\, \mathfrak{b} * E_F \,\|\, \mathfrak{g} * E_F$$
before the long-term secret key of any initiator oracle is revealed;
- L_A is the same event as L, but for a specific intended peer $\mathcal{P}_A$. Trivially $Prob[L] = \sum_i Prob[L_i]$;
- C_A(for Corruption), in which the adversary queries the hash oracle on input
$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{g} * E_A \,\|\, W_2 \,\|\, W_3$$
before peer $\mathcal{P}_A$ is corrupted; therefore we have $Prob[L_A] \leq Prob[C_A]$.

In order to obtain a bound on $Prob[C_A]$ (and thus a bound on $Prob[L]$), we construct an adversary $\mathcal{B}_1$ against the Strong-CSIDH problem.

Definition 6. [*Adversary* $\mathcal{B}_1$] Consider now an adversary $\mathcal{B}_1$ which is given a Comp-CSIDH challenge (E_S, E_T) and is given access to a $\mathcal{D}_S(\cdot,\cdot)$ oracle. First of all, it chooses a user $\mathcal{P}_A$ uniformly at random and sets its long-term public key to $E_A = E_S$. Then it sets the ephemeral public key of a type IV oracle to be $\mathfrak{r} * E_T$, for a random $\mathfrak{r} \xleftarrow{\$} cl(\mathcal{O})$. Finally, it runs Game 2. If $\mathcal{B}_1$ corrupts $\mathcal{P}_A$, the experiment aborts.

We need to recognise the hash queries that involve the user $\mathcal{P}_A$ (happening in Game 2) and those involving the type IV oracle of any party $\mathcal{P}_B$. In particular,

1. consider hash queries of the form

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, W_1 \,\|\, \mathfrak{b} * E_F \,\|\, \mathfrak{f} * E_G$$

   involving user $\mathcal{P}_A$ as initiator. We do not know $\mathcal{P}_A$'s secret key $\mathfrak{a} = \mathfrak{s}$, so we have to recognise if W_1 is actually $E_{AG} = \mathfrak{s} * E_G$. This can be done by checking if $\mathcal{D}_S(E_G, W_1) = 1$;
2. consider hash queries of the form

$$\mathcal{P}_B \,\|\, \mathcal{P}_A \,\|\, E_B \,\|\, E_A \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{b} * E_G \,\|\, W_2 \,\|\, \mathfrak{f} * E_G$$

   involving user $\mathcal{P}_A$ as responder. Again, we do not know $\mathcal{P}_A$'s secret key $\mathfrak{a} = \mathfrak{s}$, but this time it is $W_2 = \mathfrak{a} * E_F$ that we cannot compute; thus we have to recognise if W_2 is actually $\mathfrak{s} * E_F$. This can be done by checking if $\mathcal{D}_S(E_F, W_2) = 1$;
3. consider hash queries of the form

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{g} * E_A \,\|\, W_2 \,\|\, W_3$$

   involving the type IV oracle and user P_A. We have to recognise if W_1 is actually $\mathfrak{rt} * E_A = \mathfrak{g} * E_S$. This can be done by checking if $\mathcal{D}_S(E_G, W_1) = 1$. Whenever we succeed and we find that $W_1 = E_{SG} = \mathfrak{s} * E_G$, since we computed $E_G = \mathfrak{r} * E_T$, we output

$$E_Z = \overline{\mathfrak{r}} * W_1 = \overline{\mathfrak{r}}\mathfrak{s} * E_G = \overline{\mathfrak{r}}\mathfrak{s}\mathfrak{r} * E_T = \overline{\mathfrak{r}}\mathfrak{r}\mathfrak{s} E_T = \mathfrak{s} * E_T = E_{ST}.$$

We have just described an adversary $\mathcal{B}_1$ which succeeds whenever event L_A occurs in Game 2. L_A can occur only before $\mathcal{P}_A$ is corrupt, and thus $\mathcal{B}_1$'s game would have gone through. We can therefore define the upper bound

$$Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_1) \geq \frac{1}{\mu}\sum_{i=1}^{\mu} Prob[C_I] \geq \frac{1}{\mu}\sum_{i=1}^{\mu} Prob[L_I] = \frac{1}{\mu} Prob[L]$$

from which we get that

$$\big|Prob[S_3] - Prob[S_2]\big| \leq Prob[L] \leq \mu \cdot Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_1)$$

the first element at the right-hand side of the inequality in Theorem 3.

Game 4. In this game a type III oracle (a responder oracle triggered by an honest initiator, with which it agrees on the $\mathtt{ctxt}$ and which is still honest when it receives the response) chooses the session key at random without modifying the key derivation hash oracle. Consider an oracle belonging to user $\mathcal{P}_B$ with static secret key $\mathfrak{b}$ and ephemeral secret key $\mathfrak{g}$ whose intended honest peer $\mathcal{P}_A$ has static secret key $\mathfrak{a}$. The adversary can find out this change only if (call this event L) it makes a query of the form

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, W_1 \,\|\, W_2 \,\|\, \mathfrak{g} * E_F.$$

This leads us to the following inequality:

$$\big|Prob[S_4] - Prob[S_3]\big| \leq Prob[L].$$

Similarly to what we did in the previous game, we want to bound $Prob[L]$ by constructing an adversary $\mathcal{B}_2$ against the Strong-CSIDH problem.

Definition 7. [*Adversary* $\mathcal{B}_2$] Consider now an adversary $\mathcal{B}_2$ which is given a Comp-CSIDH challenge (E_S, E_T) and is given access to a $\mathcal{D}_S(\cdot,\cdot)$ oracle. It runs Game 3., re-randomizing the challenge as follows: 1) it sets the ephemeral public key of type I and II oracles to $E_F = \mathfrak{r} * E_S$ for a random $\mathfrak{r} \xleftarrow{\$} cl(\mathcal{O})$; 2) it sets the ephemeral public key of type III oracles to $E_G = \mathfrak{r}' * E_T$ for a random $\mathfrak{r}' \xleftarrow{\$} cl(\mathcal{O})$.

In this game, since we embed the challenge in two ephemeral keys, all the static secret keys are known to the adversary. We need therefore to recognise two types of hash oracle queries:

1. hash queries for type II oracles of the form

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{a} * E_G \,\|\, \mathfrak{f} * E_B \,\|\, \mathfrak{f} * E_G$$

   given the knowledge of the static secret keys, the only information to be detected is whether $W_3 = \mathfrak{f} * E_G = \mathfrak{rs} * E_G$ or not. The answer can be obtained by performing the oracle query $\mathcal{D}_S(E_G, \overline{\mathfrak{r}} W_3)$;
2. hash queries for type III oracles of the form

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, W_1 \,\|\, W_2 \,\|\, \mathfrak{g} * E_F$$

   given the knowledge of the static secret keys, the only information to be detected is whether $W_3 = \mathfrak{g} * E_F = \mathfrak{r}'\mathfrak{t} * E_F$ or not. The answer can again be obtained by performing the oracle query $\mathcal{D}_S(E_G, [\overline{\mathfrak{r}} W_3)$.

If the Strong-CSIDH oracle outputs 1, then we output

$$E_Z = \mathfrak{r}^{-1}{\mathfrak{r}'}^{-1} W_3 = \overline{\mathfrak{r}\mathfrak{r}'}\mathfrak{f}\mathfrak{g} * E_0 = \overline{\mathfrak{r}\mathfrak{r}'}\mathfrak{r}\mathfrak{s}\mathfrak{r}'\mathfrak{t} * E_0 = \overline{\mathfrak{r}}\mathfrak{r}\overline{\mathfrak{r}'}\mathfrak{r}'\mathfrak{s}\mathfrak{t} * E_0 = E_{ST}.$$

We have just described an adversary $\mathcal{B}_2$ which succeeds whenever event L occurs in Game 2. From this fact we get that

$$\left|Prob[S_4] - Prob[S_3]\right| \leq Prob[L] \leq Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_2)$$

the second element at the right-hand side of the inequality in Theorem 3.

Game 5. In this game a type II oracle (an initiator oracle whose intended peer is honest until the oracle accepts) chooses a random key E_K and modifies the key derivation hash oracle only if the intended peer is corrupted. Consider an oracle belonging to user $\mathcal{P}_A$ with static secret key $\mathfrak{a}$ and ephemeral secret key $\mathfrak{f}$: if the adversary corrupts the intended peer $\mathcal{P}_B$, the hash oracle will output $E : k$ whenever it is queried on input

$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{a} * E_G \,\|\, \mathfrak{f} * E_B \,\|\, \mathfrak{f} * E_G.$$

Analogously to what we did in Game 3, we define the following events:

- L: a query on the above input happens before the long-term secret key of any responder oracle is revealed. It follows that
$$\left|Prob[S_5] - Prob[S_4]\right| \leq Prob[L];$$
- L_B: same as L, but for a specific intended peer $\mathcal{P}_B$. Trivially, $Prob[L] = \sum_i Prob[L_i]$;
- C_B: a query on input
$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, W_1 \,\|\, W_2 \,\|\, W_3 \qquad W_2 = \mathfrak{f} * E_B = \mathfrak{b} * E_F$$
happens before user $\mathcal{P}_B$ is corrupted; therefore we have $Prob[L_B] \leq Prob[C_B]$.

As we did in the previous games, we want to find an upper bound on $Prob[L]$.

Definition 8. [*Adversary* $\mathcal{B}_3$] Consider now an adversary $\mathcal{B}_3$ which is given a Comp-CSIDH challenge (E_S, E_T) and is given access to a $\mathcal{D}_S(\cdot,\cdot)$ oracle. It runs Game 4., it embeds the challenge as follows: 1) it sets the static public key of a uniformly-at-random user $\mathcal{P}_B$ to $E_B = E_S$; 2) it sets the ephemeral public key of type I and II oracles whose intended peer is $\mathcal{P}_B$ to $E_F = \mathfrak{r} * E_T$ for a random $\mathfrak{r} \xleftarrow{\$} cl(\mathcal{O})$.

If the adversary corrupts party $\mathcal{P}_B$, the game aborts, since the corresponding static secret key is unknown. We need therefore to recognise three types of queries made to the hash oracle:

1. hash queries for which $\mathcal{P}_B$ acts as responder
$$\mathcal{P}_A \,\|\, \mathcal{P}_B \,\|\, E_A \,\|\, E_B \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{g} * E_A \,\|\, \mathfrak{b} * E_F \,\|\, \mathfrak{g} * E_F.$$
Given that both $\mathfrak{b} = \mathfrak{s}$ and $\mathfrak{t}$ are unknown, the only information we cannot compute and that has to be detected is whether $W_2 = \mathfrak{b} * E_F = \mathfrak{b} * E_S$. The answer can be obtained by performing the oracle query $\mathcal{D}_S(E_F, W_2)$;

2. hash queries for which $\mathcal{P}_B$ acts as initiator:

$$\mathcal{P}_B \,\|\, \mathcal{P}_A \,\|\, E_B \,\|\, E_A \,\|\, E_F \,\|\, E_G \,\|\, \mathfrak{b} * E_G \,\|\, \mathfrak{f} * E_A \,\|\, \mathfrak{f} * E_G$$

(note that, in this case, the second part of the challenge has not been embedded in E_F). The only information to be detected is whether $W_1 = \mathfrak{b} * E_F = \mathfrak{b} * E_S$, and the answer can be obtained by performing the oracle query $\mathcal{D}_S(E_G, W_1)$;

3. hash queries defining event C_B, i.e. made before the user $\mathcal{P}_B$ is corrupted:

$$\mathcal{P}_A \| \mathcal{P}_B \,\| E_A \| E_B \| E_F \| E_G \| W_1 \| W_2 \| W_3 \qquad W_2 = \mathfrak{f} * E_B = \mathfrak{b} * E_F$$

We have to recognise if W_2 is actually $\mathfrak{f} * E_B = \mathfrak{rt} * E_B$, and this can be done by checking if $\mathcal{D}_S(E_F, W_2) = 1$.

If the Strong-CSIDH oracle outputs 1 and realise that $W_2 = \mathfrak{s} * E_F = \mathfrak{srt} * E_0$, then we output

$$E_Z = \mathfrak{r}^{-1} W_2 = \bar{\mathfrak{r}}\mathfrak{srt} * E_0 = \bar{\mathfrak{r}}\mathfrak{r}\mathfrak{s}\mathfrak{t} * E_0 = E_{ST}.$$

We have just described an adversary $\mathcal{B}_3$ which succeeds whenever event L_B occurs in Game 5. L_B can occur only before $\mathcal{P}_B$ is corrupt, and thus $\mathcal{B}_3$'s game would have gone through. We can therefore upper bound

$$Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_3) \geq \frac{1}{\mu} \sum_{i=1}^{\mu} Prob[C_I] \geq \frac{1}{\mu} \sum_{i=1}^{\mu} Prob[L_I] = \frac{1}{\mu} Prob[L]$$

from which we get that

$$\big|Prob[S_5] - Prob[S_4]\big| \leq Prob[L] \leq \mu \cdot Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_3)$$

the third and last element at the right-hand side of the inequality in Theorem 3.

Concluding the proof. Following from how we constructed each game in the proof, whenever the games do not abort because of adversarial corruption, the adversary is provided with a random session key, completely independent of every key and sent message. Therefore

$$Pr[S_5] = \frac{1}{2}.$$

We have seen in Game 1. that

$$Prob[\mathtt{break}_{\mathtt{Unique}}] \leq \frac{\mu k^2}{N}$$

and, due to the perfect correctness of the scheme,

$$Prob[\mathtt{break}_{\mathtt{Sound}}] = 0.$$

We can therefore exploit the bounds on adversarial winning probabilities to prove Theorem 3: given an adversary $\mathcal{A}$ against protocol Π-SIDE, we have built three adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ against Strong-CSIDH such that $\mathcal{A}$ wins with advantage $Adv^{AKE}_{\Pi-SIDE}(\mathcal{A})$ at most

$$\mu \cdot Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_1) + Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_2) + \mu \cdot Adv^{St-CSIDH}_{cl(\mathcal{O})}(\mathcal{B}_3) + \frac{\mu k^2}{N}$$

where μ is the number of participants to the protocol.

The tightness loss $L = \mathcal{O}(\mu)$ that we achieve in this security proof is optimal for simple protocols such as ours. The arguments adopted by Cohn-Gordon et al. [CCG+19] still hold in our setting and the adaptation is straightforward.

Paper 2

Collisions in Supersingular Isogeny Graphs and the SIDH-based Identification Protocol.

Wissam Ghantous, Shuichi Katsumata, Federico Pintore, Mattia Veroni.

Submitted to Journal of Algebra in March 2022.

Paper 3

Sigh: faster and shorter SIDH signatures.

Wissam Ghantous, Federico Pintore, Mattia Veroni.

Cannot be published as is.

Paper 4

Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic.

Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, Mattia Veroni.

Submitted.

Paper 5

Efficiency of SIDH-based signatures (yes, SIDH).

Wissam Ghantous, Federico Pintore, Mattia Veroni.

Pre-print version available on ePrint.

ISBN 978-82-326-6469-6 (printed ver.)
ISBN 978-82-326-6453-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology