Random Walks in Number-theoretic Cryptology

Benjamin Wesolowski

École Normale Supérieure de Lyon

Habilitation à diriger des recherches

Discipline : Section CNU 25, Mathématiques

RANDOM WALKS IN NUMBER-THEORETIC CRYPTOLOGY

Benjamin Wesolowski

Rapporteurs :

Steven D. Galbraith	Professeur, The University of Auckland, Nouvelle-Zélande
Pierrick Gaudry	Directeur de recherche, CNRS, LORIA, Nancy, France
Hendrik W. Lenstra	Professeur émérite, Universiteit Leiden, Pays-Bas

Soutenance publique le 29 août 2024, devant le jury composé de :

Jean-Marc Couveignes	Professeur, Institut de Mathématiques de Bordeaux, France
Pierrick Gaudry	Directeur de recherche, CNRS, LORIA, Nancy, France
Hendrik W. Lenstra	Professeur émérite, Universiteit Leiden, Pays-Bas
Sophie Morel	Directrice de recherche, CNRS, ENS de Lyon, France
Adeline Roux-Langlois	Directrice de recherche, CNRS, GREYC, Caen, France
René Schoof	Professeur, Università di Roma “Tor Vergata”, Italie

ACKNOWLEDGEMENTS

I am deeply grateful to Steven D. Galbraith, Pierrick Gaudry and Hendrik W. Lenstra who accepted to review this manuscript, and to Jean-Marc Couveignes, Sophie Morel, Adeline Roux-Langlois and René Schoof who, with Pierrick and Hendrik, have accepted to form the jury of my *habilitation defense*.

This manuscript is the product of a decade-long journey in the world of cryptology and number theory. That journey certainly owes the most to Arjen K. Lenstra, who hooked me to this field of research with his course on *Algorithms for Public Key Cryptography*, then welcomed me to his *Laboratory for Cryptologic Algorithms* to pursue a PhD under his guidance in Lausanne. My gratitude extends to my second PhD advisor, Robert Granger, who introduced me to one of the most challenging and stimulating problems I ever worked on, and to Thorsten Kleinjung, who chewed over that problem with me for four years.

The present manuscript is about the *après*-PhD, which first led me to Amsterdam. I am deeply grateful to Ronald Cramer for his warm welcome in his team at the *Centrum Wiskunde & Informatica*. Having already met a few of its members, I knew what to expect, and I was not disappointed. I wish to thank the entire team for the wonderful year I spent there. I was lucky to work closely with Léo Ducas, who had a significant impact on my scientific journey, became a friend, and trusted me to co-advise the PhD thesis of Koen de Boer. I owe no less to Koen, who made this first advising experience a pleasure, and with whom I have kept collaborating since his resoundingly successful graduation.

I spent the following three years in Bordeaux. I wish to thank Andreas Enge for welcoming me to the LFANT team and all its members for making it such a stimulating and friendly place. I am particularly grateful to my collaborators Aurel Page, Alice Pellet–Mary and Damien Robert, from whom I learned so much. I had already collaborated with Alice, and her arrival in Bordeaux led to the development of a precious friendship.

My last move brought me back to my birth-town, Lyon. I warmly thank the number theory team of the *École Normale Supérieure de Lyon* for welcoming me. The team, the entire laboratory and the support staff offer a wonderful, rich and friendly work environment. I am also grateful to the cryptography people in Lyon, in particular Guillaume Hanrot and Damien Stehlé for their sustained support and advice.

I wish to thank all the interns and students I have had the chance to advise, in particular my two current PhD students, Pierrick Dartois and Arthur Herlédan Le Merdy.

This journey was filled with too many invaluable encounters for these meagre paragraphs to do them all justice: all my coauthors, colleagues, advisors, advisees, support staff, host institutions, the CNRS, the IACR... Thank you all! To wrap it up, a special mention goes to Cécile Pierrot, one of the first friends I made in the community.

Et finalement, un grand merci à mes amis, à ma famille, et à Benjamin (pas moi).

ABSTRACT

This *habilitation thesis* presents a selection of the author's contributions at the intersection of computational number theory and cryptography. Its main focus is on the mathematical foundations of isogeny-based and lattice-based cryptography.

CONTENTS

INTRODUCTION

Cryptography met number theory in 1976, when Diffie and Hellman [DH76] achieved what had long been considered impossible: a protocol for two people to exchange secret information on a public channel, even if they had never met before to establish some kind of password, a pre-shared key. Diffie and Hellman designed the protocol such that a spy attempting to find the secret would need to solve a presumably hard computational problem: the *discrete logarithm* problem in the multiplicative group of a finite field.

This protocol is the first *public key cryptosystem*. In public key cryptography, each party has a pair of keys: a public key and a private key. The security of a cryptosystem is formalised by computational problems such as: given a public key, can one recover a paired private key? *Proving the security* consists in proving that these problems are hard, or at least as hard as some other well-studied problems. Cryptography requires hard and versatile computational problems. Number theory provides such problems, together with a powerful toolset for their analysis. The vast majority of deployed cryptosystems rely on the presumed hardness of the *discrete logarithm* problem, and the *integer factorisation* problem. Decades of cryptanalysis forged our confidence in these classical foundations.

In 1994, Shor [Sho97] discovered a quantum algorithm of polynomial complexity solving both of these problems, threatening all deployed public-key cryptography. Research on quantum technology is accelerating, and the threat is seriously considered by the cryptographic community, which has strived to develop schemes that resist quantum algorithms: *post-quantum cryptography*. Emerging post-quantum candidates build their foundations on a handful of computational problems rooted in arithmetic and geometry, and this manuscript explores two of them: isogeny-based cryptography, and lattice-based cryptography.

What to expect from this manuscript. This *habilitation thesis* is written as part of my application for the *Habilitation à Diriger des Recherches*. As such, its primary purpose is the exposition of a representative selection of the research I have conducted since the obtention of my PhD diploma in 2018. It is not an exhaustive account of my work, but an organized selection of interconnected topics: random walks and elliptic curves, elliptic curves and ideals, ideals and random walks.

Selected results are not presented in exhaustive mathematical detail. Instead, the manuscript focuses on context and motivation, explains the main theorems and ideas, and only hints at the proofs and techniques deployed. When relevant, results beyond my own contributions are discussed to leave the reader with a good overview of the current state of the field — this document is thereby a biased survey on selected topics. For clarity, references of which I am a coauthor are indicated with double brackets, as [[KW22]], while other references have simple brackets, as [Piz90].

The three chapters of this thesis concern three different objects arising in post-quantum cryptography: supersingular isogeny graphs, oriented elliptic curves, and ideal lattices.

Chapter 1: Supersingular isogenies and endomorphisms. Elliptic curves have long played a role in cryptography, as a platform for the discrete logarithm problem. They have kept a central place in post-quantum cryptography thanks to the emergence of new computational problems resisting quantum algorithms: isogeny problems. An *isogeny* is a map connecting two elliptic curves. An *isogeny problem* is a computational problem of this form: given two elliptic curves, find an isogeny between them. Most of isogeny-based cryptography is based on the presumed hardness of the "supersingular" isogeny problem and on a network of variants. In Chapter 1, we study these problems and their applications, and the central role of random walks in so-called *supersingular isogeny graphs*. It is organised around the following contributions.

- In the article ⟦Wes21⟧, we solidify the foundations of the field by proving that two of its most emblematic problems are in fact equivalent (assuming the Generalized Riemann Hypothesis): the *supersingular isogeny path* problem, and the *endomorphism ring* problem. The first is the path-finding problem in the so-called supersingular isogeny graph. The second is the problem of finding all endomorphisms (i.e., isogenies to itself) of a supersingular elliptic curve.
- In the article ⟦PW24⟧, we prove that the endomorphism ring problem is equivalent to the problem of finding one single (non-trivial) endomorphism. This so-called *one endomorphism* problem supports the security of several cryptosystems. This new connection immediately extends the web of equivalent problems supporting isogeny-based cryptography, leads to better algorithms for their resolution, and unlocks new security proofs. Our main tool is a new equidistribution theorem for random walks in a versatile generalization of isogeny graphs.
- One important problem resisted unification with the others: the computational problem supporting the security of the SIDH cryptosystem [JD11]. In a spectacular turn of events, this variant proved easy and SIDH was broken in 2022 in the series of articles [CD23], ⟦MMP+23⟧ and [Rob23]. We recount this downfall, carefully delineating our own contribution through ⟦Wes22b, MMP+23⟧.
- In ⟦DKL+20⟧, we introduce SQIsign, a digital signature scheme whose development is intertwined with all previously mentioned results. The latest version, called SQIsignHD ⟦DLRW24⟧, makes *constructive* use of the new tools which broke SIDH, and its security lies firmly on the hardest problems of the field. The development of SQIsign and its "HD" aspects is the topic of the thesis of my PhD student Pierrick Dartois, coauthor of SQIsignHD ⟦DLRW24⟧, SQIsign2D ⟦BDD+24⟧, and algorithms for evaluating isogenies in higher dimension [DMPR23].

The following is a complete list of my articles related to supersingular isogeny graphs and their applications in cryptography.

⟦BDD+24⟧ Andrea Basso, Luca De Feo, Pierrick Dartois, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D-West: The Fast, the Small, and the Safer. Cryptology ePrint Archive, Paper 2024/760, 2024. https://eprint.iacr.org/2024/760.

⟦PW24⟧ Aurel Page and Benjamin Wesolowski. The supersingular endomorphism ring and one endomorphism problems are equivalent. *To appear in Advances in Cryptology – EUROCRYPT 2024*, 2024.

[DLRW24] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: new dimensions in cryptography. *To appear in Advances in Cryptology – EUROCRYPT 2024*, 2024.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023.

[BCC+23] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14005 of *Lecture Notes in Computer Science*, pages 405–437. Springer, 2023.

[DLLW23] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the Deuring correspondence - towards practical and secure SQISign signatures. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 659–690. Springer, 2023.

[Wes22b] Benjamin Wesolowski. Understanding and improving the Castryck–Decru attack on SIDH. Archive ouverte HAL, Report hal-04557845, 2022. `https://hal.science/hal-04557845`.

[Wes21] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *62nd IEEE Annual Symposium on Foundations of Computer Science – FOCS 2021*, pages 1100–1111. IEEE, 2021.

[DKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.

[DDF+21] Luca De Feo, Cyprien Delpech 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*, volume 13093 of *Lecture Notes in Computer Science*, pages 249–278. Springer, 2021.

[GW17] Alexandre Gélin and Benjamin Wesolowski. Loop-abort faults on supersingular isogeny cryptosystems. In Tanja Lange and Tsuyoshi Takagi, editors, *International Workshop on Post-Quantum Cryptography – PQCrypto 2017*, pages 93–106. Springer, 2017.

Chapter 2: Oriented elliptic curves. The discrete logarithm problem is a special case of the *group action inversion* problem. While Shor's algorithm solves the former, the general case still resists quantum algorithms. The Diffie–Hellman protocol (and much of the vast array of "discrete logarithm"-based cryptosystems which followed) can be

brought to the post-quantum world by replacing discrete logarithms with an appropriate "hard to invert" group action. The classical theory of *complex multiplication* induces an action of an ideal class group on a collection of elliptic curves. Exploiting this action in cryptosystems presented significant challenges, until supersingular elliptic curves were considered [CLM+18]. Complex multiplication is typically associated to *ordinary* (i.e., non-supersingular) elliptic curves. The notion of oriented elliptic curve generalizes the methods of complex multiplication to the supersingular case. In Chapter 2, we study oriented elliptic curves and their applications in cryptography. It is organised around the following contributions.

- While this "group action" branch of isogeny-based cryptography looks substantially different from the "isogeny path and endomorphism ring" problems discussed in Chapter 1, we prove in the article ⟦Wes22a⟧ that the group action inversion problem is still equivalent to the problem of computing endomorphism rings (but now, for *oriented* elliptic curves). This result further reinforces the foundational status of the endomorphism ring problem: its presumed hardness also governs the security of the "group action" branch of the field.
- In the article ⟦HW23⟧, with my PhD student Arthur Herlédan Le Merdy, we study the concrete hardness of the "oriented" endomorphism ring problem, describing and analysing the fastest classical and quantum algorithms. In the process, we prove that the complexities claimed by the previous best heuristic algorithms for the group action inversion problem can actually be achieved assuming the Generalised Riemann Hypothesis. In addition, we obtain the first polynomial time algorithm for the problem of turning an arbitrary orientation into a *primitive* orientation (a kind of "best possible" orientation).
- In the article ⟦DFK+23⟧, we introduce SCALLOP. The group acting on supersingular curves is an ideal class group. Class groups are generally hard to compute, and that has been a source of difficulty in the design of cryptosystems. SCALLOP offers a framework to define a hard-to-invert action by an easy-to-compute class group. This brings this group action one step closer to the capabilities of the discrete logarithm paradigm, while preserving its claim for post-quantum security.

The following is a complete list of my articles related to the notion of oriented elliptic curves and their applications in cryptography.

⟦HW23⟧ Arthur Herlédan Le Merdy and Benjamin Wesolowski. The supersingular endomorphism ring problem given one endomorphism. Cryptology ePrint Archive, Paper 2023/1448, 2023. https://eprint.iacr.org/2023/1448.

⟦ACD+23⟧ Sarah Arpin, James Clements, Pierrick Dartois, Jonathan Komada Eriksen, Péter Kutas, and Benjamin Wesolowski. Finding orientations of supersingular elliptic curves and quaternion orders. Cryptology ePrint Archive, Paper 2023/1268, 2023. https://eprint.iacr.org/2023/1268.

⟦DFK+23⟧ Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography – PKC 2023*, volume 13940 of *Lecture Notes in Computer Science*, pages 345–375. Springer, 2023.

[[CHVW22] Wouter Castryck, Marc Houben, Frederik Vercauteren, and Benjamin Wesolowski. On the decisional Diffie-Hellman problem for class group actions on oriented elliptic curves. *Research in Number Theory*, 8(4):99, 2022. Proceedings of the Fifteenth Algorithmic Number Theory Symposium – ANTS-XV.

[[Wes22a] Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, volume 13277 of *Lecture Notes in Computer Science*, pages 345–371. Springer, 2022.

Chapter 3: Ideal lattices. A lattice is a discrete subgroup in a Euclidean vector space. The problem of finding a shortest non-zero vector in a lattice (the *shortest vector problem*, or SVP) is a central hard problem in complexity theory, and the heart of *lattice-based cryptography*. Lattices appear naturally in algebraic number theory: the ring of integers of a number field is a lattice, and so is any (fractional) ideal. These *ideal lattices* have a special place in computational number theory, and provide a powerful playground for cryptography, via the corresponding Id-SVP problem. In Chapter 3, we study ideal lattices through the development and application of a new tool: random walks in the space of ideal lattices. It is organized around the following contributions.

- In the article [[BDPW20]], we describe random walks in the Arakelov class group, and prove their rapid-equidistribution properties (assuming the Generalized Riemann Hypothesis). The Arakelov class group is essentially the space of ideal lattices up to isometry. As a first application of this new randomization tool, we prove that Id-SVP is hard *on average* for uniformly random ideal lattices (with respect to the Haar measure): if there exist hard instances of Id-SVP, then it is hard for random ideal lattices. *Average hardness* is of primary interest for cryptographic applications. These random walks are the main topic of the PhD thesis of Koen de Boer, coauthor of [[BDPW20]] and [[BPW24]]. His thesis, of which I was *co-promotor*, was successfully defended in 2022.
- Average hardness is defined with respect to a distribution on instances (here, ideal lattices), and while uniformity for the Haar measure is mathematically the most natural candidate, it is not the best suited for applications. In the article [[FPSW23]], we prove that Id-SVP is hard on average for uniformly random prime ideals of bounded norm. This distribution is well-suited, for instance, for application to the NTRU cryptosystem.
- In the article [[BPW24]], we solve a recurring problem in computational number theory: sampling ideals in a given class, with a prescribed property (smooth, near-prime...). While it is easy to design a *heuristic* algorithm for this task (i.e., relying on unproven *ad hoc* assumptions), we provide the first *rigorous* method, under the Generalized Riemann Hypothesis. To illustrate the power of this technique, we describe the first rigorous subexponential time algorithm for some of the most emblematic problems of the domain: computing class groups and unit groups of arbitrary number fields. Previous rigorous algorithms were restricted to quadratic fields.

The following is a complete list of my articles related to the notion of ideal lattices and their applications in cryptography and computational number theory.

⟦BPW24⟧ Koen de Boer, Alice Pellet-Mary, and Benjamin Wesolowski. Rigorous methods for computational number theory. Preprint available on demand, 2024.

⟦FPSW23⟧ Joël Felderhoff, Alice Pellet-Mary, Damien Stehlé, and Benjamin Wesolowski. Ideal-SVP is hard for small-norm uniform prime ideals. In Guy N. Rothblum and Hoeteck Wee, editors, *Theory of Cryptography – 21st International Conference, TCC 2023*, volume 14372 of *Lecture Notes in Computer Science*, pages 63–92. Springer, 2023.

⟦CDW21⟧ Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Mildly short vectors in cyclotomic ideal lattices in quantum polynomial time. *Journal of the ACM*, 68(2):8:1–8:26, 2021.

⟦BDPW20⟧ Koen de Boer, Léo Ducas, Alice Pellet-Mary, and Benjamin Wesolowski. Random self-reducibility of Ideal-SVP via Arakelov random walks. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 243–273. Springer, 2020.

⟦DPW19⟧ Léo Ducas, Maxime Plançon, and Benjamin Wesolowski. On the shortness of vectors to be found by the Ideal-SVP quantum algorithm. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, volume 11692 of *Lecture Notes in Computer Science*, pages 322–351. Springer, 2019.

⟦CDW17⟧ Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short Stickelberger class relations and application to Ideal-SVP. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, volume 10210 of *Lecture Notes in Computer Science*, pages 324–348. Springer, 2017.

Other articles. The following is a complete list of my articles on themes not covered in this manuscript.

Discrete logarithms in finite fields. The following articles concern the problem of computing discrete logarithms in finite fields. We prove in ⟦KW22⟧ that they can be computed in quasi-polynomial time in finite fields of small characteristic. This result improves upon the subexponential complexity proved by Pomerance in 1987 [Pom87]. The quasi-polynomial complexity had been conjectured to be reachable since [BGJT14], where a first heuristic algorithm was proposed. We illustrate the power of the method with the record computation of a 30750-bit discrete logarithm in ⟦GKL$^+$21⟧.

⟦KW22⟧ Thorsten Kleinjung and Benjamin Wesolowski. Discrete logarithms in quasi-polynomial time in finite fields of fixed characteristic. *Journal of the American Mathematical Society*, 35(2):581–624, 2022.

⟦GKL$^+$21⟧ Robert Granger, Thorsten Kleinjung, Arjen K. Lenstra, Benjamin Wesolowski, and Jens Zumbrägel. Computation of a 30750-bit binary field discrete logarithm. *Mathematics of Computation*, 90(332):2997–3022, 2021.

[KW19] Thorsten Kleinjung and Benjamin Wesolowski. A new perspective on the powers of two descent for discrete logarithms in finite fields. In *Proceedings of the Thirteenth Algorithmic Number Theory Symposium – ANTS-XIII*, volume 2, pages 343–352. The Open Book Series, Mathematical Sciences Publishers, 2019.

Isogenies between Drinfeld modules. The following article proves that one can compute isogenies in polynomial time between Drinfeld modules over finite fields. This breaks Drinfeld analogs of isogeny-based cryptosystems [LS22]. Former attempts had exponential complexity [JN19, CGS20].

[Wes24] Benjamin Wesolowski. Computing isogenies between finite Drinfeld modules. *To appear in IACR Communications in Cryptology*, 2024.

Isogeny graphs of ordinary abelian varieties. The following articles concern the graphs formed by isogenies between ordinary abelian varieties in higher dimension.

[JW19] Dimitar Jetchev and Benjamin Wesolowski. Horizontal isogeny graphs of ordinary abelian varieties and the discrete logarithm problem. *Acta Arithmetica*, 187:381–404, 2019.

[Wes19b] Benjamin Wesolowski. Generating subgroups of ray class groups with small prime ideals. In *Proceedings of the Thirteenth Algorithmic Number Theory Symposium – ANTS-XIII*, volume 2, pages 461–478. The Open Book Series, Mathematical Sciences Publishers, 2019.

[BJW17] Ernest Hunter Brooks, Dimitar Jetchev, and Benjamin Wesolowski. Isogeny graphs of ordinary abelian varieties. *Research in Number Theory*, 3(1):28, 2017.

Verifiable delay functions and randomness. The following articles concern verifiable delay functions and the generation of random numbers. In [LW17], we describe the first scalable protocol for the public and trustworthy generation of random numbers, using a new *slow-timed hash function*. This notion of slow hash function was formalised and generalised as a *verifiable delay function* (VDF) in [BBBF18]. In [Wes19a], we construct the first practical VDF.

[BFH+24] Alex Biryukov, Ben Fisch, Gottfried Herold, Dmitry Khovratovich, Gaëtan Leurent, María Naya-Plasencia, and Benjamin Wesolowski. Cryptanalysis of algebraic verifiable delay functions. *To appear in Advances in Cryptology – CRYPTO 2024*, 2024.

[BKSW23] Karim Belabas, Thorsten Kleinjung, Antonio Sanso, and Benjamin Wesolowski. A note on the low order assumption in class group of an imaginary quadratic number fields. *Mathematical Cryptology*, 3:44–51, Jul. 2023.

[WW20] Ryan Williams and Benjamin Wesolowski. Lower bounds for the depth of modular squaring. IACR Cryptology ePrint Archive, Report 2020/1461, 2020. https://eprint.iacr.org/2020/1461.

[Wes20] Benjamin Wesolowski. Efficient verifiable delay functions. *Journal of Cryptology*, 33(4):2113–2147, 2020.

⟦Wes19a⟧ Benjamin Wesolowski. Efficient verifiable delay functions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 379–407. Springer, 2019.

⟦PW18⟧ Cécile Pierrot and Benjamin Wesolowski. Malleability of the blockchain's entropy. *Cryptography and Communications*, 10(1):211–233, 2018.

⟦LW17⟧ Arjen K. Lenstra and Benjamin Wesolowski. Trustworthy public randomness with sloth, unicorn, and trx. *International Journal of Applied Cryptography*, 3(4):330–343, 2017.

Broadcast encryption. In the following article, we design a new attribute-based broadcast encryption scheme with small keys.

⟦WJ15⟧ Benjamin Wesolowski and Pascal Junod. Ciphertext-policy attribute-based broadcast encryption with small keys. In Soonhak Kwon and Aaram Yun, editors, *Information Security and Cryptology – ICISC 2015*, volume 9558 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2015.

Notation and terminology

Rings and fields. We write $\mathbf{Z}, \mathbf{Q}, \mathbf{R}, \mathbf{C}$ and $\mathbf{F}_q$ for the ring of integers, the fields of rational, real and complex numbers, and a finite field with q elements. For any field K, we write $\overline{K}$ for an algebraic closure. For any ring R, we write $R^\times$ for the multiplicative group of invertible elements.

Complexities. We write $f = O(g)$ for the classic big O notation. We use the soft-O notation $\tilde{O}(g) = \log(g)^{O(1)} \cdot O(g)$, and the polynomial growth notation $\mathrm{poly}(f_1, \ldots, f_n) = (f_1 + \cdots + f_n)^{O(1)}$. We use the classical L-notation for subexponential growth

$$L_x(\alpha) = \exp\left(O(\log x)^\alpha (\log\log x)^{1-\alpha}\right).$$

The logarithm function log is in base 2.

Sets and probabilities. For any set S, we write $\#S$ for its cardinality. If $\mathcal{D}$ is a probability distribution, we write $x \leftarrow \mathcal{D}$ to signify that x is a random variable sampled with distribution $\mathcal{D}$. If S is a finite set, we also write $x \leftarrow S$ for x uniformly distributed in S.

The Generalized Riemann Hypothesis. Several of the results presented in this manuscript are conditional under the Riemann Hypothesis for Hecke L-functions, which we will refer to as the Generalized Riemann Hypothesis (abbreviated GRH). When needed, this assumption is explicitly mentioned in the statement. Note that the proof of such results do not use GRH in its original form, but rely on some of its established consequences, such as the effective Chebotarev density theorem of Lagarias and Odlyzko [LMO79].

The "heuristic" terminology. The meaning of "heuristic" in computational number theory differs from other areas of computer science. Let us clarify this terminology. In computational number theory (and in the present manuscript) an algorithm is said to be *heuristic* if it is believed to work, but its analysis relies on unproven *ad hoc* assumptions. For instance, an algorithm may craft an integer, and one expects this integer to "behave" like a uniformly random integer of a certain size; we can then deduce the probability of certain events, like it being prime. It is often hard to prove such a behavior, but

its assumption unlocks the analysis of the algorithm. The underlying assumption, the *heuristic assumption*, may be supported by experiments and intuition.

We make an informal distinction between *heuristic assumptions* and standard number-theoretic conjectures (like the Generalized Riemann Hypothesis). Heuristic assumptions are generally *ad hoc*, highly dependent on the situation, and are often in the spirit of: "at this point in the algorithm, all goes well". Sometimes, a heuristic assumption is not expected to be literally true: in the above example, the integer may not be uniformly distributed, and one only expects it to "behave uniformly".

A few contributions presented in this manuscript consist in turning a *heuristic* result into a *rigorous* result. This means that a heuristic algorithm already existed, and we provide a fully rigorous algorithm (either unconditional or subject to GRH).

1

Supersingular isogenies and endomorphisms

In this chapter, we present contributions related to the computational problem of finding isogenies between supersingular elliptic curves and its applications in cryptography. We explore connections between this problem, random walks in isogeny graphs, the problem of computing endomorphisms of elliptic curves, and the power of higher dimensional isogenies. We conclude this chapter by presenting the SQIsign digital signature scheme, which has benefited from all these advances from its original design to its latest improvements.

This chapter is built around the presentation of the articles (in order of appearance):

[[PW24]] Aurel Page and Benjamin Wesolowski. The supersingular endomorphism ring and one endomorphism problems are equivalent. *To appear in Advances in Cryptology – EUROCRYPT 2024*, 2024.

[[Wes21]] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *62nd IEEE Annual Symposium on Foundations of Computer Science – FOCS 2021*, pages 1100–1111. IEEE, 2021.

[[MMP+23]] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023.

[[DKL+20]] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.

[[DLRW24]] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: new dimensions in cryptography. *To appear in Advances in Cryptology – EUROCRYPT 2024*, 2024.

1.1. Introduction

Given a field k of characteristic $p > 3$, and two parameters $A, B \in k$ with $4A^3 + 27B^2 \neq 0$, the equation

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

defines a so-called *elliptic curve* over k. Elliptic curves can be more abstractly defined as abelian varieties of dimension 1, and the equation $y^2 = x^3 + Ax + B$ is merely a *short Weierstrass model* of the elliptic curve. Given an elliptic curve E, its set $E(k)$ of *k-rational points* consists of the pairs $(x, y) \in k^2$ satisfying the curve equation, together with an extra point 0_E "at infinity" (a projective solution of the equation). They naturally form an abelian group, written additively, where 0_E is the neutral element.

Elliptic curves have long had a central place in cryptography, since Miller [Mil86] and Koblitz [Kob87] proposed in 1986 to run the Diffie–Hellman key exchange protocol [DH76] over elliptic curves. In a world where large-scale quantum computers would become available, Shor's algorithm [Sho97] would render the Diffie–Hellman protocol obsolete, with or without elliptic curves. Yet, in the quest for post-quantum cryptography, elliptic curves have kept a central place thanks to the emergence of new computational problems that seem to resist quantum algorithms: isogeny problems. The first isogeny-based cryptosystems were proposed by Couveignes in 1997 [Cou06]. This work was only made public in 2006, when the idea reemerged in [RS06]. Supersingular elliptic curves appeared to be particularly well suited for the design of post-quantum cryptosystems, starting with the CGL hash function [CLG09], followed by the SIDH key exchange [JD11]. A wealth of other public-key protocols [CLM+18, DKPS19, Cos20, BMP23], ⟦DFK+23⟧, signature schemes [YAJ+17, DG19, BKV19, GPS20], ⟦DKL+20, DLRW24⟧ and other cryptosystems [DMPS19, BKW20] have since been proposed, built on the presumed hardness of isogeny problems.

Let E_1 and E_2 be two elliptic curves defined over k. An *isogeny* $\varphi : E_1 \to E_2$ is a non-constant rational map (i.e., coordinates of the output are given by fractions of polynomials in the input coordinates) that sends 0_{E_1} to 0_{E_2}. This rather simple definition automatically implies strong properties: an isogeny is a group homomorphism from $E_1(k)$ to $E_2(k)$, and its kernel over the algebraic closure, written $\ker(\varphi)$, is finite. This notion naturally leads to a computational problem.

Problem 1.1 (The isogeny problem, informally)**.** Given two elliptic curves E_1 and E_2 over a finite field, find, if it exists, an isogeny $\varphi : E_1 \to E_2$.

Versions of this problem are believed to be hard, and isogeny-based cryptography leverages this hardness. This problem is most often considered for so-called *supersingular elliptic curves*. Then, a solution is guaranteed to exist (two supersingular elliptic curves over the same field are always connected by an isogeny), and the isogeny problem appears to be at its hardest for such curves. Almost all isogeny-based cryptography considers supersingular elliptic curves.

To formally define an isogeny problem, one must specify what it means to "find an isogeny". Encoding an isogeny is not a straightforward task. It has rapidly appeared convenient to consider isogenies formed as compositions of simple building-blocks. An ℓ-isogeny, for ℓ a small prime, is such a family of simple, easy-to-work-with isogenies. An ℓ-isogeny path is a composition of ℓ-isogenies. Here is another motivation to work with supersingular elliptic curves: any pair is connected by an ℓ-isogeny path (whatever ℓ, including the typical choice $\ell = 2$). This leads to the first fully-specified version of the isogeny problem: the supersingular ℓ-isogeny path problem.

Problem 1.2 (ℓ-IsogenyPath)**.** Given a prime p, and two supersingular elliptic curves E_1 and E_2 over $\mathbf{F}_{p^2}$, find an ℓ-isogeny path from E_1 to E_2.

This problem was first considered in [CLG09], and has since assumed a foundational role in isogeny-based cryptography, raising questions such as:

- How hard is the ℓ-IsogenyPath problem? How efficient are the best algorithms? Is it the hardest of its kind?
- What kind of cryptosystems can be built from it? Can it, or variants, be used to prove the security, or to attack schemes?

A large part of the work towards answering these questions consists in defining and analyzing related problems: some that may be easier to study, or easier to connect to cryptosystems. Here are a few such problems of central interest:

- EndRing: given a supersingular elliptic curve E, compute its endomorphism ring $\mathrm{End}(E)$. An endomorphism is an isogeny from the curve to itself (or the zero-morphism). This problem has been studied as early as [Koh96], originally motivated by the importance of these structures in arithmetic geometry.
- OneEnd: given a supersingular elliptic curve, find one non-scalar endomorphism. This is a straightforward simplification of EndRing: instead of finding all endomorphisms, can one find even a single one? This problem naturally emerges in cryptosystems: it supports the collision-resistance of [CLG09] or the soundness of SQIsign [[DKL+20]].
- Isogeny: given two supersingular elliptic curves, find an isogeny between them, in any form that allows to efficiently evaluate it on points. This is perhaps the most natural form of the isogeny problem, as, contrary to ℓ-IsogenyPath, the solution is not restricted to special kinds of isogenies.
- Interpolation: given two supersingular elliptic curves, and the images of a few points through an unknown isogeny, find the isogeny. This version of the isogeny problem supports the SIDH key exchange protocol [JD11].

In this chapter, we study these problems. We connect them to one another, study the best algorithms for their resolution, and how to build cryptosystems from their presumed hardness.

1.1.1. Contributions and organisation of the chapter.

The chapter is organized as follows.

Random walks in isogeny graphs. In Section 1.2, we define ℓ-isogeny graphs, whose vertices are elliptic curves, and edges represent ℓ-isogenies between them. The ℓ-IsogenyPath problem is the pathfinding problem in these graphs. Pizer proved in [Piz90] that they are Ramanujan graphs: an optimal form of expander graphs, where random walks rapidly converge to the uniform distribution. This is a key property for cryptographic applications and for the analysis of ℓ-IsogenyPath and its friends. We illustrate this fact with a few simple examples.

We then present isogeny graphs of "higher level" as defined and analysed in our work [[PW24]]. It is often useful to consider some additional structure attached to the elliptic curves. The vertices of the graph, instead of being curves, could be pairs (E, P) where $P \in E$ is a point, or (E, α) where $\alpha \in \mathrm{End}(E)/N\,\mathrm{End}(E)$ (for some integer N). Edges would be isogenies that preserve this additional structure. In the article [[PW24]], we introduce a general framework for such graphs, and prove that, like in the classical case, random walks equidistribute optimally.

The ℓ-Isogeny Path and Endomorphism Ring problems. It was soon identified that the problems ℓ-IsogenyPath and EndRing are closely related, and heuristic computational reductions between them were described in [EHL+18]. In [[Wes21]], we prove that these two problems are indeed equivalent, assuming the Generalised Riemann Hypothesis (GRH). As in [EHL+18], this is done by passing through a third equivalent problem: MaxOrder. We present these results in Section 1.3.

Finding one endomorphism is as hard as finding them all. In Section 1.4, we present the main result of [[PW24]]: the EndRing and OneEnd problems are equivalent. In other words, finding one endomorphism of a supersingular elliptic curve is as hard as finding them all. We sketch the reduction, which makes critical use of the equidistribution for random walks in higher isogeny graphs discussed in Section 1.2.

We then present a number of consequences. First, assuming the hardness of EndRing, the Charles–Goren–Lauter hash function [CLG09] is collision-resistant. Second, EndRing is equivalent to Isogeny. Third, there exists an unconditional probabilistic algorithm to solve EndRing in time $\tilde{O}(p^{1/2})$, a result which previously required to assume GRH.

The fall of SIDH. Supersingular Isogeny Diffie-Hellman (or SIDH [JD11]) is a key exchange protocol proposed in 2011 by Jao and De Feo. It has been among the most popular isogeny-based schemes, and the first key exchange using supersingular elliptic curves. However, its security relies not on the ℓ-IsogenyPath problem, but on its cousin Interpolation: given two elliptic curves and images of a few points through some unknown isogeny, find the isogeny. Of course, there is a spectrum of hardness for this problem, depending on the order of the group generated by the revealed points. Revealing too many points was known to be dangerous, but SIDH has long been believed to be in a secure regime.

In Section 1.5, we recount the earthquake that struck isogeny-based cryptography in 2022, when the series of work [CD23], [[MMP+23]] and [Rob23] solved the Interpolation problem in an unforeseen level of generality. In particular, SIDH is now completely broken. While carefully delineating our own contribution through [[Wes22b, MMP+23]], we present the result in a greater level of generality to include the final nail hammered by Robert [Rob23].

SQIsign: signing with isogenies. In Section 1.6, we present the digital signature scheme SQIsign. We introduced this scheme in [[DKL+20]], improved it in [[DLLW23]], and redesigned it in SQIsignHD [[DLRW24]]. SQIsign is currently submitted to the NIST call for standardisation of post-quantum digital signature schemes. It boasts the smallest public keys and signatures combined of today's post-quantum portfolio.

We describe the general structure of SQIsign, with a focus on the variant SQIsignHD, which makes constructive use of the recent advances on the Interpolation problem discussed in Section 1.5.

1.1.2. Preliminaries on isogenies. In this section, we briefly review some of the most important properties of isogenies required in the rest of the text. We refer the reader to [Sil86] for a detailed account of the theory of elliptic curves and isogenies. In this text, all elliptic curves are defined over a finite field. We denote by $\mathbf{F}_{p^n}$ a finite field with p^n elements and characteristic p, and by $\overline{\mathbf{F}}_p$ an algebraic closure of $\mathbf{F}_{p^n}$.

Isomorphisms. An isogeny $\varphi : E \to E'$ is an *isomorphism* if there exists an isogeny $\hat{\varphi} : E' \to E$ such that $\hat{\varphi} \circ \varphi$ (and thereby $\varphi \circ \hat{\varphi}$) is the identity. Isomorphisms from E to itself form the group of automorphisms $\mathrm{Aut}(E)$. For E defined over $\mathbf{F}_{p^n}$, the j-invariant $j(E) \in \mathbf{F}_{p^n}$ is a complete invariant of the isomorphism class of E over the algebraic closure $\overline{\mathbf{F}}_p$: there exists an isomorphism $E \to E'$ over $\overline{\mathbf{F}}_p$ if and only if $j(E) = j(E')$.

Degree and separability. Intuitively, the *degree* of an isogeny φ measures how "complicated" it is. A simple measure of this complexity would be $\#\ker(\varphi)$, and this is close to the correct definition. The degree is actually an integer of the form $p^k \cdot \#\ker(\varphi)$, where $p > 0$ is the characteristic of the field. When $k = 0$, i.e., when $\deg(\varphi) = \#\ker(\varphi)$, the isogeny is called *separable*. The case $k > 0$ can only arise from a special isogeny: the p^k-Frobenius isogeny defined as

$$\phi_{p^k}^E : E \longrightarrow E^{(p^k)} : (x, y) \longmapsto (x^{p^k}, y^{p^k}).$$

For any isogeny φ, there is a maximal integer k such that φ factors in the form $\psi \circ \phi_{p^k}^E$. We then define $\deg(\varphi) = p^k \cdot \#\ker(\varphi)$.

The kernel encodes most of the information about a separable isogeny. Indeed, for any separable isogenies $\varphi_1 : E \to E_1$ and $\varphi_2 : E \to E_2$, if $\ker(\varphi_1) = \ker(\varphi_2)$, then there exists an isomorphism $\psi : E_1 \to E_2$ such that $\varphi_2 = \psi \circ \varphi_1$. Moreover, for any finite subgroup $G \subset E$, there exists a separable isogeny φ_G with kernel G. In virtue of its unicity (up to isomorphism), we write E/G for the codomain of φ_G.

Prime decomposition. If $\deg(\varphi) = \ell$ is a prime number, we say that φ is an ℓ-isogeny. Any isogeny can be written as a composition of ℓ-isogenies: if $\deg(\varphi) = \prod_i \ell_i$ with each ℓ_i prime, there exist isogenies φ_i of appropriate domains and codomains such that each φ_i is an ℓ_i-isogeny and $\varphi = \varphi_n \circ \cdots \circ \varphi_1$.

Endomorphisms. An *endomorphism* of E is an isogeny from E to itself (or the zero morphism). The collection of all endomorphisms of E (over $\overline{\mathbf{F}}_p$) forms the endomorphism ring $\mathrm{End}(E)$. As its name suggests, it is a ring for pointwise addition, and composition of maps. For any integer $m \in \mathbf{Z}$, the multiplication-by-m map $[m]_E : E \to E : P \mapsto m \cdot P$ is an endomorphism. Abusing notation, we will often consider $\mathbf{Z}$ as a subring of $\mathrm{End}(E)$, since this construction $m \mapsto [m]_E$ is injective. Elements of this subring $\mathbf{Z} \subset \mathrm{End}(E)$ are called *scalar* endomorphisms.

Dual. For any isogeny $\varphi : E_1 \to E_2$, there exists a unique isogeny $\hat{\varphi} : E_2 \to E_1$, the *dual* of φ, such that $\hat{\varphi} \circ \varphi = [\deg(\varphi)]$.

Torsion and supersingularity. The m-torsion of E is the subgroup $E[m] = \ker([m]_E)$. Multiplication by m is separable if and only if $\gcd(m, p) = 1$, in which case $E[m] \cong (\mathbf{Z}/m\mathbf{Z})^2$. The situation at the prime p leads to a dichotomy:

- either $E[p^n] \cong \mathbf{Z}/p^n\mathbf{Z}$ for all $n \geq 0$, and we say E is *ordinary*, or
- $E[p^n] = \{0_E\}$ for all $n \geq 0$, and we say E is *supersingular*.

1.1.3. Computing with isogenies. Isogenies are rather abstract objects, yet we wish to work with them computationally: store them, evaluate them. One could represent an isogeny by explicit formulas, as a rational map, but that rapidly becomes impractical with increasing degrees. Alternatively, an isogeny can be described by generators of its kernel. Let $G \subset E$ be a subgroup generated by $(P_1, \ldots, P_n)$. The data of $(P_1, \ldots, P_n)$ then encodes the isogeny $\varphi_G : E \to E/G$. We call this the *kernel representation* of an isogeny, and Vélu proposed formulas which given the generators of G, return the codomain E/G and evaluate φ_G at any point in $O(\#G)$ arithmetic operations [Vél71]. This complexity has recently been improved to $O(\sqrt{\#G})$ [BDFLS20]. This method is well-suited for isogenies of reasonably small degree. Yet, in cryptographic applications, one will routinely encounter isogenies of degree of the order of 2^{256}. Another solution is needed.

One can exploit the multiplicativity of the degree: composing small degree isogenies rapidly produces large degree ones. Consider a sequence $\varphi_i : E_{i-1} \to E_i$. The composition $\varphi_n \circ \cdots \circ \varphi_1$ can be represented by a tuple $(\varphi_1, \ldots, \varphi_n)$ where each φ_i is in kernel representation. The length grows linearly in n, but the degree grows exponentially. One

can thus efficiently manipulate isogenies of degree 2^n as sequences of n isogenies of degree 2. We call such a sequence an *isogeny path*.

The isogeny path representation is an example of an *efficient representation*: a representation of an isogeny φ that allows to store and evaluate it in polynomial time in $\log(\deg(\varphi))$.

Definition 1.3 (Efficient representation)**.** Let $\mathscr{A}$ be a polynomial time algorithm. It is an *efficient isogeny evaluator* if for any $D \in \{0,1\}^*$ such that $\mathscr{A}(\texttt{validity}, D)$ outputs $\top$, there exists an isogeny $\varphi : E \to E'$ (defined over some finite field $\mathbf{F}_q$) such that:

(1) on input $(\texttt{curves}, D)$, $\mathscr{A}$ returns (E, E'),
(2) on input $(\texttt{degree}, D)$, $\mathscr{A}$ returns $\deg(\varphi)$,
(3) on input $(\texttt{eval}, D, P)$ with $P \in E(\mathbf{F}_{q^k})$, $\mathscr{A}$ returns $\varphi(P)$.

If furthermore D is of polynomial size in $\log(\deg\varphi)$ and $\log q$, then D is an *efficient representation* of φ (with respect to $\mathscr{A}$).

When we say that an isogeny is in efficient representation, the algorithm $\mathscr{A}$ is often left implicit. There are only a handful of known algorithms to evaluate isogenies, so one can think of $\mathscr{A}$ as an algorithm that implements each of these, and D would start with an indicator of which algorithm to use. In the following sections, we will see more recent and powerful examples of efficient representations — including a *universal* method: the *interpolation representation* discussed in Section 1.5.2.

1.2. Random walks in isogeny graphs

1.2.1. The standard supersingular isogeny graph. An *isogeny graph* is a multigraph where vertices are elliptic curves up to isomorphism, and edges represent certain isogenies between them. Fix two distinct prime numbers p and ℓ. The (full) ℓ-isogeny graph over $\overline{\mathbf{F}}_p$ is defined as follows:

- Its vertices are the (isomorphism classes of) elliptic curves over $\overline{\mathbf{F}}_p$.
- Its edges are all the isogenies of degree ℓ (also called ℓ-isogenies). More precisely, there is an edge of multiplicity m from E_1 to E_2 if there are m distinct subgroups $G \subset E$ of order ℓ such that $E_2 \cong E_1/G$.

This graph has infinitely many vertices, and exactly one of its connected components is finite: the *supersingular* component $SS_\ell(p)$, whose vertices are the supersingular elliptic curves. We call this component *the supersingular ℓ-isogeny graph over $\overline{\mathbf{F}}_p$*. Almost all isogeny-based cryptography takes place in this component. Here are a few first observations about this graph.

- The supersingular ℓ-isogeny graph counts $p/12 + O(1)$ vertices (isomorphism classes of supersingular elliptic curves). They each admit a model defined over $\mathbf{F}_{p^2}$.
- The graph is $(\ell+1)$-out-regular. One can efficiently enumerate the neighbors of any given vertex (in time polynomial in ℓ and $\log p$), and thereby navigate in the graph.
- The ℓ-IsogenyPath problem is simply the pathfinding problem in the supersingular ℓ-isogeny graph.

1.2.2. Rapid mixing of random walks. Given a vertex E_0 of the ℓ-isogeny graph, a random walk (of length n) from E_0 is a sequence $(E_i)_{i=0}^n$ where each E_{i+1} is a random neighbor of E_i. Unless otherwise specified, we will consider the uniform distribution on neighbors.

These random walks are useful in isogeny-based cryptography for one particular reason: they have rapid mixing properties. In fact, the supersingular ℓ-isogeny graph is a Ramanujan graph, a family of graphs in which random walks equidistribute optimally: the

target of a random walk of length at least $O(\log p)$ is indistinguishable from a uniformly random vertex.

For a more formal statement, let us recall the notion of total variation distance. Given a random variable X with values in a discrete set $\mathscr{X}$, we say it has distribution $f : \mathscr{X} \to [0,1]$ if $f(x) = \Pr[X = x]$ for every $x \in \mathscr{X}$. We also write $f(A) = \sum_{x \in A} f(x)$ for any $A \subseteq \mathscr{X}$. For two distributions f_1 and f_2 over the same set $\mathscr{X}$, their *total variation distance* is

$$\frac{1}{2}\|f_1 - f_2\|_1 = \frac{1}{2}\sum_{x \in \mathscr{X}} |f_1(x) - f_2(x)| = \sup_{A \subseteq \mathscr{X}} |f_1(A) - f_2(A)|.$$

The rapid-mixing of supersingular ℓ-isogeny graphs is the following well-known proposition. In Section 1.2.4, we present the much more general version of ⟦PW24, Theorem 3.10⟧.

Proposition 1.4. *Let E be a supersingular elliptic curve over $\mathbf{F}_{p^2}$, and $\ell \neq p$ a prime number. Let $\varepsilon > 0$. There is a bound $n = O(\log_\ell(p) - \log_\ell(\varepsilon))$ such that the endpoint of a uniform random walk of length at least n from E in the ℓ-isogeny graph is at total variation distance at most ε from the stationary distribution f, which is $f(E) = \frac{24}{(p-1)\#\operatorname{Aut}(E)}$.*

Proof. This is a standard consequence of Pizer's proof that the supersingular ℓ-isogeny graph is Ramanujan [Piz90]. Details can be found, for instance, in ⟦BCC⁺23, Theorem 11⟧ for the length of the walk, and in ⟦BCC⁺23, Theorem 7, Item 2⟧ for the description of the stationary distribution. □

The stationary distribution is at statistical distance $O(1/p)$ of the uniform distribution (because $\#\operatorname{Aut}(E) = 2$ for almost all curves, and $\#\operatorname{Aut}(E) = O(1)$ for the rare exceptions). For all cryptographic purposes, the stationary and uniform distributions are indistinguishable, and we often conflate them in the rest of the text.

1.2.3. Cryptographers care about random walks. Random walks play a key role in isogeny-based cryptography. A critical point is that cryptography requires computational problems that are hard *on average*: problems which cannot be solved efficiently for *random inputs*, with *non-negligible probability*. Thus designing and analysing cryptosystems requires good randomisation tools. We now illustrate this with a few classical applications of random ℓ-isogeny walks.

Average-case problems. Here is an average-case version of ℓ-IsogenyPath.

Problem 1.5 (Average-case ℓ-IsogenyPath)**.** Given a prime p, and two uniformly random supersingular elliptic curves E_1 and E_2 over $\mathbf{F}_{p^2}$, find an ℓ-isogeny path from E_1 to E_2.

One can prove that the CGL hash function [CLG09] is preimage-resistant if *average-case* ℓ-IsogenyPath is hard (i.e., if no efficient algorithm solves it with good probability). The CGL hash function was the first isogeny-based construction using random walks in the supersingular ℓ-isogeny graph. The idea is the following. Fixing a reference curve E_0, a binary string $x \in \{0,1\}^n$ encodes a (non-backtracking) walk $\varphi_x : E_0 \to E_x$ of length n in the 2-isogeny graph (the graph has degree 3, so there are two possibilities for each "next step", arbitrarily labelled 0 and 1; for the first step, we arbitrarily discard one of the three possibilities). The CGL hash function is the function

$$\mathrm{CGL}_{E_0} : \{0,1\}^* \longrightarrow \mathbf{F}_{p^2} : x \longmapsto j(E_x)$$

which sends any binary string to the j-invariant of the target curve E_x of the encoded walk φ_x. Finding a preimage for that function CGL_{E_0} thus amounts to finding a path $E_0 \to E_x$ in the ℓ-isogeny graph. The cryptographic notion of *preimage-resistance* is an "average case" property: CGL_{E_0} is preimage-resistant if it is hard to find preimages *with good probability for uniformly random inputs of a certain length*. This is where random

walks come in: a random input $x \in \{0,1\}^n \subset \{0,1\}^*$ for f encodes a random walk of length n. When n is large enough, the output of this random walk is close to uniform. From there, it is easy to deduce that if *average-case* ℓ-IsogenyPath is hard, then the CGL hash function is preimage-resistant (with small technicalities to deal with, like the fact that $\{0,1\}^*$ only encodes non-backtracking walks).

Worst-case to average-case reductions. The CGL construction is already an interesting first application of the rapid-mixing property, but the result is not yet satisfactory. The core assumption of isogeny-based cryptography is the *worst-case* hardness of the ℓ-IsogenyPath problem: no algorithm can solve it efficiently on *every* input. Clearly, if it is hard on average, then it is hard in the worst case. But we wish to prove the converse: if it is hard in the worst case (our core assumption), then it is hard on average. Random walks unlock such a proof. Suppose there is an efficient algorithm $\mathscr{A}$ for *average-case* ℓ-IsogenyPath, and let E_1 and E_2 be any instance of ℓ-IsogenyPath (the worst case). Proceed as follows:

(1) First, generate two random ℓ-isogeny walks $\varphi_i : E_i \to E_i'$ such that each E_i' is indistinguishable from uniform.
(2) Call $\mathscr{A}$ on input E_1' and E_2' (an average-case instance); it returns an ℓ-isogeny path $\psi : E_1' \to E_2'$ with good probability.
(3) Return $\hat{\varphi}_2 \circ \psi \circ \varphi_1$.

This is a simple example of a *worst-case to average-case reduction*.

Design and analysis of algorithms. As a last illustration of the power of random walks in ℓ-isogeny graphs, we sketch the fastest known algorithm to solve ℓ-IsogenyPath.

Proposition 1.6. *The* ℓ-IsogenyPath *problem can be solved in expected time* $(\ell + \log p)^{O(1)} \cdot p^{1/2}$.

Sketch of the proof. This can be done by a straigtforward meet-in-the-middle algorithm. Let n as in Proposition 1.4 such that the output of a random walk of length n is indistinguishable from uniform. Then, generate many random walks $\varphi_i : E_1 \to E_i'$ until one has found $p^{1/2}$ distinct targets E_i', and store them. Now, generate random walks $\psi : E_2 \to E$ until the random target E is isomorphic to one of the curves E_i' (this happens with probability $\Omega(p^{-1/2})$, so requires an expected number of trials $O(p^{1/2})$). For this i, return $\hat{\psi} \circ \varphi_i$. □

1.2.4. Higher level isogeny graphs. It is often useful to attach some additional data to elliptic curves. For instance, fix an integer $N > 0$, and consider pairs (E, C) where $C \subset E[N]$ is a cyclic subgroup of order N. We can then consider isogenies that preserve this structure: an isogeny $(E_1, C_1) \to (E_2, C_2)$ is an isogeny $\varphi : E_1 \to E_2$ such that $\varphi(C_1) = C_2$. This naturally leads to a generalization of isogeny graphs, where vertices are pairs (E, C) and edges are such "structure preserving" isogenies.

This particular example is the isogeny graph with *level N Borel structure*. They are the first example of higher level isogeny graphs that appeared in isogeny-based cryptography, in ⟦BCC+23⟧. In that article, we prove that random walks in these graphs equidistribute optimally, leading to the first general purpose statistically zero-knowledge proof of isogeny knowledge (a cryptographic protocol to prove that one knows a solution to ℓ-IsogenyPath without revealing it). A key observation is that a vertex (E, C) of this graph can be interpreted as an N-isogeny $E \to E/C$, hence random walks in this graph can be used to equidistribute not just curves, but isogenies.

Other types of additional data can be attached to elliptic curves, and the corresponding graphs and random walks have proved to be powerful tools. The most general framework to date is introduced in our article ⟦PW24⟧. In the rest of this section, we present this

framework and the corresponding equidistribution result. For the highest level of generality, we resort to the language of categories. The first category of interest is $\mathrm{SS}_\Sigma(p)$, underlying the standard isogeny graph.

Definition 1.7. Let Σ be a set of prime numbers. The category $\mathrm{SS}_\Sigma(p)$ has

- objects: supersingular elliptic curves over $\overline{\mathbf{F}}_p$;
- morphisms $\mathrm{Hom}_\Sigma(E, E')$: isogenies of degree a product of the primes in Σ.

The next step consists in attaching extra structure to objects of this category. This is done through the construction of a "category of elements".

Definition 1.8. Let $\mathcal{C}$ be a category and $\mathcal{F}\colon \mathcal{C} \to \mathrm{Sets}$ be a functor. The *category of elements* $\mathrm{El}(\mathcal{F})$ is the category with

- objects: pairs (c, x) where $c \in \mathcal{C}$ and $x \in \mathcal{F}(c)$;
- morphisms $(c, x) \to (c', x')$: morphisms $f \in \mathrm{Hom}_{\mathcal{C}}(c, c')$ s.t. $\mathcal{F}(f)(x) = x'$.

Example 1.9. Assume $p \nmid N$. Let Σ be the set of primes not dividing N. Define the functor $\mathrm{Cyc}_N\colon \mathrm{SS}_\Sigma(p) \to \mathrm{Sets}$ by:

- $\mathrm{Cyc}_N(E)$ is the set of cyclic subgroups of order N of E;
- for every isogeny $\varphi \in \mathrm{Hom}_\Sigma(E, E')$, the map $\mathrm{Cyc}_N(\varphi)$ is $C \mapsto \varphi(C)$.

Then $\mathrm{El}(\mathrm{Cyc}_N)$ is the category of pairs (E, C) where E is a supersingular elliptic curves and C is a cyclic subgroup of order N — the so-called level N Borel structure used in ⟦BCC+23⟧.

Example 1.10. Let Σ be the set of primes not dividing N. Let End/N denote the functor $\mathrm{SS}_\Sigma(p) \to \mathrm{Sets}$ defined by

- $(\mathrm{End}/N)(E) = \mathrm{End}(E)/N\,\mathrm{End}(E)$;
- for $\varphi\colon E \to E'$, the map $(\mathrm{End}/N)(\varphi)$ is $\alpha \mapsto \varphi\alpha\hat{\varphi}$.

Then $\mathrm{El}(\mathrm{End}/N)$ is is the category of supersingular elliptic curves equipped with an endomorphism modulo N. This example plays an important role in the equivalence between the problems EndRing and OneEnd presented in Section 1.4.

We now introduce the graphs of interest.

Definition 1.11. Let $\mathcal{F}\colon \mathrm{SS}_\Sigma(p) \to \mathrm{Sets}$ be a functor with $\mathcal{F}(E)$ finite for all E. Let $\ell \in \Sigma$ be a prime different from p. We define the graph $\mathcal{G}^\ell_{\mathcal{F}}$ with:

- vertices: isomorphism classes of objects in $\mathrm{El}(\mathcal{F})$;
- edges: let $(E, x) \in \mathrm{El}(\mathcal{F})$; edges from (E, x) are isogenies $\varphi \in \mathrm{Hom}_\Sigma(E, E')$ of degree ℓ, modulo automorphisms of $(E', \mathcal{F}(\varphi)(x))$.

Returning to Example 1.9, if ℓ is a prime not dividing Np, the graph $\mathcal{G}^\ell_{\mathrm{Cyc}_N}$ is the ℓ-isogeny graph of supersingular elliptic curves with Borel structure studied in [Arp23] and ⟦BCC+23⟧. When $N = 1$ this is the standard supersingular ℓ-isogeny graph.

This general construction of graph $\mathcal{G}^\ell_{\mathcal{F}}$ is perhaps too general. For random walks to behave well, one needs the functor $\mathcal{F}$ to preserve some of the underlying arithmetic structure. We require it to satisfy the $(\mathrm{mod}\ N)$-*congruence property*.

Definition 1.12. Let $\mathcal{F}\colon \mathrm{SS}_\Sigma(p) \to \mathrm{Sets}$ be a functor and $N \geq 1$ an integer. We say that $\mathcal{F}$ *satisfies the* $(\mathrm{mod}\ N)$-*congruence property* if for every $E \in \mathrm{SS}(p)$ and every $\varphi, \psi \in \mathrm{End}_\Sigma(E)$ such that $\varphi - \psi \in N\,\mathrm{End}(E)$, we have $\mathcal{F}(\varphi) = \mathcal{F}(\psi)$.

The functors Cyc_N and End/N from Examples 1.9 and 1.10 above do satisfy the $(\mathrm{mod}\ N)$-congruence property.

Stated informally, the equidistribution theorem we prove in ⟦PW24⟧ is the following.

Theorem 1.13 (Informal formulation of ⟦PW24, Theorem 3.10⟧)**.** *If $\mathcal{F}$ satisfies the* (mod N)*-congruence property, then random walks in $\mathcal{G}_{\mathcal{F}}^{\ell}$ equidistribute optimally.*

The optimality refers to the fact that the graphs can be disconnected or multipartite (clear obstructions to equidistribution) but the equidistribution is "as good as possible" under these constraints.

Independently, a similar result was proved by Codogni and Lido [CL23], in the case where the extra data is of a particular kind: a *level structure*, expressed in terms of N-torsion points. Example 1.9 fits in that framework, but Example 1.10 does not. Both proofs use Deligne's bounds on coefficients of modular forms, but the proof in [CL23] is purely algebro-geometric, whereas ours proceeds via the Deuring correspondence and the Jacquet–Langlands correspondence; as a result, the two proofs could have different interesting generalisations.

1.3. The ℓ-Isogeny Path and Endomorphism Ring problems

Recall that given an elliptic curve E, an endomorphism is an isogeny from E to itself (or the zero morphism), and the set of all endomorphisms of E, written $\mathrm{End}(E)$, is a ring. Loops in ℓ-isogeny graphs provide endomorphisms, hinting at a connection between path-finding problems and computing endomorphism rings. This connection is a fundamental aspect of isogeny-based cryptography.

The endomorphism ring is always a lattice; and in the supersingular case, it is a lattice of rank 4. The *endomorphism ring problem* consists in finding a basis of this lattice.

Problem 1.14 (EndRing)**.** Given a prime p, and a supersingular elliptic curve E over $\mathbf{F}_{p^2}$, find four endomorphisms of E (in an efficient representation) that generate $\mathrm{End}(E)$ as a lattice.

The EndRing problem, and its connection to ℓ-IsogenyPath, has been studied as early as [Koh96], before any cryptographic motivation. With the increasing practical relevance of these problems, it has become critical to understand their relation. It was soon suspected that they were equivalent, and heuristic reductions were described in [EHL+18]. The following theorem is the main result from our article ⟦Wes21⟧.

Theorem 1.15 (⟦Wes21⟧)**.** *The problems* ℓ-IsogenyPath *and* EndRing *are equivalent under probabilistic polynomial time reductions, assuming the Generalized Riemann Hypothesis.*

This section gives an overview of its proof. One of the main ingredients is an important piece of the underlying theory which will play a role in the rest of this chapter: the Deuring correspondence.

1.3.1. The Deuring correspondence. The *Deuring correspondence* states that the map $E \mapsto \mathrm{End}(E)$ is essentially a bijection (even an equivalence of categories) between supersingular elliptic curves and a certain family of rings: maximal orders in a quaternion algebra $B_{p,\infty}$.

First, let us define more precisely this family of rings. The *quaternion algebra* $B_{p,\infty}$ is defined as the algebra of dimension 4 over $\mathbf{Q}$ with basis $1, i, j, k$ satisfying the multiplication rules $i^2 = -q$, $j^2 = -p$, and $k = ij = -ji$, with

$$q = \begin{cases} 1 \text{ if } p \equiv 3 \bmod 4, \text{or} \\ 2 \text{ if } p \equiv 5 \bmod 8, \text{and otherwise} \\ \text{the smallest prime such that } q \equiv 3 \bmod 4 \text{ and } \left(\frac{p}{q}\right) = -1. \end{cases}$$

An *order* in $B_{p,\infty}$ is a discrete subring containing a basis of the algebra (i.e., a subring that is also a lattice of rank 4). It is *maximal* if not contained in any other order.

Given a supersingular elliptic curve E over $\mathbf{F}_p$, the endomorphism algebra $\mathrm{End}(E)\otimes_{\mathbf{Z}}\mathbf{Q}$ is isomorphic to $B_{p,\infty}$. Through this isomorphism, the endomorphism ring $\mathrm{End}(E)$ is a maximal order in the algebra. This so-called *Deuring correspondence* provides a bijection between the isomorphism classes

$$\left\{\begin{array}{c}\text{Isomorphism classes of}\\ \text{supersingular elliptic curves } E \text{ over } \mathbf{F}_{p^2}\end{array}\right\} / \mathrm{Gal}(\mathbf{F}_{p^2}/\mathbf{F}_p) \longleftrightarrow \left\{\begin{array}{c}\text{Isomorphism classes of}\\ \text{maximal orders } \mathcal{O} \text{ in } B_{p,\infty}\end{array}\right\}.$$

The MaxOrder *problem.* The MaxOrder problem is the computational incarnation of this bijection.

Problem 1.16 (MaxOrder)**.** Given a prime p, and a supersingular elliptic curve E over $\mathbf{F}_{p^2}$, find four quaternions in $B_{p,\infty}$ that generate a maximal order $\mathcal{O}$ such that $\mathcal{O} \cong \mathrm{End}(E)$.

This problem is similar to EndRing: it asks to compute the endomorphism ring of a curve. But while EndRing asks for actual endomorphisms that generate $\mathrm{End}(E)$, the MaxOrder problem only asks for the ring structure of $\mathrm{End}(E)$, up to isomorphism.

The MaxOrder problem was already identified in [EHL+18] as lying at the heart of the equivalence between EndRing and ℓ-IsogenyPath. Indeed, following in the footsteps of [EHL+18], we prove in ⟦Wes21⟧ that EndRing and ℓ-IsogenyPath are equivalent by proving that each is equivalent to MaxOrder.

The correspondence between morphisms. As already mentioned, this correspondence extends to an equivalence of categories. Morphisms on the side of elliptic curves are isogenies and the zero morphism. Morphisms on the side of orders are ideals: any left ideal I in a maximal order $\mathcal{O}_1$ is a right ideal in another order $\mathcal{O}_2$. The zero ideal is in correspondence with the zero morphism of curves. When I is non-zero, we call it a *connecting ideal* between $\mathcal{O}_1$ and $\mathcal{O}_2$, and we think of it as a morphism from $\mathcal{O}_1$ to $\mathcal{O}_2$. For such an ideal I, we call $\mathcal{O}_L(I) = \mathcal{O}_1$ its *left order*, and $\mathcal{O}_R(I) = \mathcal{O}_2$ its *right order*. A (non-zero) morphism from $\mathcal{O}_1$ to $\mathcal{O}_2$ is a (non-zero) left ideal I in $\mathcal{O}_1$ such that $\mathcal{O}_R(I) \cong \mathcal{O}_2$. The correspondence between morphisms is defined as follows:

- An isogeny $\varphi : E_1 \to E_2$, corresponds to the ideal $I_\varphi = \mathrm{Hom}(E_2, E_1) \circ \varphi \subseteq \mathrm{End}(E_1)$. It is a left ideal in $\mathrm{End}(E_1)$, and a right ideal in $\mathcal{O}_R(I) \cong \mathrm{End}(E_2)$.
- A left ideal $I \subseteq \mathrm{End}(E)$ corresponds to the isogeny $\varphi_I = E \to E/E[I]$ where $E[I] = \bigcap_{\alpha\in I} \ker \alpha$ (at least in the separable case, when the norm of I is coprime to p).

For any φ and I, we have $I_{\varphi_I} = I$ and $\varphi_{I_\varphi} = \varphi$ (up to an isomorphism of the target). Furthermore, the degree of a non-zero isogeny matches the *reduced norm* $\mathrm{Nrd}(I) = \sqrt{[\mathcal{O}_L(I) : I]}$ of the corresponding ideal I.

Remark 1.17. Aspects of this correspondence are better captured when replacing the category of maximal orders with the category of invertible right $\mathcal{O}$-modules for some reference maximal order $\mathcal{O}$. See for instance [Voi21, Theorem 42.3.2]. That point of view avoids issues raised by working *up to isomorphism* as above.

As MaxOrder asks to compute the map between objects $E \mapsto \mathrm{End}(E)$, one can ask the analogous question for morphisms: can the maps $\varphi \mapsto I_\varphi$ and $I \mapsto \varphi_I$ be computed efficiently? The first positive answer to this question is the following lemma: we can translate between isogenies and ideals if we already know the endomorphism ring of the source curve.

Lemma 1.18. *Assuming the Generalized Riemann Hypothesis, there is an algorithm such that the following holds. Let E_0 be a supersingular elliptic curve, $\mathcal{O}_0$ an order in $B_{p,\infty}$, and $\iota : \mathcal{O}_0 \longrightarrow \mathrm{End}(E_0)$ a bijection (computationally represented by a basis of $\mathcal{O}_0$ and an efficient representations of its image). Then,*

- *on input ι and an isogeny $\varphi : E_0 \to E_1$ (in isogeny path representation), the algorithm returns the left $\mathcal{O}_0$-ideal I_φ,*
- *on input ι and a left $\mathcal{O}_0$-ideal I, returns the isogeny φ_I (in isogeny path representation).*

The algorithm runs in polynomial time in the length of the input, and in the largest prime factor of the degree of φ and norm of I.

Sketch of the proof. The technique finds its roots in the heuristic result [GPS20, Lemma 6]. It was revisited in [EHL+18], and the first rigorous algorithm is given in ⟦Wes21⟧.

The idea is to first deal with the case where the degree (or norm) is powersmooth: all its prime-power factors are small. One can then deal with each prime-power factor independently, and reconstitute the output via the following fact: if I and J are left $\mathcal{O}_0$-ideals of coprime norm, we have $\ker \varphi_{I\cap J} = \ker \varphi_I + \ker \varphi_J$. Translating between ideals and isogenies for a small prime-power factor ℓ^e is rather straightforward, as there are $O(\ell^e)$ possible kernels or ideals, and a correct guess can be validated efficiently.

Dealing with large prime powers is much trickier, and requires reducing the problem to the powersmooth case. The idea consists in chopping the long isogeny path (or ideal) into smaller chunks, and iteratively find powersmooth alternative paths to each intermediate step. There lies the technical heart of ⟦Wes21⟧: an algorithm which, on input a left $\mathcal{O}_0$-ideal, finds an equivalent ideal whose norm is powersmooth (two left $\mathcal{O}_0$-ideals are equivalent if their right orders are isomorphic). A heuristic algorithm for this task was first presented in [KLPT14], known as the KLPT algorithm. We prove in ⟦Wes21, Theorem 6.4⟧ that the same task can be performed assuming only GRH. □

1.3.2. The quaternion analog of the ℓ-IsogenyPath problem. Through the Deuring correspondence, a computational problem involving supersingular elliptic curves can be translated into a computational problem involving quaternions. For instance, as the ℓ-IsogenyPath problem asks one to find an isogeny of degree a power of ℓ between two curves, its quaternionic analog is: given two (isomorphism classes of) maximal orders $\mathcal{O}_1$ and $\mathcal{O}_2$, find an ideal of norm a power of ℓ connecting them.

While ℓ-IsogenyPath is believed to be hard, this quaternion version turns out to be easy. The first heuristic resolution of this problem is the KLPT algorithm [KLPT14], already mentioned in the proof of Lemma 1.18. In the article ⟦Wes21⟧, we prove that it can be solved in polynomial time assuming GRH.

Theorem 1.19 (Special case of ⟦Wes21, Theorem 6.3⟧)**.** *There is an algorithm which given two maximal orders $\mathcal{O}_1$ and $\mathcal{O}_2$ in $B_{p,\infty}$ and a prime ℓ, finds a left $\mathcal{O}_1$-ideal I of norm a power of ℓ such that $\mathcal{O}_R(I) \cong \mathcal{O}_2$, and runs in expected polynomial time in ℓ and in the size of the input, assuming the Generalized Riemann Hypothesis.*

The reader may already see how this could help reducing ℓ-IsogenyPath to MaxOrder: an oracle for MaxOrder can turn an instance of ℓ-IsogenyPath into an instance of its quaternionic analog, which is easy. We would then need to translate the quaternionic solution back to an isogeny, perhaps with Lemma 1.18, but that requires an additional ingredient.

1.3.3. Finding a reference supersingular elliptic curve. The translation between ideals and isogenies with Lemma 1.18 plays a key role in proving that the three problems ℓ-IsogenyPath, MaxOrder and EndRing are equivalent. Yet, applying the lemma requires prior knowledge of the endomorphism ring of one of the curves involved. We are thus interested in generating a "reference" curve, a special supersingular elliptic curve E_0 for which we know, by construction, a solution to both MaxOrder and EndRing. Then, Lemma 1.18 can be used to connect instances of ℓ-IsogenyPath, MaxOrder or EndRing to that reference curve E_0.

Such "special" elliptic curves can be found by turning to the usual suspects, such as the elliptic curves of j-invariant 0 or 1728. Which curve to pick depends on the congruence class of $p \bmod 8$, but the general idea is always the same: find an elliptic curve over $\mathbf{Q}$ which has supersingular reduction modulo p, and has complex multiplication by an order of small discriminant. Then, the Frobenius endomorphism π_p together with the reduction of the complex multiplication already reveals a large part of the endomorphism ring.

Lemma 1.20 ([EHL+18, Proposition 3])**.** *There is an algorithm that for any prime $p > 2$ computes an elliptic curve E_0 over $\mathbf{F}_p$, a basis $(1, \alpha_1, \alpha_2, \alpha_3)$ of a maximal order $\mathcal{O}_0$ of $B_{p,\infty}$, and three endomorphisms $\beta_1, \beta_2, \beta_3$ of E_0 (in efficient representation), such that*

$$\mathcal{O}_0 \longrightarrow \mathrm{End}(E_0) : 1, \alpha_1, \alpha_2, \alpha_3 \longmapsto [1], \beta_1, \beta_2, \beta_3$$

is an isomorphism, and runs in time polynomial in $\log p$ (if $p \equiv 1 \bmod 8$, we assume GRH).

Sketch of the proof. If $p \equiv 3 \bmod 4$, then the curve E_0 defined by $y^2 = x^3 - x$ is supersingular. It is defined over $\mathbf{F}_p$, so has the Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$. Furthermore, if $\alpha \in \mathbf{F}_{p^2}$ satisfies $\alpha^2 = -1$, it is easy to check that $\iota : (x, y) \mapsto (-x, \alpha y)$ is also an endomorphism. These endomorphisms generate almost all $\mathrm{End}(E_0)$: we actually have

$$\mathrm{End}(E_0) = \mathbf{Z} \oplus \mathbf{Z}\iota \oplus \mathbf{Z}\frac{\iota + \iota\pi}{2} \oplus \mathbf{Z}\frac{1 + \pi}{2}.$$

Therefore we can consider $(\beta_1, \beta_2, \beta_3) = (\iota, \frac{\iota+\iota\pi}{2}, \frac{1+\pi}{2})$. Since $\iota^2 = [-1]$ and $\pi^2 = [-p]$, we can set the corresponding quaternions $(\alpha_1, \alpha_2, \alpha_3) = (i, \frac{i+ij}{2}, \frac{1+j}{2})$. The case $p \equiv 5 \bmod 8$ enjoys similar explicit formulae, and the case $p \equiv 1 \bmod 8$ requires to first find a small quadratic non-residue q modulo p (hence the need for GRH), then generate an elliptic curve over $\mathbf{Q}$ with complex multiplication by $\sqrt{-q}$ (hence the need for q to be small). □

1.3.4. ℓ-IsogenyPath **is equivalent to** MaxOrder**.** We now have all the ingredients to sketch the proof that ℓ-IsogenyPath is equivalent to MaxOrder.

MaxOrder *reduces to* ℓ-IsogenyPath. Suppose we wish to solve MaxOrder for some elliptic curve E, and we have access to an oracle for ℓ-IsogenyPath.

From Lemma 1.20, we can find a supersingular curve E_0 together with a maximal order $\mathcal{O}_0$ and an isomorphism $\mathcal{O}_0 \longrightarrow \mathrm{End}(E_0)$. Using the oracle for ℓ-IsogenyPath, one can find an isogeny path $\varphi : E_0 \to E$. It remains to "transport" the information from E_0 along the path to E. This is where Lemma 1.18 comes in: one can compute the ideal I_φ, then return the right-order of I_φ, which is isomorphic to $\mathrm{End}(E)$.

ℓ-IsogenyPath *reduces to* MaxOrder. Suppose we wish to solve ℓ-IsogenyPath for two elliptic curve E_1 and E_2, and we have access to an oracle for MaxOrder.

It is enough to show how to find an ℓ-isogeny path $E_0 \to E_1$ for the special curve E_0 from Lemma 1.20. Indeed, the same can be applied to $E_0 \to E_2$, and the composition gives a path $E_1 \to E_2$. Now, the oracle for MaxOrder reveals an order $\mathcal{O}_1$ isomorphic to $\mathrm{End}(E_1)$. There is an efficient algorithm to find a connecting ideal between $\mathcal{O}_0$ and $\mathcal{O}_1$ [KV10, Algorithm 3.5]. One can then find an equivalent ideal of norm a power of ℓ (thanks to Lemma 1.19), and use Lemma 1.18 to translate this ideal to the corresponding isogeny: a ℓ-isogeny path $E_0 \to E_1$.

1.3.5. MaxOrder **is equivalent to** EndRing**.** Finally, we sketch the proof that the problems MaxOrder and EndRing are equivalent.

MaxOrder *reduces to* EndRing. The reduction from MaxOrder to EndRing is perhaps the least surprising direction. On one hand, the endomorphism ring $\mathrm{End}(E)$ is a Euclidean lattice with the positive definite integral quadratic form $\deg : \mathrm{End}(E) \to \mathbf{Z}$. On the other hand, the algebra $B_{p,\infty}$ is a Euclidean space with the reduced norm $\mathrm{Nrd} : B_{p,\infty} \to \mathbf{Z}$. A ring embedding $\mathrm{End}(E) \to B_{p,\infty}$ preserves the Euclidean structure. The idea is thus to find a subring in $B_{p,\infty}$ that has the same "geometry" as $\mathrm{End}(E)$, and deduce that they are isomorphic as rings.

Concretely, from a basis $(\beta_i)_{i=1}^4$ of the lattice $\mathrm{End}(E)$, one can compute the Gram matrix $(\langle \beta_i, \beta_j \rangle)_{i,j=1}^4$, then find a basis of $B_{p,\infty}$ with the same Gram matrix. This amounts to solving a few homogeneous quadratic equations over $\mathbf{Q}$, which can be done with [`Sim05`, `Sim06`].

EndRing *reduces to* MaxOrder. The reduction from EndRing to MaxOrder may appear more difficult, as solving EndRing requires finding actual endomorphisms of E, while solving MaxOrder only seems to provide abstract "quaternionic" information on the endomorphism ring. The idea is to relate E to the reference curve E_0 for which we know both actual endomorphisms and an embedding in the quaternion world. The reduction works as follows.

(1) Generate E_0 together with an isomorphism $\iota : \mathcal{O}_0 \to \mathrm{End}(E_0)$ (Lemma 1.20).
(2) Solve MaxOrder to find an order $\mathcal{O}$ in $B_{p,\infty}$ isomorphic to $\mathrm{End}(E)$.
(3) Find a connecting ideal I between $\mathcal{O}_0$ and $\mathcal{O}$ ([`KV10`, Algorithm 3.5]).
(4) Find the corresponding isogeny $\varphi_I : E_0 \to E$ (Lemma 1.18).
(5) We obtain an isomorphism of algebras

$$\jmath : \mathrm{End}(E_0) \otimes \mathbf{Q} \longrightarrow \mathrm{End}(E) \otimes \mathbf{Q} : \beta \longmapsto (\varphi_I \circ \beta \circ \hat{\varphi}_I) \otimes \frac{1}{\deg(\varphi_I)}. \tag{1.1}$$

(6) The composition $\jmath \circ (\iota \otimes \mathbf{Q}) : B_{p,\infty} \to \mathrm{End}(E) \otimes \mathbf{Q}$ is also an isomorphism.
(7) Output the basis of $\jmath \circ (\iota \otimes \mathbf{Q})(\mathcal{O}) = \mathrm{End}(E)$.

Of course, one should ensure that the output basis consists of endomorphisms *in efficient representation*. There a several ways to do this. The main obstacle is the division by $\deg(\varphi_I)$ in the map (1.1). The originally proposed solution is to ensure that $\deg(\varphi_I)$ is powersmooth, so there is an efficient algorithm to divide points. Alternatively, a more recent algorithm allows to divide any endomorphism in efficient representation by any integer [[`HW23`, Theorem 4.1]] (see Section 2.4.1).

1.4. Finding one endomorphism is as hard as finding them all

While the endomorphism ring problem asks to find, in a sense, all the endomorphisms of a supersingular curve, it has appeared hard to find even a single one. Scalar multiplications $[m]$ for $m \in \mathbf{Z}$ are trivial to find, so we exclude them. This is the (supersingular) *one endomorphism* problem.

Problem 1.21 (OneEnd)**.** Given a prime p and a supersingular elliptic curve E over $\mathbf{F}_{p^2}$, find an endomorphism in $\mathrm{End}(E) \setminus \mathbf{Z}$ in efficient representation.

The connection between EndRing and OneEnd bears important consequences on the hardness of EndRing and on its connection with variants of the isogeny problem. On the cryptographic side, the OneEnd problem naturally emerges when analyzing the security of certain schemes. Most notably, it is easy to prove that some version of the CGL hash function is *collision-resistant* if OneEnd is hard. This result alone is unsatisfactory, as OneEnd seems, at first glance, simpler that the pinnacle problem of the field, EndRing. Unfortunately, former heuristic arguments suggesting that OneEnd should be as hard as

EndRing do not withstand close scrutiny, and actually fail in simple cases.

In the article ⟦PW24⟧, we prove the following theorem.

Theorem 1.22 (⟦PW24, Theorem 1.1⟧)**.** *The* EndRing *and* OneEnd *problems are equivalent, under probabilistic polynomial time reductions.*

In Section 1.4.1 below, we present the proof strategy. This theorem was our motivation to prove the general rapid-mixing in higher level isogeny graphs, Theorem 1.13. In Section 1.4.2, we present a few important consequences of this theorem: the collision-resistance of the CGL hash function, the equivalence between EndRing and the "pure" Isogeny problem, and the fastest known unconditional algorithm to solve EndRing.

1.4.1. Proving that EndRing **reduces to** OneEnd**.** The ideas behind our reduction are as follows. Assume we have an oracle $\mathscr{O}$ for OneEnd and we want to compute $\mathrm{End}(E)$ for a given E.

The ring $\mathrm{End}(E)$ is a lattice of dimension 4 and volume $p/4$ (with respect to the quadratic form $\deg : \mathrm{End}(E) \to \mathbf{Z}$). A solution of EndRing consists in four endomorphisms that generate all the others. Given a collection of endomorphisms, one can decide whether they generate the whole ring $\mathrm{End}(E)$ by computing the volume of the lattice they generate, and comparing it to $p/4$. Once a generating set has been found, it is easy to deduce a basis.

A first flawed attempt. We thus need a way to generate several endomorphisms of E. Naively, one could repeatedly call $\mathscr{O}(E)$, hoping to eventually obtain a generating set. This can fail, for instance if the oracle is deterministic and $\mathscr{O}(E)$ always returns the same endomorphism.

To circumvent this issue, it was proposed in [EHL+18] to randomise the curve. More precisely, one constructs a richer, randomised oracle $\textsc{Rich}^{\mathscr{O}}$ from $\mathscr{O}$ as follows. On input E, walk randomly on the 2-isogeny graph, resulting in an isogeny $\varphi\colon E \to E'$. This graph has rapid mixing properties, so E' is close to uniformly distributed among supersingular curves. Now, call the oracle $\mathscr{O}$ on E', to get an endomorphism $\beta \in \mathrm{End}(E')$. The composition $\alpha = \hat{\varphi} \circ \beta \circ \varphi$ is an endomorphism of E, the output of $\textsc{Rich}^{\mathscr{O}}$.

With this randomisation, there is hope that calling $\textsc{Rich}^{\mathscr{O}}$ repeatedly on E could yield several independent endomorphisms that would eventually generate $\mathrm{End}(E)$. This method is essentially what is proposed in [EHL+18, Algorithm 8]. In that article, it is heuristically assumed that endomorphisms produced by $\textsc{Rich}^{\mathscr{O}}$ are very nicely distributed, and they deduce that a generating set for $\mathrm{End}(E)$ is rapidly obtained. This heuristic has a critical flaw: one can construct oracles that contradict it. Consider an integer $M > 1$, and suppose that for any input E, the oracle $\mathscr{O}$ returns an endomorphism from the strict subring $\mathbf{Z} + M\,\mathrm{End}(E)$. Then, the above algorithm would fail, because the randomisation $\textsc{Rich}^{\mathscr{O}}$ would still be stuck within the subring $\mathbf{Z} + M\,\mathrm{End}(E)$. Worse, juggling with several related integers M, we will see that there are oracles for which this algorithm only stabilises after an exponential time.

Invariance by conjugation. The core of our method rests on the idea that this issue is, in essence, the only possible obstruction. The key is *invariance by conjugation*. If $\varphi, \varphi' : E \to E'$ are two random walks of the same length, and β is an endomorphism of $\mathrm{End}(E')$, the elements $\alpha = \hat{\varphi} \circ \beta \circ \varphi$ and $\alpha' = \hat{\varphi}' \circ \beta \circ \varphi'$ are equally likely outputs of $\textsc{Rich}^{\mathscr{O}}$. These two elements are conjugates of each other in $\mathrm{End}(E)/N\,\mathrm{End}(E)$ for any odd integer N, as

$$\alpha = \frac{\hat{\varphi} \circ \varphi'}{[\deg(\varphi')]} \circ \alpha' \circ \frac{\hat{\varphi}' \circ \varphi}{[\deg(\varphi')]} \bmod N.$$

From there, one can prove that the output of $\textsc{Rich}^{\mathscr{O}}$ follows a distribution that is invariant by conjugation: each output is as likely as any of its conjugates, modulo odd integers N (up to some bound). It is a consequence of Theorem 1.13, our general equidistribution result.

Intuitively, for the outputs of $\textsc{Rich}^{\mathscr{O}}$ to be "stuck" in a subring (such as $\mathbf{Z}+M\operatorname{End}(E)$ above), that subring must itself be stable by conjugation (modulo odd integers N). There comes the next key: every subring of $\operatorname{End}(E)$ (of finite index not divisible by p) stable by conjugation modulo all integers is of the form $\mathbf{Z}+M\operatorname{End}(E)$. So samples from $\textsc{Rich}^{\mathscr{O}}$ must *eventually* generate a ring of the form $\mathbf{Z}+M\operatorname{End}(E)$. From a basis of $\mathbf{Z}+M\operatorname{End}(E)$, it is easy to recover a basis of $\operatorname{End}(E)$ essentially by dividing by M (using a method developed in [Rob22] and ⟦HW23, Theorem 4.1⟧).

Local obstructions. This intuition does not immediately translate into an algorithm, as an oracle could be "bad" without really being stuck in a subring. Imagine an oracle that outputs an element of $\mathbf{Z}+2^e\operatorname{End}(E)$ (and not in $\mathbf{Z}+2^{e+1}\operatorname{End}(E)$) with probability 2^{e-n} for each $e\in[0,\dots,n-1]$. A sequence of samples $(\alpha_i)_i$ would eventually generate $\operatorname{End}(E)$, but only after an amount of time exponential in n. This particular case could be resolved as follows: for each sample α, identify the largest e such that $\beta=(2\alpha-\operatorname{Tr}(\alpha))/2^e$ is an endomorphism. A sequence of samples $(\beta_i)_i$ would rapidly generate $\mathbf{Z}+2\operatorname{End}(E)$, from which one easily recovers $\operatorname{End}(E)$. This resolution first identifies the prime 2 as the source of the obstruction, then "reduces" each sample "at 2". In general, such obstructive primes would appear as factors of $\operatorname{disc}(\alpha)$. Identifying these primes, and ensuring that each sample is "reduced" at each of them, one gets, in principle, a complete algorithm. However, factoring $\operatorname{disc}(\alpha)$ could be hard. Instead, we implement an optimistic approach: we identify obstructive pseudo-primes using a polynomial time partial-factoring algorithm. The factors may still be composite, but it is fine: the algorithm will either behave as if they were prime, or reveal a new factor.

1.4.2. Consequences. We now discuss some consequences of Theorem 1.22.

1.4.2.1. *Collision resistance of the Charles–Goren–Lauter hash function.* The first cryptographic construction based on the supersingular isogeny problem is the CGL hash function [CLG09]

$$\mathrm{CGL}_{E_0}:\{0,\dots,\ell-1\}^*\longrightarrow\mathbf{F}_{p^2},$$

introduced in Section 1.2.3. Recall that any binary string $x\in\{0,\dots,\ell-1\}^*$ encodes a non-backtracking random walk $\varphi_x:E_0\to E_x$ from a source E_0 in the 2-isogeny graph, and $\mathrm{CGL}_{E_0}(x)=j(E_x)$ is the j-invariant of the target of this walk. As discussed in Section 1.2.3, using the rapid equidistribution of random walks, it is rather straightforward to prove that if ℓ-$\textsc{IsogenyPath}$ is hard, then CGL_{E_0} is preimage-resistant.

Its resistance to collisions is considerably more delicate. A collision for CGL_{E_0} is a pair of distinct inputs $x,x'\in\{0,\dots,\ell-1\}^*$ such that $\mathrm{CGL}_{E_0}(x)=\mathrm{CGL}_{E_0}(x')$. In other words, a collision consists in two distinct paths $\varphi_x,\varphi_{x'}:E_0\to E'$ to the same target E' (up to isomorphism). The composition $\hat{\varphi}_{x'}\circ\varphi_x$ is a non-scalar endomorphism of E_0. At first glance, this is good news: if $\textsc{OneEnd}$ is hard for E_0, then finding collisions is hard.

This reasoning also hints at a weakness: one who already knows $\operatorname{End}(E_0)$ could perhaps find collisions in polynomial time. This is indeed the case [EHL+18]. To prove collision-resistance from the hardness of $\textsc{EndRing}$, one must consider a family of hash functions CGL_E where E is random. Let $\textsc{SampleSS}(p)$ be an algorithm sampling a uniformly random supersingular elliptic curve over $\mathbf{F}_{p^2}$. We define the *advantage* of a collision-finding algorithm $\mathscr{A}$ for the CGL family of hash functions as

$$\mathrm{Adv}^{\mathscr{A}}_{\mathrm{CGL}}(p)=\Pr\left[\begin{array}{c|c}m\neq m'\text{ and}\\ \mathrm{CGL}_E(m)=\mathrm{CGL}_E(m')\end{array}\;\middle|\;\begin{array}{c}E\leftarrow\textsc{SampleSS}(p)\\ (m,m')\leftarrow\mathscr{A}(E)\end{array}\right].$$

The hash function is *collision-resistant* if all efficient algorithms $\mathscr{A}$ have negligibly small advantage. It was heuristically argued in [EHL+18] that the collision-resistance of this construction is equivalent to EndRing. Theorem 1.22 unlocks the proof: in [[PW24]], we prove that a collision-finding algorithm with good running time and advantage can solve EndRing efficiently.

Theorem 1.23 (Collision-resistance of the CGL hash function, [[PW24, Theorem 8.1]])**.** *For any algorithm $\mathscr{A}$, there is an algorithm to solve* EndRing *in expected polynomial time in* $\log(p)$*, in* $\mathrm{Adv}^{\mathscr{A}}_{\mathrm{CGL}}(p)^{-1}$ *and in the expected running time of* $\mathscr{A}$*.*

Sketch of the proof. Since EndRing is equivalent to OneEnd (Theorem 1.22), it is sufficient to prove that $\mathscr{A}$ can be used to solve OneEnd. First, we observe that a successful collision for CGL_E gives a non-scalar endomorphism of E: if $\varphi, \psi\colon E \to E'$ are two distinct non-backtracking walks, then $\hat{\varphi} \circ \psi \in \mathrm{End}(E) \setminus \mathbf{Z}$.

Therefore, the algorithm $\mathscr{A}$ finding a collision for $\mathrm{CGL}_{E'}$ (with good probability for uniformly random E') in fact solves OneEnd (with good probability for uniformly random E'). To solve OneEnd on an arbitrary (non-random) input E, one first computes a random walk $\varphi : E \to E'$ with uniformly random target E', calls the oracle to find $\alpha \in \mathrm{End}(E') \setminus \mathbf{Z}$, and returns $\hat{\varphi} \circ \alpha \circ \varphi \in \mathrm{End}(E) \setminus \mathbf{Z}$. □

1.4.2.2. *The endomorphism ring problem is equivalent to the isogeny problem.* We have already established that the problem EndRing is equivalent to ℓ-IsogenyPath (assuming the Generalised Riemann Hypothesis, Theorem 1.15). The latter problem asks to find isogenies of a very specific form: ℓ-isogeny paths. Lifting this restriction yields the more general Isogeny problem.

Problem 1.24 (Isogeny)**.** Given a prime p and two supersingular elliptic curves E and E' over $\mathbf{F}_{p^2}$, find an isogeny from E to E' in efficient representation.

From Theorem 1.15, it is easy to see that Isogeny reduces to EndRing.

Proposition 1.25. *Assuming the Generalised Riemann Hypothesis, the problem* Isogeny *reduces to* EndRing *in probabilistic polynomial time.*

Proof. The Isogeny problem immediately reduces to ℓ-IsogenyPath, which is equivalent to EndRing (Theorem 1.15). □

The converse reduction is trickier. As a solution to Isogeny is not guaranteed to have smooth degree, previous techniques have failed to prove that it is equivalent to EndRing. Theorem 1.22 unlocks this equivalence. Better yet, Theorem 1.26 below is unconditional. In particular, it implies that EndRing reduces to the ℓ-IsogenyPath independently of the Generalised Riemann Hypothesis.

Theorem 1.26 ([[PW24, Theorem 8.6]])**.** *The* EndRing *problem reduces to* Isogeny *in probabilistic polynomial time.*

Sketch of the proof. Since EndRing is equivalent to OneEnd (Theorem 1.22), it suffices to prove that OneEnd reduced to Isogeny. Let E be a supersingular curve for which we want to solve OneEnd, and let $\mathscr{A}$ be an algorithm for Isogeny. The idea is quite simple: generate a random walk $\varphi : E \to E'$, then call $\mathscr{A}$ to find an isogeny $\psi : E' \to E$, and return $\psi \circ \varphi \in \mathrm{End}(E)$. The unpredictibility of φ makes it possible to prove that whatever $\mathscr{A}$ does, the endomorphism $\psi \circ \varphi$ is non-scalar with overwhelming probability. □

1.4.2.3. *An unconditional algorithm for* EndRing *in time* $\tilde{O}(p^{1/2})$*.* We have established EndRing as a (or *the*) foundational problem of isogeny-based cryptography: all rests on its presumed hardness. But how hard is it? The fastest known algorithms have complexity

in $\tilde{O}(p^{1/2})$. However, all previous algorithms reaching that complexity have relied on unproven assumptions such as the Generalised Riemann Hypothesis.

With Theorem 1.22, we can prove that EndRing can be solved in time $\tilde{O}(p^{1/2})$ *unconditionally*. In contrast, the previous fastest unconditional algorithm had complexity $\tilde{O}(p)$ and only returned a full-rank subring of the endomorphism ring [Koh96, Theorem 75].

The first method to reach complexity $\tilde{O}(p^{1/2})$ under the Generalised Riemann Hypothesis consists in reducing EndRing to ℓ-IsogenyPath (with Theorem 1.15), and solving ℓ-IsogenyPath by a generic graph path-finding algorithm (with Proposition 1.6). Unconditionally, we can follow the same strategy, but using the unconditionally reduction from EndRing to ℓ-IsogenyPath (Theorem 1.26).

Theorem 1.27 ([[PW24, Theorem 8.8]]). *There is an algorithm solving the* EndRing *problem in expected time* $\tilde{O}\left(p^{1/2}\right)$.

Proof. This follows from the fact that there is an algorithm of complexity $\tilde{O}\left(p^{1/2}\right)$ for the 2-isogeny path problem (a folklore *meet-in-the-middle* strategy, see Proposition 1.6), and EndRing reduces to polynomially many instances of ℓ-IsogenyPath (Theorem 1.26). □

1.5. The fall of SIDH

Supersingular Isogeny Diffie-Hellman (SIDH) is a key exchange protocol proposed in 2011 by Jao and De Feo [JD11]. For the following eleven years, it was the crown jewel of isogeny-based cryptography. The influence of SIDH is notably illustrated by its incarnation *Supersingular Isogeny Key Encapsulation* (SIKE) [JAC+17], a primitive submitted to the call for standardization of new quantum-safe cryptographic primitives by the American National Institute of Standards and Technology (NIST). SIKE reached the final round of the process in 2022.

Yet, the security of SIDH (hence, SIKE) is not guaranteed by the hardness of the "pure" isogeny problem. It instead relies on a variant, where the image of some torsion points under a hidden isogeny are also revealed. This has come to be known as the *supersingular isogeny with torsion* (SSI-T) problem. It can be seen as an interpolation problem, which we formalise as follows.

Problem 1.28 (Interpolation). Let $\varphi : E \to E'$ be an isogeny. Given E, E', some points $P_i \in E$ and their images $\varphi(P_i)$, and a point $Q \in E$, compute $\varphi(Q)$.

This problem comes with a spectrum of difficulty, according to how many points are revealed — or more precisely, according to the order of the group generated by the revealed points. This problem has been shown to be weaker than the pure isogeny problem in a line of work pioneered by Petit [Pet17] in 2017 and expanded by multiple papers in the following years [KMP+21, BdQL+19, FKMT22]. SIDH remained immune to these attacks: they only tackled versions of Interpolation where the revealed points have large order — much larger than the points revealed by SIDH.

That was until 2022, when Castryck and Decru published a preprint [CD23] solving this problem in polynomial time for SIDH parameters, and effectively breaking even the highest security levels of SIKE. An earthquake in the isogeny world. A rapid series of works followed, culminating in a complete break of the Interpolation problem in any regime where the input fully determines the hidden isogeny, and the provided points have smooth order.

Our contribution to the attack first came in the form of the preprint [[Wes22b]], later published in the merged article [[MMP+23]]. In the preprint:

- We improve the efficiency of the attack, describing a much more direct approach than [CD23]. Essentially, Castryck and Decru reconstitute the isogeny bit-by-bit,

with iterative trial and error. In contrast, the algorithm in ⟦Wes22b, MMP⁺23⟧ recovers the secret in one go, dividing the complexity by the bit-length of the isogeny. As a result, the attack has become fast enough for constructive applications: it runs in a matter of milliseconds for interesting sets of parameters ⟦BDD⁺24⟧.
- We prove that the algorithm runs in polynomial time, assuming the Generalised Riemann Hypothesis, when the endomorphism ring of the domain curve is known (as in SIKE). In contrast, the original attack relied on heuristic assumptions. When the endomorphism ring is not known, we point out that the complexity is *heuristically* subexponential.

Robert then broke another barrier in [Rob23], proving that the polynomial running time can be reached even when the endomorphism ring of the source is not known.

1.5.1. Isogeny interpolation. In this section we choose to present the algorithm in a rather elementary form as in ⟦Wes22b⟧, yet including the powerful trick of Robert [Rob23]. The resulting Theorem 1.29 is close to the most general version known today, and benefits from a simple proof. Simple enough, perhaps, that the reader may come to wonder how it was missed for eleven years of scrutinizing SIDH.

Theorem 1.29. *Let $\varphi : E \to E'$ be an isogeny. Given:*
- *E, E', $\deg(\varphi)$,*
- *an integer $N > \deg(\varphi)$ coprime to $\deg(\varphi)$,*
- *a basis (P_1, P_2) of $E[N]$,*
- *the images $\varphi(P_1)$ and $\varphi(P_2)$,*
- *and a point $Q \in E$,*

one can compute $\varphi(Q)$ (or assert that the input is ill-formed) in polynomial time in the length of the input and in the largest prime factor of N.

Remark 1.30. Note that the statement can be improved by a number of other tricks: one can remove the coprimality condition, loosen the bound to $N^2 \gg \deg(\varphi)$, or replace $E[N]$ with more general subgroups.

The critical insight of Castryck and Decru came from looking at isogenies in higher dimension. For a complete picture of the situation, one should delve into the theory of abelian varieties, but for the simplified exposition below, it is sufficient to consider products of elliptic curves. A product $E_1 \times \cdots \times E_n$ is an abelian variety of dimension n.

Definition 1.31. Consider elliptic curves $E_1, \ldots, E_n, E'_1, \ldots, E'_n$, and isogenies $\varphi_{i,j} : E_i \to E'_j$. The matrix $M = (\varphi_{i,j})_{i,j}$ defines a map

$$\Phi : E_1 \times \cdots \times E_n \longrightarrow E'_1 \times \cdots \times E'_n : (P_i)_i \longmapsto \left(\sum_i \varphi_{i,j}(P_i) \right)_j .$$

We call $\tilde{M} = (\hat{\varphi}_{j,i})_{i,j}$ the dual matrix, which induces the map

$$\tilde{\Phi} : E'_1 \times \cdots \times E'_n \longrightarrow E_1 \times \cdots \times E_n : (P_i)_i \longmapsto \left(\sum_i \hat{\varphi}_{j,i}(P_i) \right)_j .$$

We say that Φ is an *N-isogeny*, for $N \in \mathbf{Z}_{>0}$, if $\tilde{M}M = N \cdot I_n$ (with I_n the identity matrix).

These N-isogenies are higher-dimensional analogs of "isogenies of degree N". In particular, when N is coprime to the characteristic p, an N-isogeny is determined by its kernel (which has order N^n), up to automorphisms of the target. A generalisation of Vélu's formulae allows to compute an N-isogeny from its kernel, in time polynomial in the largest prime factor of N. We are going to use the following simplified version of that fact.

Lemma 1.32. *There is an algorithm such that for any N-isogeny $\Phi : X \to Y$ over $\mathbf{F}_q$ the following holds: given X, Y, a generating set of $\ker(\Phi)$, and points $(P_i)_{i=1}^k \in X(\mathbf{F}_q)^k$, the algorithm returns $(\nu \circ \Phi(P_i))_{i=1}^k$ for some $\nu \in \mathrm{Aut}(Y)$ in polynomial time in q and in the largest prime factor of N.*

The reader familiar with theta models of abelian varieties can deduce this lemma from [LR12].

The strategy of the attack is the following. There is a secret isogeny $\varphi : E \to E'$, which we only know how to evaluate on the N-torsion, for some N (indeed, we know the image of a basis of $E[N]$, and N being smooth, we can efficiently rewrite any N-torsion point as a combination of the basis points, by a discrete logarithm computation). We can craft a matrix of isogenies containing φ as one of the entries, and the other entries are carefully chosen for the resulting matrix to define an N-isogeny. Each entry of the matrix can be evaluated on the N-torsion, which allows us the reconstruct the kernel (a subgroup of the N-torsion). From the kernel, we can evaluate the N-isogeny on any other input (Lemma 1.32), which in turn allows us to evaluate any entry of the matrix, including φ, on any input.

This strategy is most easily implemented when $N - \deg(\varphi)$ is a perfect square.

Proposition 1.33. *Theorem 1.29 holds if $A = N - \deg(\varphi)$ is a perfect square.*

Sketch of the proof. Let $a = \sqrt{A} \in \mathbf{Z}$. Consider the isogeny

$$\Psi = \begin{pmatrix} [a] & -\hat{\varphi} \\ \varphi & [a] \end{pmatrix} : E \times E' \longrightarrow E \times E'.$$

Let $\iota : E \to E \times E'$ be the inclusion and $\pi : E \times E' \to E'$ be the projection. We have

$$\pi \circ \Psi \circ \iota(Q) = \pi \circ \Psi(Q, 0) = \pi([a]Q, \varphi(Q)) = \varphi(Q).$$

Therefore, if we can evaluate Ψ, we can evaluate $\varphi = \pi \circ \Psi \circ \iota$. We show that we can evaluate Ψ by proving that it is an N-isogeny, and we know its kernel. Indeed, we have

$$\begin{aligned} \hat{\Psi} \circ \Psi &= \begin{pmatrix} [a] & \hat{\varphi} \\ -\varphi & [a] \end{pmatrix} \begin{pmatrix} [a] & -\hat{\varphi} \\ \varphi & [a] \end{pmatrix} \\ &= \begin{pmatrix} [a^2 + \deg(\varphi)] & 0 \\ 0 & [a^2 + \deg(\varphi)] \end{pmatrix} = N \cdot I_2, \end{aligned}$$

so Ψ is an N-isogeny. Furthermore, we have

$$\begin{aligned} \ker(\Psi) &= \{(P, Q) \in (E \times E')[N] \mid [a]P = \hat{\varphi}(Q) \text{ and } \varphi(P) = -[a]Q\} \\ &= \{([-a]P, \varphi(P)) \mid P \in E[N]\} \\ &= \langle ([-a]P_1, \varphi(P_1)), ([-a]P_2, \varphi(P_2)) \rangle, \end{aligned}$$

so we have a generating set for $\ker(\Psi)$. By Lemma 1.32, we can evaluate Ψ at any point. Therefore, we can compute $\varphi(Q)$ by evaluating $\pi \circ \Psi \circ \iota(Q)$. Note that Lemma 1.32 only evaluates $\nu \circ \varphi$ for some unknown $\nu \in \mathrm{Aut}(E)$. This defect can be corrected by applying the evaluation algorithm to (P_1, P_2, Q), which returns $(\nu \circ \varphi(P_1), \nu \circ \varphi(P_2), \nu \circ \varphi(Q))$. Then, one can recover ν as the unique automorphism sending each $\varphi(P_i)$ to $\nu \circ \varphi(P_i)$. □

The general case with known endomorphism ring. Of course, one cannot generally expect A to be a perfect square. This assumption is used to build an isogeny from E of degree A: the isogeny $[a] : E \to E$. For arbitrary A, the very same method can be adapted if one knows an isogeny $\gamma : E \to C$ of degree A: simply use γ in place of $[a]$. Given the endomorphism ring of E, one can always find such an isogeny γ in polynomial time

(assuming GRH ⟦Wes22b⟧), which proves that SIDH is broken in the special case where the endomorphism ring of the source curve E is known.

The general case. The final nail in SIDH's coffin was hammered by Robert [Rob23], by using endomorphisms in higher dimension. The case where $A = N - \deg(\varphi)$ is a perfect square leverages the fact that $\mathrm{End}(E)$ contains a very simple endomorphism of norm A: multiplication by $\sqrt{A}$. Robert observed that if A is a sum of two squares, then $\mathrm{End}(E \times E)$ contains a similarly simple endomorphism of norm A. Writing $N - \deg(\varphi) = a^2 + b^2$, we can do the same as above with

$$\Psi = \begin{pmatrix} a & b & -\hat{\varphi} & 0 \\ -b & a & 0 & -\hat{\varphi} \\ \varphi & 0 & a & -b \\ 0 & \varphi & b & a \end{pmatrix}.$$

Not every integer is a sum of two squares, but all are sums of four squares. In all generality, one can write $N - \deg(\varphi) = a^2 + b^2 + c^2 + d^2$, and use the matrix

$$\Psi = \begin{pmatrix} a & -b & -c & -d & -\hat{\varphi} & 0 & 0 & 0 \\ b & a & d & -c & 0 & -\hat{\varphi} & 0 & 0 \\ c & -d & a & b & 0 & 0 & -\hat{\varphi} & 0 \\ d & c & -b & a & 0 & 0 & 0 & -\hat{\varphi} \\ \varphi & 0 & 0 & 0 & a & b & c & d \\ 0 & \varphi & 0 & 0 & -b & a & -d & c \\ 0 & 0 & \varphi & 0 & -c & d & a & -b \\ 0 & 0 & 0 & \varphi & -d & -c & b & a \end{pmatrix}.$$

This proves Theorem 1.29.

1.5.2. The post-SIDH era. While the resolution of the Interpolation problem in polynomial time put an end to SIDH, it has not affected the core problems of isogeny-based cryptography: EndRing and its friends.

Since its devastating entrance, the interpolation algorithm has become a powerful *constructive* tool for isogeny-based cryptography. Indeed, it provides a new way to represent isogenies: given the degree d, a basis (P_1, P_2) of the N-torsion of the source (for some large enough, smooth N), and the images $(\varphi(P_1), \varphi(P_2))$, one can evaluate the isogeny φ on any other input. In other words, the tuple $(d, P_1, P_2, \varphi(P_1), \varphi(P_2))$ is an efficient representation of φ — called the *interpolation representation*, or the *HD representation* due to the role of higher dimensions. Contrary to previous methods like the isogeny-path representation, it works for arbitrary isogenies, even with large prime degree. In fact, it is in a sense a *universal* efficient representation: any efficient representation of an isogeny can be converted to an interpolation representation in polynomial time.

The first constructive application of this method was the digital signature scheme SQIsignHD, discussed in the next section. The subsequent "HD rush" [CLP23, BMP23, Ler23] came along practical improvements of the "HD machinery": fast algorithms to evaluate isogenies in higher dimension [Kun22, DMPR23].

1.6. The development of SQIsign

So far, we have focused on the theoretical foundations and cryptanalysis of isogeny-based cryptography. We conclude this chapter with a *constructive* contribution: the SQIsign digital signature scheme ⟦DKL+20, DLRW24⟧, whose development is intertwined with all previously presented results.

There had been several attempts at building an isogeny-based digital signature scheme before SQIsign. However, they were impractically slow, requiring several seconds [YAJ+17],

minutes [DG19], or more [GPS20] to sign and verify. The most efficient candidate was CSI-Fish [BKV19] which still takes approximately half a second to sign or verify, and requires a costly sub-exponential precomputation (currently out of reach beyond the lowest security level).

The design philosophy of SQIsign is rooted in the equivalence between EndRing and ℓ-IsogenyPath presented in Section 1.3. The idea is the following. In a digital signature scheme, a signer holds a secret key sk and a public key pk. A signature scheme consists in

(1) a *signing* procedure: on input the secret key sk, and a message m, the signer produces a signature σ. We expect this task to absolutely require knowledge of the secret key, so only the legitimate key holder can produce a valid signature.
(2) a *verification* procedure: on input the public key pk, a message m, and a signature σ, the verification certifies whether or not σ was indeed produced by the signing procedure on input m, with the secret key associated to pk.

In SQIsign, the public key is a supersingular elliptic curve $\mathsf{pk} = E$, and the secret key is its endomorphism ring $\mathsf{sk} = \mathrm{End}(E)$. Recovering the secret key $\mathrm{End}(E)$ from the public key E is precisely EndRing, a supposedly hard problem. This is a good start. To design a signing procedure, we need a task that can only be performed with knowledge of $\mathrm{End}(E)$. This is where the equivalence between EndRing and ℓ-IsogenyPath comes in. At least on an intuitive level, if signing requires solving an instance of ℓ-IsogenyPath involving E, the signer must know $\mathrm{End}(E)$, the secret key. This suggests a signing procedure of this form: let the message m encode some supersingular elliptic curve E_m. Using knowledge of $\mathrm{End}(E)$, the signer solves ℓ-IsogenyPath between E and E_m. The signature is an ℓ-isogeny path $\sigma : E \to E_m$. A verifier would recompute E_m, and check that σ is indeed a path between the "message curve" E_m and the public key E.

1.6.1. The SQIsign identification protocol. To turn this intuition into a secure scheme, it is useful to first design an identification protocol. An identification protocol provides more flexibility, and can be transformed into a signature scheme by standard techniques (e.g., via the Fiat-Shamir transform [FS86]). A *prover* (instead of a signer) knows the pair $(\mathsf{sk}, \mathsf{pk})$, and tries to convince a verifier (who knows only pk) that they know the secret sk. The main difference with a signature scheme is that the prover and verifier can directly interact, exchanging messages back and forth.

The SQIsign identification protocol, introduced in [[DKL+20]], has the following structure (which the reader may recognize to be a *sigma protocol*):

(1) *Commitment phase:* the prover generates a random pair $(E_{\mathsf{com}}, \mathrm{End}(E_{\mathsf{com}}))$. They keep $\mathrm{End}(E_{\mathsf{com}})$ secret, and send E_{com} to the verifier.
(2) *Challenge phase:* the verifier generates a non-backtracking random walk $\varphi_{\mathsf{chl}} : E \to E_{\mathsf{chl}}$ of length n in the 2-isogeny graph (for some fixed n). In other words, φ_{chl} is a random isogeny of degree 2^n with cyclic kernel. The verifier sends φ_{chl} to the prover.
(3) *Response phase:* knowing the secret key $\mathsf{sk} = \mathrm{End}(E)$ and $\varphi_{\mathsf{chl}} : E \to E_{\mathsf{chl}}$, the prover can compute $\mathrm{End}(E_{\mathsf{chl}})$. Now, knowing both $\mathrm{End}(E_{\mathsf{chl}})$ and $\mathrm{End}(E_{\mathsf{com}})$, the prover can compute an isogeny $\varphi_{\mathsf{rsp}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$. The prover sends φ_{rsp} to the verifier.
(4) *Verification phase:* the verifier checks that φ_{rsp} is indeed an isogeny connecting the commitment curve E_{com} to the challenge curve E_{chl}.

Intuitively, at the end of a successful interaction, the verifier should be convinced because they saw the prover solve an instance of ℓ-IsogenyPath involving E — a task that should require knowledge of $\mathrm{End}(E)$. More precisely, the instance involves E_{chl} rather than E, but given the isogeny $\varphi_{\mathsf{chl}} : E \to E_{\mathsf{chl}}$, knowledge of $\mathrm{End}(E)$ is equivalent to knowledge of

$\mathrm{End}(E_{\mathsf{chl}})$. Of course, we will give a formal argument, but first, a fatal flaw needs to be fixed. As it is presented, a cheating prover (who does not know the secret $\mathrm{End}(E)$) can fool the verifier. Indeed, the cheating prover could generate a commitment curve E_{com} by choosing a random isogeny $\varphi_{\mathsf{com}}^{\mathsf{cheat}} : E \to E_{\mathsf{com}}$. They may not be able to compute $\mathrm{End}(E_{\mathsf{com}})$, but it does not matter. In response to a challenge $\varphi_{\mathsf{chl}} : E \to E_{\mathsf{chl}}$, they would simply respond with $\varphi_{\mathsf{rsp}}^{\mathsf{cheat}} = \varphi_{\mathsf{chl}} \circ \hat{\varphi}_{\mathsf{com}}^{\mathsf{cheat}}$.

There is a simple fix to this issue, by ensuring that no part of the response φ_{rsp} factors through the challenge φ_{chl}. More precisely: the prover ensures (and the verifier checks) that $\hat{\varphi}_{\mathsf{chl}} \circ \varphi_{\mathsf{rsp}}$ has cyclic kernel.

1.6.2. Special soundness. With this fix, one can actually prove that this protocol "proves knowledge" of at least some non-trivial part of $\mathrm{End}(E)$. *Proving knowledge* can be formalized via a property called *special soundness*. A *transcript* of the protocol is the sequence of messages exchanged, i.e., the tuple $(E_{\mathsf{com}}, \varphi_{\mathsf{chl}}, \varphi_{\mathsf{rsp}})$; it is *accepting* if the corresponding verification succeeds. We say that the above protocol has special soundness if, given two accepting transcripts $(E_{\mathsf{com}}, \varphi_{\mathsf{chl}}, \varphi_{\mathsf{rsp}})$ and $(E_{\mathsf{com}}, \varphi'_{\mathsf{chl}}, \varphi'_{\mathsf{rsp}})$ with the same commitment E_{com} but distinct challenges $\varphi_{\mathsf{chl}} \neq \varphi'_{\mathsf{chl}}$, one can find a non-scalar endomorphism of E. The motivation for this definition is the following. If a prover can successfully respond to the challenge with good probability, then they can respond to at least two distinct challenges (for a fixed commitment). If they can respond to two distinct challenges for one fixed commitment, then the prover is capable of producing two accepting transcripts $(E_{\mathsf{com}}, \varphi_{\mathsf{chl}}, \varphi_{\mathsf{rsp}})$ and $(E_{\mathsf{com}}, \varphi'_{\mathsf{chl}}, \varphi'_{\mathsf{rsp}})$ with $\varphi_{\mathsf{chl}} \neq \varphi'_{\mathsf{chl}}$. With special soundness, this implies that the prover is capable of finding a non-scalar endomorphism of E.

In other words, if the protocol has special soundness, then a *good* prover (one who can successfully respond with good probability) necessarily knows a non-scalar endomorphism of E. Only one non-scalar endomorphism, and not the full ring $\mathsf{sk} = \mathrm{End}(E)$? Yes, but one is sufficient: as proved in Section 1.4, finding one non-scalar endomorphism is as hard as finding them all.

Proposition 1.34. *The SQIsign identification protocol has special soundness.*

Sketch of the proof. Suppose we have two accepting protocol transcripts $(E_{\mathsf{com}}, \varphi_{\mathsf{chl}}, \varphi_{\mathsf{rsp}})$ and $(E_{\mathsf{com}}, \varphi'_{\mathsf{chl}}, \varphi'_{\mathsf{rsp}})$ with distinct challenges $\varphi_{\mathsf{chl}} \neq \varphi'_{\mathsf{chl}}$. Then, by construction, $\alpha = \hat{\varphi}'_{\mathsf{chl}} \circ \varphi'_{\mathsf{rsp}} \circ \hat{\varphi}_{\mathsf{rsp}} \circ \varphi_{\mathsf{chl}}$ is in $\mathrm{End}(E)$. It remains to prove that it is not a scalar. By contradiction, suppose $\alpha = [\lambda]$ for some $\lambda \in \mathbf{Z}$. We deduce

$$\hat{\varphi}'_{\mathsf{rsp}} \circ \varphi'_{\mathsf{chl}} \circ [\lambda] = [\deg(\varphi'_{\mathsf{chl}}) \deg(\varphi'_{\mathsf{rsp}})] \circ \hat{\varphi}_{\mathsf{rsp}} \circ \varphi_{\mathsf{chl}}.$$

Since both $\hat{\varphi}'_{\mathsf{rsp}} \circ \varphi'_{\mathsf{chl}}$ and $\hat{\varphi}_{\mathsf{rsp}} \circ \varphi_{\mathsf{chl}}$ have cyclic kernel (thanks to the "fix" against the cheating prover!), we deduce that $\lambda = \deg(\varphi'_{\mathsf{chl}}) \deg(\varphi'_{\mathsf{rsp}})$ (the largest integer dividing the left- and right-hand side respectively), hence $\hat{\varphi}'_{\mathsf{rsp}} \circ \varphi'_{\mathsf{chl}} = \hat{\varphi}_{\mathsf{rsp}} \circ \varphi_{\mathsf{chl}}$. As $\deg(\varphi_{\mathsf{chl}}) = \deg(\varphi'_{\mathsf{chl}}) = 2^n$, we get

$$\ker(\varphi_{\mathsf{chl}}) = \ker(\hat{\varphi}_{\mathsf{rsp}} \circ \varphi_{\mathsf{chl}})[2^n] = \ker(\hat{\varphi}'_{\mathsf{rsp}} \circ \varphi'_{\mathsf{chl}})[2^n] = \ker(\varphi'_{\mathsf{chl}}).$$

This contradicts the fact that φ_{chl} and φ'_{chl} are *distinct* non-backtracking paths. □

1.6.3. The response. The response phase can be summarized as follows: knowing both $\mathrm{End}(E_{\mathsf{chl}})$ and $\mathrm{End}(E_{\mathsf{com}})$, the prover can compute in polynomial time an isogeny $\varphi_{\mathsf{rsp}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$. The prover sends this "response isogeny" φ_{rsp} to the verifier.

Implementing this idea in a secure way is not as straightforward as this summary suggests. The tricky part is that the response φ_{rsp} (and the way it is encoded) should not leak any non-trivial information about the secret. A naive application of the reduction from ℓ-IsogenyPath to EndRing (via MaxOrder) following Section 1.3.4 will result in a solution $\varphi_{\mathsf{rsp}} : E_{\mathsf{com}} \to E_{\mathsf{chl}}$ which "passes through" the special curve E_0: it factors

as $\varphi_{\mathsf{rsp}} = \varphi_2 \circ \varphi_1$ with $\varphi_1 : E_{\mathsf{com}} \to E_0$ and $\varphi_2 : E_0 \to E_{\mathsf{chl}}$. From such a response φ_{rsp}, a curious verifier can extract the factor $\varphi_2 : E_0 \to E_{\mathsf{chl}}$, and since the endomorphism ring of E_0 is publicly known, they can recover $\mathrm{End}(E_{\mathsf{chl}})$. From $\mathrm{End}(E_{\mathsf{chl}})$ and $\varphi_{\mathsf{chl}} : E \to E_{\mathsf{chl}}$, they can recover the secret key $\mathrm{End}(E)$.

One needs a better way to generate a response, one which would not leak the secret. In formal terms, we want the protocol to have the *zero-knowledge* property. SQIsign [[DKL+20]] and SQIsignHD [[DLRW24]] propose two distinct ways to select and represent the response. The set of possible responses is the space

$$\mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}})$$

of isogenies from E_{com} to E_{chl} (and the zero map). It is a lattice of rank 4, with a Euclidean structure induced by the positive definite integral quadratic form $\deg : \mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}}) \to \mathbf{Z}$. The methods discussed in Section 1.3 show that given $\mathrm{End}(E_{\mathsf{chl}})$ and $\mathrm{End}(E_{\mathsf{com}})$, one can find an isogeny $E_{\mathsf{com}} \to E_{\mathsf{chl}}$; from there, one can deduce not just one isogeny, but a complete basis of $\mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}})$.

The original SQIsign. SQIsign [[DKL+20]] selects the response as follows:

(1) Fix a large enough power of two 2^n, and solve the norm equation $\deg(\varphi) = 2^n$ for $\varphi \in \mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}})$ (in other words, find a lattice point of prescribed norm). We show in SQIsign [[DKL+20]] that this can be done in polynomial time by a variant of [KLPT14], without "passing through" the special curve E_0.
(2) The solution φ is initially represented as a formal linear combination of the basis of $\mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}})$. This representation cannot be revealed: it would leak the basis, which would leak the secret. Instead, it is converted to a 2-isogeny path of length n. This path is the response.

There are two issues with this approach. The first is efficiency. For Step 1, one needs 2^n of the order of $p^{3.5}$, and converting such a large isogeny into a 2-isogeny path is costly. After algorithmic improvements and low-level optimizations, we showed in [[DLLW23]] that signatures can be computed in the order of 400ms (for the NIST-I security level, which is roughly 128 bits of classical security). While not *absurdly* slow, it is orders of magnitudes slower than other post-quantum signature schemes (for instance, from the family of lattice-based cryptography).

The second issue concerns the security proof. It is difficult to argue that this response is *zero-knowledge*, that it does not leak any information about the secret. The reason is that the distribution of the solution φ of the norm equation is rather mysterious. While we can check that φ does not "pass through" a special curve E_0, there is no formal guarantee that this carefully crafted φ is totally innocuous. To prove the zero-knowledge property in [[DKL+20]], we resort to a heuristic assumption about the norm equation solver. This situation is unsatisfactory.

On the bright side, public keys and signatures are very compact. For the NIST-I security level, public keys are 64 bytes, and signatures are 205 bytes. In particular, the signature and public key sizes combined are an order of magnitude smaller than all other post-quantum signature schemes (they are 5.8 times smaller than the lattice-based scheme Falcon [PFH+17]).

SQIsignHD. SQIsignHD [[DLRW24]] resolves both drawbacks, bringing down the signing time to the order of 30ms, and enabling a much cleaner security proof. The signatures are even more compact, at a record-breaking 109 bytes for the NIST-I security level. This leap forward was enabled by the algorithmic breakthrough underlying the attack on SIDH. As discussed in Section 1.5.2, the attack provides a new way to represent an isogeny φ:

the *interpolation representation* $(\deg(\varphi), P_1, P_2, \varphi(P_1), \varphi(P_2))$, where (P_1, P_2) is a basis of a sufficiently large torsion subgroup. This representation applies to arbitrary isogenies, unlike the very constrained ℓ-isogeny path representation.

SQIsignHD selects the response as follows:

(1) Fix a bound $r > 0$, and sample a uniformly random $\varphi \in \mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}}) \cap \mathcal{B}(r)$, where $\mathcal{B}(r) = \{\varphi \mid \deg(\varphi) < r^2\}$ is the ball of radius r (with respect to the quadratic form deg).
(2) The solution φ is initially represented as a formal linear combination of the basis of $\mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}})$. Consider a basis (P_1, P_2) of $E[2^n]$ for n sufficiently large, and represent the response isogeny φ as $(\deg(\varphi), P_1, P_2, \varphi(P_1), \varphi(P_2))$.

In contrast with the mysterious distribution of φ in the original SQIsign, the uniform distribution $\varphi \in \mathrm{Hom}(E_{\mathsf{com}}, E_{\mathsf{chl}}) \cap \mathcal{B}(r)$ is well-understood, making it much easier to prove that this response does not leak any information about the secret.

SQIsignHD comes at a cost: the verifier has to check that the tuple

$$(\deg(\varphi), P_1, P_2, \varphi(P_1), \varphi(P_2))$$

indeed represents an isogeny $E_{\mathsf{com}} \to E_{\mathsf{chl}}$. To do so, they need to evaluate it with the interpolation algorithm, Theorem 1.29. This requires the computation of an isogeny between abelian varieties of dimension 2, 4 or 8. When $2^n - d$ is a perfect square, a 2-dimensional isogeny is sufficient. When $2^n - d$ is a sum of two squares, one can use a 4-dimensional isogeny. In all other cases, one needs an 8-dimensional isogeny.

In ⟦DLRW24⟧, we propose two constructions:

- SQIsign8D: this version benefits from a fully rigorous security analysis, but requires the computation of an isogeny in dimension 8. This has not been implemented, and is likely to be impractical.
- SQIsign4D: this version benefits from essentially the same security proof, but requires heuristic assumptions. These heuristics account for the fact that $2^n - d$ is restricted to sums of two squares. The computation of a 4-dimensional isogeny is more practical, but no optimized implementation is available yet.

In the recent preprint ⟦BDD+24⟧, we successfully devise a variant in dimension 2, which cumulates the advantages of all other variants. It benefits from a heuristic-free security proof, and boasts a verification time of the order of 4.5ms, and a signing time of the order of 80ms (even 50ms for a heuristic variant) on a 2GHz processor Intel Xeon Gold 6338 (Ice Lake). There is today significant activity around SQIsign, and improvements are arriving fast [SEMR23, RK24, JMKR23]. What could have been considered a prohibitively slow construction a few years ago is becoming increasingly competitive. It is currently submitted to the call for standardization of post-quantum digital signature schemes by the American National Institute of Standards and Technology (NIST).

2

Oriented elliptic curves

In this chapter, we present contributions that relate to oriented elliptic curves and their applications in cryptography. Orientations bring the methods of *complex multiplication* to supersingular elliptic curves. In particular, they induce an action of class groups on supersingular curves. The presumed hardness of inverting this action is the foundation of the "group action" branch of isogeny-based cryptography. We explore the connection of this branch with the rest of isogeny-based cryptography, showing that it is also supported by the hardness of computing endomorphisms. We study the fastest algorithms for the underlying problems, and build collections of orientations with useful properties.

This chapter is built around the presentation of the articles (in order of appearance):

⟦Wes22a⟧ Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, volume 13277 of *Lecture Notes in Computer Science*, pages 345–371. Springer, 2022.

⟦HW23⟧ Arthur Herlédan Le Merdy and Benjamin Wesolowski. The supersingular endomorphism ring problem given one endomorphism. Cryptology ePrint Archive, Paper 2023/1448, 2023. https://eprint.iacr.org/2023/1448.

⟦DFK+23⟧ Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography – PKC 2023*, volume 13940 of *Lecture Notes in Computer Science*, pages 345–375. Springer, 2023.

2.1. Introduction

The Diffie–Hellman key exchange is among the most emblematic protocols of modern cryptography. Introduced in 1974 [DH76], this paradigm-defining protocol allows two parties, Alice and Bob, to establish a shared secret (a key) over an insecure communication channel. It is remarkably simple. Alice and Bob first agree on a cyclic group G (written multiplicatively) and a generator g of this group. Alice secretly chooses a random integer a, and sends the group element g^a to Bob via the public communication channel. On his side, Bob also chooses a random integer b, and sends g^b to Alice. Both of them are now

able to compute the value

$$(g^a)^b = (g^b)^a,$$

a shared secret, to be used as a key to encrypt subsequent communications. An outsider eavesdropping on the communication channel knows g, g^a and g^b. Recovering the shared value g^{ab} from this information should be infeasible, and is known as the computational Diffie-Hellman problem. This problem, and the security of the protocol, is closely tied to the presumed hardness of the *discrete logarithm problem* in the group G: given g^a, recover a. Diffie and Hellman suggested to choose for group G the multiplicative group $\mathbf{F}_q^\times$ of a finite field. A more modern choice for G is the group of rational points $E(\mathbf{F}_q)$ of an elliptic curve over a finite field, in which the discrete logarithm problem seems to be as hard as it gets. Not only the Diffie–Hellman key exchange has remained to this day one of the most commonly used cryptographic protocols securing the Internet, it has sparked a long series of cryptosystems whose security relies on the hardness of computing discrete logarithms.

Shor's quantum algorithm [Sho97] is capable of solving the discrete logarithm problem in polynomial time in any group, rendering the Diffie–Hellman protocol and its lineage obsolete in a post-quantum world. Yet, it can be rescued, by replacing the group exponentiation $(a, g) \mapsto g^a$ with some other group action. A *group action* of a group G (with neutral element e) on a set X is a map

$$\star : G \times X \longrightarrow X : (g, x) \longmapsto g \star x$$

such that for any $x \in X$ and $g, h \in G$, we have

- *Compatibility:* $g \star (h \star x) = (gh) \star x$, and
- *Identity:* $e \star x = x$.

We say that the action is *effective* if the group operations and group action can be computed efficiently (as well as other natural tasks, like sampling random group elements). Such group actions are a powerful tool to build cryptographic schemes. For instance, when G is abelian, we can immediately generalize the Diffie–Hellman key exchange:

(1) Alice and Bob agree on an effective group action, and on a "reference point" $x_0 \in X$.
(2) Alice samples a random secret element $g_A \in G$, and sends $x_A = g_A \star x_0 \in X$ to Bob.
(3) Similarly, Bob samples a secret $g_B \in G$, and sends $x_B = g_B \star x_0 \in X$ to Alice.
(4) Alice can compute $x_{AB} = g_A \star x_B$, and Bob can compute $x_{BA} = g_B \star x_A$.

The compatibility property and the commutativity of the group imply that $x_{AB} = x_{BA}$; it is Alice's and Bob's shared secret. An eavesdropper can intercept x_0, $x_A = g_A \star x_0$ and $x_B = g_B \star x_0$. A *cryptographic group action* is one for which recovering the shared secret from the intercepted data is hard. In particular, it must be hard to "invert" the group action: given $x, y \in X$, compute g such that $y = g \star x$, when it exists. This problem is known as the *group action inversion problem*, or the *vectorisation problem*.

The Diffie-Hellman protocol is a particular case of the above protocol, where the action is given by exponentiation in a finite cyclic group. Given a finite abelian group G of order N, we have an effective action

$$(\mathbf{Z}/N\mathbf{Z})^\times \times G \longmapsto G : (n, g) \longmapsto g^n.$$

The corresponding vectorisation problem is the discrete logarithm problem: given g and g^n, compute n. But while Shor's algorithm can solve the discrete logarithm problem, no polynomial time (quantum) algorithm is known for the vectorisation problem is general. An effective group action which is "hard to invert" even for quantum algorithms would lead to a post-quantum Diffie–Hellman, and may bring to the post-quantum world many "discrete logarithm"-based cryptosystems.

Complex multiplication. Couveignes proposed in 1997 [Cou06] to use a group action arising from the theory of complex multiplication. The endomorphism ring of an elliptic curve E over $\mathbf{C}$ is either $\mathbf{Z}$ or an order $\mathcal{O}$ in a quadratic number field. In the latter case, we say that E has *complex multiplication by* $\mathcal{O}$. A central result of the theory states that the class group $\mathrm{Cl}(\mathcal{O})$ acts (*freely* and *transitively*!) on the collection $\mathcal{E}\ell\ell_{\mathbf{C}}(\mathcal{O})$ of isomorphism classes of elliptic curves with complex multiplication by $\mathcal{O}$.

This result reduces to finite fields. An elliptic curve over $\overline{\mathbf{F}}_p$ is *ordinary* when its endomorphism ring is an order $\mathcal{O}$ in a quadratic number field. Complex multiplication induces an action

$$\star : \mathrm{Cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}) \longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}),$$

where $\mathcal{E}\ell\ell_p(\mathcal{O})$ is the collection of isomorphism classes of pairs (E, ι) where E is an elliptic curve over $\overline{\mathbf{F}}_p$ and $\iota : \mathcal{O} \to \mathrm{End}(E)$ is a ring isomorphism. Couveignes conjectured in [Cou06] that the vectorization problem for this action is hard to invert, and it can be used in the generalized Diffie–Hellman key exchange.

This idea did not get much attention for a while. First, there was no "post-quantum" motivation yet. That changed in 2006 when Rostovtsev and Stolbunov independently rediscovered the idea [RS06], this time with the observation that the corresponding vectorization problem seemed hard even for quantum computers. Second, and perhaps more importantly, computing the action appeared to be very inefficient, making it unsuitable for any real-world application. Indeed, the best algorithms to compute this action are very sensitive to the degree of the field over which certain torsion subgroups are defined. It is very difficult to find ordinary elliptic curves in which the torsion behaves well enough. That, in turn, changed with the development of CSIDH in 2018 [CLM+18], combining two observations:

(1) the torsion is much easier to control for supersingular elliptic curves, and
(2) there is a similar group action for supersingular elliptic curves.

The idea is to use supersingular elliptic curves E defined over a prime-order field $\mathbf{F}_p$ (rather than the general case $\mathbf{F}_{p^2}$). Then, the subring $\mathbf{Z}[\pi] \subset \mathrm{End}(E)$ generated by the Frobenius endomorphism π is a quadratic order, and we have an action of $\mathrm{Cl}(\mathbf{Z}[\pi])$ like in the ordinary case.

Orientations. The notion of *orientation* introduced by Colò and Kohel [CK20] provides a generalized framework for these group actions, and have since proved to play a ubiquitous role in isogeny-based cryptography. Given a quadratic order $\mathcal{O}$, and an elliptic curve E, an $\mathcal{O}$*-orientation* of E is an embedding

$$\iota : \mathcal{O} \longrightarrow \mathrm{End}(E).$$

The pair (E, ι) is an $\mathcal{O}$-oriented curve. More precise definitions are provided in Section 2.2. Orientations generalize complex multiplication by providing an action

$$\star : \mathrm{Cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}) \longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}),$$

where $\mathcal{E}\ell\ell_p(\mathcal{O})$ is the collection of isomorphism classes of (primitively) $\mathcal{O}$-oriented elliptic curves over $\overline{\mathbf{F}}_p$.

Remark 2.1. The curves in $\mathcal{E}\ell\ell_p(\mathcal{O})$ are either all ordinary (and coincide with the definition of $\mathcal{E}\ell\ell_p(\mathcal{O})$ for ordinary curves given in the previous paragraph), or all supersingular.

The vectorization problem for this group action is the $\mathcal{O}$-Vect problem. Its presumed hardness for interesting families of orders $\mathcal{O}$ is the foundation of many cryptosystems [CLM+18, DG19, CLP23], ⟦DFK+23⟧.

2.1.1. Contributions and organisation of the chapter. The chapter is organized as follows.

Oriented elliptic curves. We start in Section 2.2 by presenting some background about oriented elliptic curves. We provide the basic definitions, discuss how to represent and manipulate them computationally, and how to compute the action of the class group. We sketch a parallel with the classical theory of ordinary elliptic curves and isogeny volcanoes.

Orientations and the supersingular endomorphism ring problem. The $\mathcal{O}$-VECT problem can be seen as an isogeny problem (the action of an ideal on an oriented curve corresponds to some kind of isogeny — see Section 2.2). Yet, this vectorization problem looks substantially different from the computational problems studied in Chapter 1. Does the endomorphism ring problem still play a foundational role in the "group action" subfamily of isogeny-based cryptography?

In Section 2.3, we present the positive answer obtained in the article ⟦Wes22a⟧. The main result, Theorem 2.18, states that the vectorization problem for the action of $\mathrm{Cl}(\mathcal{O})$ is equivalent to the endomorphism ring problem for $\mathcal{O}$-oriented curves. In particular, the security of the CSIDH cryptosystem [CLM+18] is equivalent to the problem of computing the endomorphism ring of supersingular curves defined over $\mathbf{F}_p$.

In the same article, we study the closely related problem $\mathcal{O}$-UBER (introduced in the article ⟦DDF+21⟧), and prove that it is equivalent to a harder variant of the endomorphism ring problem.

The supersingular endomorphism ring problem given one endomorphism. In Section 2.4, we discuss the concrete hardness of the problem $\mathcal{O}$-VECT, through the results of the article ⟦HW23⟧. More precisely, the article ⟦HW23⟧ considers the following problem: given a supersingular curve and one of its endomorphisms (non-scalar), find all the other endomorphisms. Knowing one endomorphism is essentially the same as knowing an orientation, so thanks to the equivalences of ⟦Wes22a⟧, this "endomorphism ring problem given one endomorphism" is essentially equivalent to $\mathcal{O}$-VECT. Only "essentially", because of an important nuance about the so-called *primitivity* of orientations. As a first step, we close this gap by proving in ⟦HW23⟧ that any orientation can efficiently be made primitive (Theorem 2.22).

We then solve $\mathcal{O}$-VECT (hence the endomorphism ring problem given one endomorphism). The fastest algorithms formerly required heuristic assumptions, and we prove that the same running times can be achieved under the Generalized Riemann Hypothesis (Theorem 2.20).

Knowing the structure of the acting group. Orientations provide a powerful framework to build cryptographic group actions. The acting group is the class group $\mathrm{Cl}(\mathcal{O})$ of a quadratic order. Computing the structure of a class group (even its order) is a notoriously difficult task (which we investigate further in Section 3.5.2). Interestingly, one does not need to know the structure of $\mathrm{Cl}(\mathcal{O})$ to compute its action on oriented curves, and run a Diffie–Hellman-like protocol. Still, the mystery around $\mathrm{Cl}(\mathcal{O})$ has proved to be a source of difficulty [BKV19], and a better understanding of the acting group may unlock more advanced cryptographic applications.

In Section 2.5, we present the result of the article ⟦DFK+23⟧: how to construct a family of oriented curves for easy-to-compute class groups $\mathrm{Cl}(\mathcal{O})$. We discuss its security, and how the careful choice of the order $\mathcal{O}$ impacts the hardness of the associated vectorization problem.

2.2. Oriented elliptic curves

Let K be a quadratic number field, and let $\mathcal{O}$ be an arbitrary order in K.

Definition 2.2 (Orientation)**.** A *K-orientation* on an elliptic curve E is an embedding $\iota : K \hookrightarrow \mathrm{End}(E) \otimes \mathbf{Q}$. It is an *$\mathcal{O}$-orientation* if $\iota(\mathcal{O}) \subseteq \mathrm{End}(E)$. It is a *primitive $\mathcal{O}$-orientation* if $\iota(\mathcal{O}) = \iota(K) \cap \mathrm{End}(E)$. Such a pair (E, ι) is called a (primitively) *$\mathcal{O}$-oriented* elliptic curve, and we say that E is (primitively) *$\mathcal{O}$-orientable*.

Remark 2.3. If (E, ι) is an $\mathcal{O}$-oriented elliptic curve, we will often consider ι as an embedding of $\mathcal{O}$ into $\mathrm{End}(E)$ (which naturally extends to an embedding of K into $\mathrm{End}(E) \otimes \mathbf{Q}$).

If E is ordinary, the endomorphism algebra $\mathrm{End}(E) \otimes \mathbf{Q}$ is itself a quadratic number field, so the curve can only be K-oriented if $K \cong \mathrm{End}(E) \otimes \mathbf{Q}$, and there are $\#\,\mathrm{Gal}(K/\mathbf{Q}) = 2$ distinct K-orientations. The situation is much richer for supersingular elliptic curves, since infinitely many quadratic fields embed in infinitely many ways in the quaternion algebra $B_{p,\infty} \cong \mathrm{End}(E) \otimes \mathbf{Q}$. The choice of an orientation (i.e., of a quadratic subfield of $\mathrm{End}(E) \otimes \mathbf{Q}$) allows one to transpose ideas and theorems from the classical theory of complex multiplication to the context of supersingular elliptic curves.

Given a K-oriented elliptic curve (E, ι), any isogeny $\varphi : E \to E'$ induces a K-orientation $\varphi_*(\iota)$ on E' defined as

$$\varphi_*(\iota)(\alpha) = (\varphi \circ \iota(\alpha) \circ \hat{\varphi}) \otimes \frac{1}{\deg(\varphi)}.$$

Definition 2.4 (Oriented isogeny)**.** Given two K-oriented elliptic curves (E, ι) and (E', ι'), an isogeny $\varphi : (E, \iota) \to (E', \iota')$ is *K-oriented* if $\iota' = \varphi_*(\iota)$.

We write $\mathcal{E}\ell\ell_p(\mathcal{O})$ for the set of primitively $\mathcal{O}$-oriented elliptic curves over $\overline{\mathbf{F}}_p$ up to K-oriented isomorphism.

Proposition 2.5 ([Onu21, Proposition 3.2])**.** *The curves in $\mathcal{E}\ell\ell_p(\mathcal{O})$ are supersingular if and only if p does not split completely in K and does not divide the conductor of $\mathcal{O}$.*

In this chapter, we only consider the supersingular case: we assume throughout that p does not split completely in K and does not divide the conductor of $\mathcal{O}$.

Class groups acting on sets of elliptic curves. Fix an oriented curve $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$. An $\mathcal{O}$-ideal $\mathfrak{a}$ induces a subgroup

$$E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}} \ker(\iota(\alpha)),$$

and a separable isogeny $\varphi_\mathfrak{a} : E \to E^\mathfrak{a}$ of kernel $E[\mathfrak{a}]$ called the $\mathfrak{a}$-multiplication. The target $E^\mathfrak{a}$ is the $\mathfrak{a}$-transform of (E, ι). This construction induces an action of $\mathcal{O}$-ideals on the set $\mathcal{E}\ell\ell_p(\mathcal{O})$, defined by

$$\mathfrak{a} \star (E, \iota) = (E^\mathfrak{a}, (\varphi_\mathfrak{a})_*(\iota)),$$

which factors through $\mathrm{Cl}(\mathcal{O})$. This action, well understood for ordinary elliptic curves with complex multiplication, was first studied in the context of oriented supersingular curves in [CK20] and [Onu21].

Theorem 2.6 ([Onu21])**.** *The action*

$$\mathrm{Cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}) \longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}) : ([\mathfrak{a}], (E, \iota)) \longmapsto \mathfrak{a} \star (E, \iota)$$

is free and has one or two orbits. For any orbit A, and any $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$, either $(E, \iota) \in A$, or both $(E, \bar{\iota})$ and $(E^{(p)}, \iota^{(p)})$ are in A. Here, $\bar{\iota}$ is the orientation ι composed with the canonical involution, and $\iota^{(p)}$ is the orientation induced by the Frobenius isogeny.

Proof. This theorem combines [Onu21, Proposition 3.3] and [Onu21, Theorem 3.4]. The statement about $(E, \bar{\iota})$ is from the proof of [Onu21, Proposition 3.3]. □

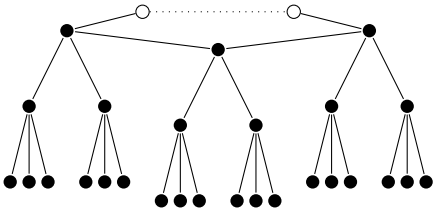


FIGURE 2.1. A (truncated) 3-isogeny volcano. The cycle at the top is the crater (level 0). The points just below the surface are at level 1, and points below them are at level 2. The full volcano continues to lower levels, forming an infinite $(3+1)$-regular graph.

2.2.1. Isogeny volcanoes. If (E,ι) is a primitively $\mathcal{O}$-oriented curve, and $\varphi : (E,\iota) \to (E',\iota')$ is a K-oriented isogeny, the target (E',ι') is not necessarily primitively $\mathcal{O}$-oriented. It could be primitively oriented by another order in K, inducing the following classification of isogenies.

Definition 2.7. Let $\varphi : (E,\iota) \to (E',\iota')$ be a K-oriented isogeny, and suppose that ι is a primitive $\mathcal{O}$-orientation, and ι' a primitive $\mathcal{O}'$-orientation. If $\deg(\varphi)$ is prime then one of the following three possibilities holds:

- the isogeny is *horizontal* when $\mathcal{O} = \mathcal{O}'$,
- the isogeny is *ascending* when $\mathcal{O} \subsetneq \mathcal{O}'$ (then, $[\mathcal{O}' : \mathcal{O}] = \deg(\varphi)$), and
- the isogeny is *descending* when $\mathcal{O} \supsetneq \mathcal{O}'$ (then, $[\mathcal{O} : \mathcal{O}'] = \deg(\varphi)$).

We say that an isogeny of composite degree is horizontal, ascending or descending if it factors as prime degree isogenies all of that same type.

This classification of isogenies gives the graph of oriented ℓ-isogenies the structure of an *isogeny volcano*, as in the classical case of ordinary curves [FM02]. Let K be a quadratic number field, and

$$\mathcal{E}\ell\ell_p(K) = \bigcup_{\mathcal{O}\subset K} \mathcal{E}\ell\ell_p(\mathcal{O})$$

be the collection of isomorphism classes of K-oriented curves. Let $\ell \neq p$ be a prime number, and consider the graph with vertex set $\mathcal{E}\ell\ell_p(K)$, and with edges representing oriented ℓ-isogenies between them.

Recall that any order $\mathcal{O}$ in K is of the form $\mathcal{O} = \mathbf{Z} + f\mathcal{O}_K$, where $\mathcal{O}_K$ is the maximal order, and $f = [\mathcal{O}_K : \mathcal{O}]$ is the conductor of $\mathcal{O}$. This classification organises the graph in levels: the level of a vertex $(E,\iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ is the valuation at ℓ of the conductor of $\mathcal{O}$. Ascending isogenies go from level i to level $i-1$, horizontal isogenies preserve the level, and descending isogenies go from level i to level $i+1$. Each connected component of the graph is called an ℓ-isogeny volcano, due to its particular shape (see Figure 2.1):

- Horizontal isogenies only exist at level 0 (known as the *crater*). The subgraph formed by the crater is regular of degree $d \leq 2$.
- From any vertex in the crater, there are d horizontal ℓ-isogenies, and $\ell + 1 - d$ descending ℓ-isogenies.
- From any vertex below the crater, there is 1 ascending ℓ-isogeny, and ℓ descending ℓ-isogenies.

Remark 2.8. A similar structure arises on the side of quaternions, through the Deuring correspondence (see Section 1.3.1). Instead of oriented curves, one can consider "oriented orders" in the quaternion algebra $B_{p,\infty}$. Let $\mathfrak{O} \subset B_{p,\infty}$ be a maximal order, and $\iota : K \to B_{p,\infty}$ an embedding. Let $\mathcal{O} \subset K$ be an order. The pair $(\mathfrak{O},\iota)$ is a *primitively $\mathcal{O}$-oriented order* if $\iota(\mathcal{O}) = \mathfrak{O} \cap \iota(K)$. A left $\mathfrak{O}$-ideal I is ascending (respectively horizontal, or

descending), if $\mathcal{O}_R(I)\cap\iota(K) \supsetneq \iota(\mathcal{O})$ (respectively $\mathcal{O}_R(I)\cap\iota(K) = \iota(\mathcal{O})$, or $\mathcal{O}_R(I)\cap\iota(K) \subsetneq \iota(\mathcal{O})$). Ascending, horizontal, or descending ideals correspond to ascending, horizontal, or descending isogenies through the Deuring correspondence, and we similarly obtain volcanoes of oriented orders.

2.2.2. Computing the action of the class group. We conclude this section with a few words on the first computational question raised by the class group action: given $\mathfrak{a}$ and (E,ι), compute $\mathfrak{a}\star(E,\iota)$. Any cryptographic application exploiting this action requires an efficient algorithm for this task — if not for any $\mathfrak{a}$, at least for some interesting family of ideals.

Computationally, an $\mathcal{O}$-orientation ι is encoded as a generator ω of $\mathcal{O}$ together with an efficient representation (Definition 1.3) of the endomorphism $\iota(\omega)$. An ideal $\mathfrak{a}$ is encoded as a pair (α, N) where $N = N(\mathfrak{a})$ is the norm and $\alpha \in \mathfrak{a}$ is an element such that $\mathfrak{a} = \langle\alpha, N\rangle$.

2.2.2.1. *The classical method.* There is a simple algorithm to compute $\mathfrak{a}\star(E,\iota)$ in time polynomial in the length of the encoding of (E,ι), in $\log(N(\mathfrak{a}))$ and in the largest prime-power factor of $N(\mathfrak{a})$. It consists in factoring $\mathfrak{a}$ into a product of prime ideals, and applying the action of each factor iteratively. For $\mathfrak{l}$ a prime ideal of norm ℓ, we proceed as follows:

(1) Write $\mathfrak{l} = \langle\alpha, \ell\rangle$ for some $\alpha \in \mathcal{O}$.
(2) Generate a basis (P_1, P_2) of $E[\ell]$; compute $Q_i = \iota(\alpha)(P_i)$.
(3) Find coefficients $a_1, a_2 \in \mathbf{Z}$ not both divisible by ℓ such that $a_1Q_1 + a_2Q_2 = 0$.
(4) Compute the separable isogeny $\varphi_{\mathfrak{a}} : E \to E^{\mathfrak{a}}$ with kernel $E[\mathfrak{l}] = \langle a_1P_1 + a_2P_2\rangle$.
(5) Return $\mathfrak{a}\star(E,\iota) = (E^{\mathfrak{a}}, (\varphi_{\mathfrak{a}})_*(\iota))$.

Each step can be performed within the claimed running time. A first bottleneck is the requirement to work with a basis of $E[\ell]$: in the worst case, it requires looking for points in a field extension of degree $O(\ell)$ — while polynomial time in ℓ, this can be very costly in practice. One may want to restrict the action to primes ℓ such that the points in $E[\ell]$ are defined over the base field $\mathbf{F}_{p^2}$ or a small extension.

The orientation $(\varphi_{\mathfrak{a}})_*(\iota)$ deserves more attention: it needs to be represented in an efficient way, a way that allows one to evaluate the corresponding endomorphisms $(\varphi_{\mathfrak{a}})_*(\iota)(\alpha)$ efficiently. For a while, the only polynomial-time method available was to represent it as a composition, as it is defined:

$$(\varphi_{\mathfrak{a}})_*(\iota)(\alpha) = (\varphi_{\mathfrak{a}} \circ \iota(\alpha) \circ \hat{\varphi}_{\mathfrak{a}}) \otimes \frac{1}{\deg(\varphi_{\mathfrak{a}})}.$$

Each factor of the composition can be evaluated efficiently, and the division by $\deg(\varphi_{\mathfrak{a}}) = \ell$ can be computed in polynomial time in ℓ.

However, this method "degrades" the quality of the representation after iterative applications. Indeed, after applying k actions of $\mathfrak{l}$, the resulting orientation requires a division by ℓ^k — a task that requires exponential time in k in the worst case. Therefore this classical method only allows one to evaluate the action of ideals $\mathfrak{a}$ of powersmooth norm (meaning that all its prime-power factors are bounded). The result has a "degraded" representation, making each further action costlier to evaluate.

2.2.2.2. *The case of CSIDH.* The most emblematic orientation in isogeny-based cryptography is the orientation by the Frobenius used in CSIDH [CLM+18]. Let $SS(\mathbf{F}_p)$ be the set of supersingular elliptic curves defined over the prime-order field $\mathbf{F}_p$, up to $\mathbf{F}_p$-isomorphism. For any $E \in SS(\mathbf{F}_p)$, its Frobenius endomorphism ϕ_p^E satisfies $\phi_p^E \circ \phi_p^E = [-p]$. Therefore, writing $\pi = \sqrt{-p}$, any $E \in SS(\mathbf{F}_p)$ has a $\mathbf{Z}[\pi]$-orientation given by

$$\iota : \mathbf{Z}[\pi] \longrightarrow \mathrm{End}(E) : \pi \longmapsto \phi_p^E.$$

Either this orientation is primitive, or its extension to $\mathbf{Z}[(1+\pi)/2]$ (when $\mathbf{Z}[\pi]$ is not maximal) is primitive. Reciprocally, one can show that if E is $\mathbf{Z}[\pi]$-orientable, then it is

isomorphic to a curve defined over $\mathbf{F}_p$. Hence the set $SS(\mathbf{F}_p)$ is essentially the union of $\mathcal{E}\ell\ell_p(\mathbf{Z}[\pi])$ and possibly $\mathcal{E}\ell\ell_p(\mathbf{Z}[(1+\pi)/2])$ (when $\mathbf{Z}[\pi]$ is not maximal).

One can apply the algorithm from Section 2.2.2.1 to compute the action of ideals, with one advantage: there is no need to compute the induced orientation $(\varphi_{\mathfrak{a}})_*(\iota)$, because it is automatically given by the Frobenius. In particular, there is no "degradation" issue, and one can apply the action of any *smooth* ideal (instead of powersmooth).

There is another major advantage: choosing a prime of the form $p = 4\ell_1 \dots \ell_n - 1$ where the ℓ_i are distinct odd primes, for any supersingular elliptic curve E over $\mathbf{F}_p$, we have $E[\ell_i] \subseteq E(\mathbf{F}_{p^2})$. Furthermore, each ℓ_i splits in $\mathbf{Z}[\pi]$. This means that for any i, there are two ideals of norm ℓ_i, and their action can be computed using arithmetic in $\mathbf{F}_{p^2}$ (even $\mathbf{F}_p$, with finer observations!), as opposed to extensions of degree $O(\ell_i)$ in general. This good control of the rationality of torsion subgroups is the reason why CSIDH, using supersingular curves, is considerably more practical than its ordinary predecessors [Cou06, RS06].

2.2.2.3. *Post-SIDH progress.* The fall of SIDH (see Section 1.5) came with a powerful new tool: the isogeny interpolation algorithm (Theorem 1.29). In the article ⟦HW23⟧, we exploit this tool to break the "powersmooth" barrier of the classical action-evaluation algorithm: we describe an algorithm to compute the action of any *smooth* ideal in polynomial time. This is explained in more detail in Section 2.4.1. The last barrier was lifted by Page and Robert in [PR23], where they describe a polynomial time algorithm to evaluate the action of *any* ideal. The practicality of [PR23] is not yet clear, and former methods may remain the preferred choice in certain contexts.

2.3. Orientations and the supersingular endomorphism ring problem

The action of class groups on oriented supersingular curves can be used to build cryptosystems. Since this action is defined by isogenies, it naturally qualifies as "isogeny-based cryptography". Yet, it feels like a substantially different paradigm from the kind of isogeny-based cryptography explored in Chapter 1.

In the article ⟦Wes22a⟧, we prove a strong connection between the endomorphism ring problem and the problem of inverting the group action. These computational equivalences further reinforce the foundational status of the endomorphism ring problem: its presumed hardness also governs the security of the "group action" family of isogeny-based cryptography. In particular, the reductions imply that the security of the CSIDH cryptosystem is equivalent to a version of the endomorphism ring problem. In this section, we present this bridge between class group problems and endomorphism ring problems, culminating in Theorem 2.18, the main result of ⟦Wes22a⟧.

2.3.1. Class group action problems. Recall the motivation for $\mathcal{O}$-orientations: they induce a group action, and if this group action is "hard to invert", then we can build secure cryptographic protocols. Inverting the group action is formalized as the vectorization problem. Let us specialize this notion to the context of oriented curves.

Definition 2.9 ($\mathcal{O}$-Vect)**.** Given primitively $\mathcal{O}$-oriented curves $(E,\iota), (E',\iota') \in \mathcal{E}\ell\ell_p(\mathcal{O})$, find an $\mathcal{O}$-ideal $\mathfrak{a}$ such that $(E',\iota') \cong \mathfrak{a} \star (E,\iota)$.

Instead of this "pure" vectorization problem, we work in ⟦Wes22a⟧ with two variants: one seemingly weaker, and the other seemingly stronger. One of the conclusions of the article is that they all are equivalent. The weaker variant, *ineffective* $\mathcal{O}$-Vect, does not require the solution to preserve the orientation. The main motivation for this omission is that per Theorem 2.6, such a solution always exists (even when the action has two orbits).

Problem 2.10 (Ineffective $\mathcal{O}$-Vect)**.** Given (E,ι) and (E',ι') in $\mathcal{E}\ell\ell_p(\mathcal{O})$, find an $\mathcal{O}$-ideal $\mathfrak{a}$ such that $E' \cong E^{\mathfrak{a}}$.

The stronger variant, *effective* $\mathcal{O}$-VECT, additionally asks to evaluate the action of $\mathfrak{a}$ on some arbitrary curve $(F, \jmath)$.

Problem 2.11 (Effective $\mathcal{O}$-VECT)**.** Given $(E, \iota), (E', \iota'), (F, \jmath) \in \mathcal{E}\ell\ell_p(\mathcal{O})$, find an $\mathcal{O}$-ideal $\mathfrak{a}$ (or decide that it does not exist) such that $(E', \iota') \cong \mathfrak{a} \star (E, \iota)$, and an efficient representation of $\varphi_\mathfrak{a} : (F, \jmath) \to \mathfrak{a} \star (F, \jmath)$.

For an adversary to break the key exchange in Section 2.1, they would need to solve the *effective* $\mathcal{O}$-VECT problem. This point used to be critical, since for a long time, no general polynomial-time algorithm was known to evaluate the action of arbitrary ideals. In ⟦Wes22a⟧, we circumvented the obstacle by proving that both variants are equivalent to a third problem: computing the endomorphism ring.

Now, one may observe that since the ineffective $\mathcal{O}$-VECT problem does not require the solution to preserve the orientation ι', the problem remains well-defined if we drop ι' from the input. This modification seems to make the problem much harder, and this presumed hardness (for large discriminant) has been introduced in ⟦DDF+21⟧ as the *Uber isogeny assumption*.

Problem 2.12 (Ineffective $\mathcal{O}$-UBER)**.** Given $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ and a primitively $\mathcal{O}$-orientable curve E', find an $\mathcal{O}$-ideal $\mathfrak{a}$ such that $E' \cong E^\mathfrak{a}$.

We similarly have an effective variant.

Problem 2.13 (Effective $\mathcal{O}$-UBER)**.** Given $(E, \iota), (F, \jmath) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ and a primitively $\mathcal{O}$-orientable curve E', find an $\mathcal{O}$-ideal $\mathfrak{a}$ (or decide that it does not exist) such that $E' \cong E^\mathfrak{a}$, and an efficient representation of $\varphi_\mathfrak{a} : (F, \jmath) \to \mathfrak{a} \star (F, \jmath)$.

Ineffective $\mathcal{O}$-VECT immediately reduces to ineffective $\mathcal{O}$-UBER. The results from ⟦Wes22a⟧ clarify the hardness gap between these problems.

2.3.2. Oriented endomorphism ring problems. In the article ⟦Wes22a⟧, we introduced three *oriented* variants of the endomorphism ring problem. The first is simply the endomorphism ring problem when the input is an oriented curve.

Problem 2.14 ($\mathcal{O}$-ENDRING)**.** Given $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$, compute $\mathrm{End}(E)$.

Remark 2.15. In light of the results of ⟦Wes21⟧ presented in Section 1.3, when we say "compute $\mathrm{End}(E)$", we mean both finding a basis of $\mathrm{End}(E)$ and an explicit isomorphism with a quaternionic order (i.e., solving both ENDRING and MAXORDER simultaneously).

The $\mathcal{O}$-ENDRING problem is a priori easier than ENDRING, as the orientation readily provides some information about the endomorphism ring. The second variant is the same problem but without being provided the orientation.

Problem 2.16 (ENDRING$|_\mathcal{O}$)**.** Given a primitively $\mathcal{O}$-orientable curve E, compute $\mathrm{End}(E)$.

Finally, we consider the seemingly harder problem of computing the endomorphism ring *and* an orientation.

Problem 2.17 ($\mathcal{O}$-ENDRING*)**.** Given a primitively $\mathcal{O}$-orientable curve E, compute $\mathrm{End}(E)$ and a primitive orientation $\iota : \mathcal{O} \to \mathrm{End}(E)$.

Clearly $\mathcal{O}$-ENDRING reduces to ENDRING$|_\mathcal{O}$, which in turn reduces to $\mathcal{O}$-ENDRING*. There seems to be a significant gap between $\mathcal{O}$-ENDRING and ENDRING$|_\mathcal{O}$ — except in the special case where the orientation is provided by the Frobenius, like in CSIDH. However, the distinction between ENDRING$|_\mathcal{O}$ and $\mathcal{O}$-ENDRING* is not clear. We prove in ⟦Wes22a, Corollary 1⟧ that they are equivalent when $\mathrm{disc}(\mathcal{O}) = O(p^{1/2})$. We later improved that bound to $O(p)$ in ⟦CHVW22⟧, and Eriksen and Leroux improved it further to $O(p^{4/3})$ in [EL24].

2.3.3. Two classes of problems. The main contribution of ⟦Wes22a⟧ is the identification of two classes of equivalent problems.

Theorem 2.18 (⟦Wes22a⟧)**.** *Assume the Generalized Riemann Hypothesis. If the factorization of* $\mathrm{disc}(\mathcal{O})$ *is known, then*

(1) $\mathcal{O}$-EndRing *and (effective or ineffective)* $\mathcal{O}$-Vect *are equivalent, and*
(2) $\mathcal{O}$-EndRing* *and (effective or ineffective)* $\mathcal{O}$-Uber *are equivalent,*

under probabilistic polynomial time reductions in the size of the instances and in $\#\,\mathrm{Cl}(\mathcal{O})[2]$.

The dependency in $\#\,\mathrm{Cl}(\mathcal{O})[2]$ only appears in the "vectorization to endomorphism ring" direction. Note that in all cases that are currently of interest, the discriminant of $\mathcal{O}$ is essentially one large prime (or prime power). Then, $\#\,\mathrm{Cl}(\mathcal{O})[2] = O(1)$ causes no trouble, and the factorization of $\mathrm{disc}(\mathcal{O})$ comes for free.

Previous work. The first article to investigate the relation between EndRing and the vectorisation problem was [CPV20], in the particular case of curves defined over $\mathbf{F}_p$ (with $\sqrt{-p} \in \mathcal{O}$, the orientation used in the CSIDH cryptosystem). They prove that knowledge of the endomorphism ring of a CSIDH public key allows one to recover the ideal class of the secret key. This surprising result, however, only implied a subexponential reduction from breaking CSIDH to computing endomorphism rings. In essence, they prove a reduction from the *ineffective* vectorisation problem, but not from its effective variant — at a time where the distinction was critical. In ⟦Wes22a⟧, we circumvent this issue, proving the first polynomial time reduction from the effective vectorisation problem (hence breaking CSIDH) to the endomorphism ring problem.

Subsequent work. The above theorems have been improved since ⟦Wes22a⟧. In the article ⟦CHVW22⟧, we develop new techniques for the "decisional $\mathcal{O}$-DiffieHellman problem", and as an application, we reduce the dependency in $\#\,\mathrm{Cl}(\mathcal{O})[2]$ to a subexponential quantity. For the reduction from $\mathcal{O}$-Vect to $\mathcal{O}$-EndRing, the article [EL24] has since entirely removed the dependency in $\#\,\mathrm{Cl}(\mathcal{O})[2]$, as well as the need for the factorization of the discriminant and the Generalized Riemann Hypothesis.

2.3.4. Finding a reference oriented elliptic curve. Recall that in Section 1.3, to prove the equivalence between EndRing and ℓ-IsogenyPath, the first ingredient was to build a "reference" curve E_0, a special curve with known endomorphism ring. Such a reference curve is equally important in proving the equivalence between $\mathcal{O}$-EndRing and $\mathcal{O}$-Vect. But now, the reference curve must be primitively $\mathcal{O}$-oriented.

In ⟦Wes22a⟧, we present a general algorithm to generate such a curve in $\mathcal{E}\ell\ell_p(\mathcal{O})$ in polynomial time. This already requires knowledge of the factorization of $\mathrm{disc}(\mathcal{O})$.

Theorem 2.19 (⟦Wes22a, Lemma 4⟧)**.** *Assume the Generalized Riemann Hypothesis. There is an algorithm which, given a prime* p*, a quadratic order* $\mathcal{O}$*, and the factorization of* $\mathrm{disc}(\mathcal{O})$*, returns (or asserts that it does not exist) a primitively oriented curve* $(E_0, \iota_0) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ *together with a basis of its endomorphism ring* $\mathrm{End}(E_0)$*, and runs in polynomial time in* $\log p$ *and* $\log(|\,\mathrm{disc}(\mathcal{O})|)$.

Sketch of the proof. The algorithm consists in first solving the quaternion analog of the problem (find a maximal order $\mathfrak{O} \subset B_{p,\infty}$ in which $\mathcal{O}$ primitively embeds), then solve the "reverse Deuring correspondence" (find an elliptic curve with endomorphism ring isomorphic to $\mathfrak{O}$). The second step uses techniques discussed in Section 1.3.

Let us say a bit more about the first step. Embedding $\mathcal{O}$ into $B_{p,\infty}$, then extending the quadratic order to a maximal quaternionic order $\mathfrak{O}$, can be done in a rather straightforward manner, by solving quadratic equations. However, the resulting embedding $\mathcal{O} \to \mathfrak{O}$ is not necessarily primitive. Instead, we consider $\mathcal{O}_K$ the maximal order in $K = \mathcal{O} \otimes \mathbf{Q}$, and find

an embedding $\mathcal{O}_K \to \tilde{\mathfrak{O}}$ to some maximal order $\tilde{\mathfrak{O}}$ in $B_{p,\infty}$. This embedding is necessarily primitive, since $\mathcal{O}_K$ is maximal. To find a primitive embedding $\mathcal{O} \to \mathfrak{O}$, we carefully work our way down the "volcano of oriented orders" (see Remark 2.8). □

2.3.5. $\mathcal{O}$-EndRing **reduces to** $\mathcal{O}$-Vect**.** Let us sketch the reduction from $\mathcal{O}$-EndRing to $\mathcal{O}$-Vect. Suppose we are given an instance $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ of the $\mathcal{O}$-EndRing problem, and we have an oracle solving the $\mathcal{O}$-Vect problem. Find $(E_0, \iota_0) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ together with a basis of its endomorphism ring $\mathrm{End}(E_0)$, as in Theorem 2.19. The oracle finds an $\mathcal{O}$-ideal $\mathfrak{a}$ such that $E \cong E_0^{\mathfrak{a}}$. The isogeny $\varphi_{\mathfrak{a}} : E_0 \to E$ corresponds to the ideal $I_{\varphi_{\mathfrak{a}}} = \mathrm{End}(E_0) \cdot \iota_0(\mathfrak{a})$, and we have $\mathcal{O}_R(I_{\varphi_{\mathfrak{a}}}) \cong \mathrm{End}(E)$. Knowing the ideal $\mathfrak{a}$, the orientation ι_0 and a basis of $\mathrm{End}(E_0)$, we can compute a basis of $I_{\varphi_{\mathfrak{a}}}$. The right-order $\mathcal{O}_R(I_{\varphi_{\mathfrak{a}}})$ can be computed with [Rón92, Theorem 3.2], thereby solving $\mathcal{O}$-EndRing for (E, ι).

The equivalence between MaxOrder and EndRing (discussed in Section 1.3.5) plays a role under the hood, to translate the order $\mathcal{O}_R(I_{\varphi_{\mathfrak{a}}})$ (only *isomorphic* to $\mathrm{End}(E)$) into the actual endomorphism ring $\mathrm{End}(E)$.

2.3.6. $\mathcal{O}$-Vect **reduces to** $\mathcal{O}$-EndRing**.** Reducing $\mathcal{O}$-Vect to $\mathcal{O}$-EndRing is more delicate. Suppose we are given an instance $(E, \iota), (E', \iota') \in \mathcal{E}\ell\ell_p(\mathcal{O})$ of $\mathcal{O}$-Vect, and we know the endomorphism rings $\mathrm{End}(E)$ and $\mathrm{End}(E')$. Knowing the endomorphism rings (or equivalently, isomorphic orders in $B_{p,\infty}$), it is easy to find a "connecting ideal" between them, corresponding to an isogeny $E \to E'$. However, in general, such an ideal is not related to an $\mathcal{O}$-ideal $\mathfrak{a}$ (equivalently, the isogeny $E \to E'$ does not preserve the orientations). Careful steps which we do not detail here allow one to craft a special connecting ideal of the form $\mathrm{End}(E) \cdot \iota(\mathfrak{a})$, from which one can recover an $\mathcal{O}$-ideal $\mathfrak{a}$.

2.4. The supersingular endomorphism ring problem given one endomorphism

Given a supersingular elliptic curve E and a non-scalar endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbf{Z}$, how hard is it to find all the other endomorphisms of E? This question naturally emerges in the context of oriented curves, as the data of an orientation is essentially the same as one endomorphism (a generator of the orientation). In the article ⟦HW23⟧, we prove the following theorem.

Theorem 2.20 (⟦HW23, Theorem I and II⟧)**.** *There is an algorithm which, given a supersingular curve E and an endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbf{Z}$, computes the endomorphism ring of E in*

- *classical expected time $l^{O(1)} |\operatorname{disc}(\mathbf{Z}[\alpha])|^{1/4}$, or*
- *quantum subexponential time $l^{O(1)} L_{|\operatorname{disc}(\mathbf{Z}[\alpha])|}(1/2)$,*

where l is the length of the input, assuming the Generalized Riemann Hypothesis.

This question closely relates to the hardness of the $\mathcal{O}$-EndRing problem (and therewith $\mathcal{O}$-Vect, thanks to the equivalence in Theorem 2.18). Indeed, in the $\mathcal{O}$-EndRing problem, one is given a primitive orientation, hence an endomorphism. Reciprocally, any endomorphism α induces an orientation of E by the order $\mathbf{Z}[\alpha]$. However, this orientation is not necessarily primitive, a requirement in $\mathcal{O}$-EndRing. Thanks to this observation, we prove Theorem 2.20 in three steps. Suppose we are given a curve E and an endomorphism $\alpha \in \mathrm{End}(E) \setminus \mathbf{Z}$.

(1) *Primitivization.* First, we extend the orientation $\mathbf{Z}[\alpha] \to \mathrm{End}(E)$ to a primitive orientation $\iota : \mathcal{O} \to \mathrm{End}(E)$, for some superorder $\mathcal{O} \supseteq \mathbf{Z}[\alpha]$. This is the Primitivization step. The Primitivization problem was first introduced in [ACL+23], where it was believed to be hard. We show that it can actually be solved in polynomial time.

(2) *Reduction.* Now that we have a primitively $\mathcal{O}$-oriented curve (E, ι), we can apply the equivalence of Theorem 2.18: to compute $\mathrm{End}(E)$, it is sufficient to solve the $\mathcal{O}$-Vect problem.
(3) *Vectorization.* Finally, we solve $\mathcal{O}$-Vect. Heuristic algorithms for this step had already been described, and the contribution of ⟦HW23⟧ lies in the design and analysis of rigorous algorithms, assuming the Generalized Riemann Hypothesis

The reduction step was the object of Section 2.3. We now discuss the algorithms for Primitivization and $\mathcal{O}$-Vect.

2.4.1. Primitivization. The primitivization problem is the following computational problem.

Problem 2.21 (Primitivization)**.** Given an $\mathcal{O}_0$-oriented elliptic curve (E, ι_0), find a superorder $\mathcal{O} \supseteq \mathcal{O}_0$ and a *primitive* orientation $\iota : \mathcal{O} \to \mathrm{End}(E)$ such that $\iota|_{\mathcal{O}_0} = \iota_0$.

For supersingular elliptic curves, this problem is first considered in [ACL+23], where a quantum subexponential-time algorithm is given. In the article ⟦HW23⟧, we prove that given the factorization of $\mathrm{disc}(\mathcal{O}_0)$, the problem can be solved in classical polynomial expected time. The idea comes from observing that Primitivization is very similar to the endomorphism ring problem for ordinary elliptic curves, which can be solved in polynomial time with Robert's algorithm [Rob22] (up to some factorization).

Indeed, given an ordinary elliptic curve E over the finite field $\mathbf{F}_q$, the q-Frobenius endomorphism π_q generates a quadratic subring $\mathbf{Z}[\pi_q] \subseteq \mathrm{End}(E)$. The full endomorphism ring is given by the primitivization of the inclusion $\mathbf{Z}[\pi_q] \to \mathrm{End}(E)$ (recall that in the ordinary case, the endomorphism ring is a quadratic ring). Adapting Robert's algorithm, we get the following theorem.

Theorem 2.22 (⟦HW23, Corollary 5.2⟧)**.** *There is a probabilistic polynomial time algorithm solving* Primitivization*, given the factorization of* $\mathrm{disc}(\mathcal{O}_0)$*.*

Sketch of the proof. The input orientation $\iota_0 : \mathcal{O}_0 \to \mathrm{End}(E)$ is given by a generator ω_0 of $\mathcal{O}_0$, and an efficient representation of its image $\alpha_0 = \iota_0(\omega_0)$. Without loss of generality, we can write $\omega_0 = f\omega$ where f is the conductor of $\mathcal{O}_0$, and ω is a generator of the maximal order of $\mathcal{O}_0 \otimes \mathbf{Q}$.

Let ℓ be a prime factor of f. If α_0/ℓ is *not* an endomorphism, we say that the orientation is *locally primitive at* ℓ. Otherwise, α_0/ℓ is an endomorphism, and the orientation $\iota_0 : \mathbf{Z}[f\omega] \to \mathrm{End}(E) : f\omega \mapsto \alpha_0$ can be extended to the superorder

$$\mathbf{Z}[(f/\ell)\omega] \to \mathrm{End}(E) : (f/\ell)\omega \mapsto \alpha_0/\ell.$$

One can perform this step recursively for all prime factors of the conductor, until the orientation is locally primitive at each of them. When the orientation is locally primitive at every prime factor of f, it is a primitive orientation.

A question remains: how to test whether α_0/ℓ is an endomorphism (i.e., whether ℓ divides α_0 in $\mathrm{End}(E)$), and if is it, how to find an efficient representation for it? Note that ℓ divides α_0 if and only if $\ker([\ell]) \subseteq \ker(\alpha_0)$, so this task is easy when ℓ is a small prime number. But dealing with large primes (or even large prime powers) requires more care. An idea of [Rob22] which we generalize in ⟦HW23⟧ consists in using the *interpolation representation* (see Section 1.5.2): for any isogeny $\varphi : E \to E'$, powersmooth integer $N > \deg(\varphi)$, and basis (P_1, P_2) of $E[N]$, the tuple $(\deg(\varphi), P_1, P_2, \varphi(P_1), \varphi(P_2))$ is an efficient representation of φ. Furthermore, for any tuple of the same form (d, P_1, P_2, Q_1, Q_2), one can efficiently verify whether it actually represents an isogeny. Therefore, we can choose an integer N coprime to ℓ, and consider a tuple $(\deg(\alpha_0), P_1, P_2, \alpha_0(P_1), \alpha_0(P_2))$ representing α_0. Then, α_0/ℓ is an endomorphism if and only if it is represented by

$$(\deg(\alpha_0)/\ell^2, P_1, P_2, [\ell^{-1} \bmod N]\alpha_0(P_1), [\ell^{-1} \bmod N]\alpha_0(P_2)).$$

One can efficiently verify whether that tuple actually represents an isogeny, and when it does, it readily provides an efficient representation of α_0/ℓ. □

2.4.1.1. *A first consequence: computing the action of smooth ideals.* Recall that the classical algorithm to evaluate the action $\mathfrak{a} \star (E, \iota)$ runs in polynomial time in the largest prime-power factor of $N(\mathfrak{a})$. Therefore, to be efficient, it requires $\mathfrak{a}$ to be powersmooth. The reason is that the induced isogeny

$$\varphi_*(\iota)(\alpha) = (\varphi \circ \iota(\alpha) \circ \hat{\varphi}) \otimes \frac{1}{\deg(\varphi)}.$$

is represented in the most straightforward way, as a composition of φ, $\iota(\alpha)$, $\hat{\varphi}$, and a division by $\deg(\varphi)$. The dependence in the prime-power factors comes from the division.

One may be tempted to decompose $\mathfrak{a}$ as a product of powersmooth ideals, and applying the action of each factor iteratively. However, iterative applications of the classical algorithm increases the denominator, hence "degrades" the quality of the representation of the orientation. A solution would be, at each step, to clear the denominator: finding another representation of the orientation. This can be done by representing the orientation with the interpolation method, like in the Primitivization algorithm. Concretely, to avoid the division, consider the division-free orientation by the order $\mathbf{Z} + \deg(\varphi)\mathcal{O}$. Now, instead of dividing by $\deg(\varphi)$, solve Primitivization, which returns a good representation of the primitive $\mathcal{O}$-orientation $\varphi_*(\iota)$. This method allows one to compute the action of $\mathfrak{a}$ in time polynomial in the largest prime factor of $\mathfrak{a}$, instead of its largest prime-power factor.

2.4.1.2. *A second consequence: ascending the volcano.* Recall that fixing a quadratic field K, and a prime ℓ, the graph of K-oriented ℓ-isogenies forms a so-called volcano. A vertex in the volcano is a primitively $\mathcal{O}$-oriented curve (E, ι) for some order $\mathcal{O} = \mathbf{Z} + \ell^i f\mathcal{O}_K$, where $\gcd(f, \ell) = 1$, and i is the level of the vertex in the volcano.

An ascending isogeny from (E, ι) reaches a curve (E', ι') primitively oriented by $\mathcal{O}' = \mathbf{Z} + \ell^{i-1} f\mathcal{O}_K$. The conductor decreases by a factor ℓ. In particular, the discriminant decreases, so the size of the class group decreases, and the corresponding vectorization problem gets simpler. This motivates the problem of *ascending the volcano*: given an instance of, say, $\mathcal{O}$-EndRing, one could first try to ascend to the level of $\mathcal{O}'$-oriented curves for an order $\mathcal{O}'$ of smallest possible discriminant, where the problem is easier (then "transport the endomorphism information" back down through the vertical isogenies).

Previously, ascending walks in the volcano were limited to small powers of small primes, because of the degradation induced by division of endomorphisms. We proved in ⟦Wes22a, Theorem 5⟧ that $(\mathbf{Z} + c\mathcal{O})$-EndRing reduces to $\mathcal{O}$-EndRing in polynomial time in the largest prime-power factor of c. Thanks to the new division method, we obtain the following theorem, unlocking large powers of small primes.

Theorem 2.23 (⟦HW23, Theorem 7.11⟧)**.** *Let c be a positive integer, and $\mathcal{O}$ a quadratic order. The $(\mathbf{Z} + c\mathcal{O})$-*EndRing *problem reduces to $\mathcal{O}$-*EndRing *in probabilistic polynomial time in the length of the input, and in the largest prime factor of c.*

2.4.2. Vectorization. The resolution of $\mathcal{O}$-Vect is the heart of Theorem 2.20, dominating the running times.

Classical algorithm. The case of ordinary elliptic curves (and its natural generalization to CSIDH) has been studied as early as [GHS02], and the development of theoretical tools such as [JMV09] unlocked a rigorous analysis of these algorithms. We now present the solution for general orientations, which combines these "ordinary" methods with the new technique for the evaluation of the action of smooth ideals (Section 2.4.1.1).

The fastest known classical algorithm for the $\mathcal{O}$-VECT problem is a meet-in-the-middle algorithm. We are given two $\mathcal{O}$-oriented elliptic curves (E, ι) and (E', ι') (we suppose they are in the same orbit). Note that the orbit has size $\#\operatorname{Cl}(\mathcal{O}) \approx |\operatorname{disc}(\mathbf{Z}[\alpha])|^{1/2}$.

(1) First, generate random ideals $\mathfrak{a}_i$ until the list $T = (\mathfrak{a}_i \star (E, \iota))_i$ contains about $|\operatorname{disc}(\mathbf{Z}[\alpha])|^{1/4}$ distinct isomorphism classes.
(2) Second, generate random ideals $\mathfrak{b}$ until $\mathfrak{b} \star (E', \iota')$ is an entry in the list T, say $\mathfrak{b} \star (E', \iota') = \mathfrak{a}_i \star (E, \iota)$. Return the ideal class $[\mathfrak{b}]^{-1}[\mathfrak{a}_i]$.

If one can sample ideals $\mathfrak{a}$ uniformly distributed in the class group, the corresponding curves $\mathfrak{a} \star (E, \iota)$ are uniformly distributed in the orbit. Using such a procedure, each of the above steps requires approximately $|\operatorname{disc}(\mathbf{Z}[\alpha])|^{1/4}$ samples, leading to the claimed running time.

The main difficulty thus resides in sampling uniformly random classes $[\mathfrak{a}]$ in a way that the action $\mathfrak{a} \star (E, \iota)$ can be computed efficiently. This is where random walks come in.

We define random walks in $\mathcal{E}\ell\ell_p(\mathcal{O})$ as follows. Consider a collection $\mathcal{S}$ of ideals in $\mathcal{O}$. An *$\mathcal{S}$-step* from a starting point $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ consists in sampling $\mathfrak{a} \in \mathcal{S}$ uniformly at random, and going to $\mathfrak{a} \star (E, \iota)$. An *$\mathcal{S}$-walk* of length k is a sequence of k consecutive $\mathcal{S}$-steps. A classical choice for $\mathcal{S}$ is the set $\mathcal{S} = \mathscr{P}_B$ of all prime ideals of norm at most some bound B. It is well-known that $\mathscr{P}_B$-walks have rapid-mixing properties.

Theorem 2.24 ([JMV09], adapted to the context of oriented curves)**.** *Let $\varepsilon > 0$. Assuming the Generalized Riemann Hypothesis, there are bounds*

$$B = O((\log|\operatorname{disc}(\mathcal{O})|)^3), \text{ and}$$
$$\kappa = \log(|\operatorname{disc}(\mathcal{O})|) \cdot \operatorname{poly}(\log\log(|\operatorname{disc}(\mathcal{O})|), \log(1/\varepsilon))$$

such that for any $k \geq \kappa$, the endpoint of a $\mathscr{P}_B$-walk in $\mathcal{E}\ell\ell_p(\mathcal{O})$ of length k is at total variation distance at most ε from the uniform distribution in the orbit of the starting point.

A $\mathscr{P}_B$-walk corresponds to the action of a B-smooth ideal. Smooth, but not powersmooth. This was not an issue in the classical case of ordinary elliptic curves. For the general case, it motivated the development of the algorithm described in Section 2.4.1.1 for the action of smooth ideals. The equidistribution property of random walks, together with the efficient algorithm to compute them, results in a rigorous meet-in-the-middle algorithm for the $\mathcal{O}$-VECT problem, assuming the Generalized Riemann Hypothesis.

Quantum algorithm. The subexponential quantum resolution of the $\mathcal{O}$-VECT proven in [[HW23]] is based on the work of Childs, Jao and Soukharev [CJS14] to construct an isogeny between two given isogenous ordinary elliptic curves. In particular, we use the fact that given two oriented elliptic curves $(E_0, \iota_0), (E_1, \iota_1) \in \mathcal{E}\ell\ell_p(\mathcal{O})$ in the same orbit, finding an $\mathfrak{O}$-ideal $\mathfrak{a}$ such that $\mathfrak{a} \star (E_0, \iota_0) = (E_1, \iota_1)$ can be viewed as an instance of the hidden shift problem.

Problem 2.25 (HIDDENSHIFT)**.** Given a finite abelian group $(A, +)$, a finite set $S \subset \{0,1\}^m$, and two black-box functions $f_0, f_1 : A \to S$ where f_0 is injective and such that there exists an element $s \in S$ verifying $f_1(x) = f_0(s + x)$ for any $x \in S$, find the element s called the *shift* hidden by f_0 and f_1.

Defining $f_0, f_1 : \operatorname{Cl}(\mathcal{O}) \to \mathcal{E}\ell\ell_p(\mathcal{O})$ as

$$f_0([\mathfrak{a}]) = \mathfrak{a} \star (E, \iota), \text{ and}$$
$$f_1([\mathfrak{a}]) = \mathfrak{a} \star (E', \iota'),$$

one can use Kuperberg's quantum algorithm to solve the HIDDENSHIFT problem (hence $\mathcal{O}$-VECT) in a subexponential number of queries to f_0 and f_1.

Theorem 2.26 (Theorem 7.1. [Kup05]). *There is a quantum algorithm for* HiddenShift *over abelian groups with time and query complexity* $2^{O(\sqrt{\log n})}$*, where* n *is the size of the abelian group.*

The complexity of evaluating the functions f_i (a "query") presents a challenge: it requires evaluating the action of arbitrary ideals. We can generalize the strategy of [CJS14]: find a smooth representative of the class $[\mathfrak{a}]$, then apply the action of that representative with the method of Section 2.4.1.1. A subexponential smoothness bound leads to a subexponential complexity. Today, the algorithm of [PR23] enables a more straightforward approach, since it can evaluate the action of any ideal in polynomial time.

2.5. Knowing the structure of the acting group

Orientations provide a powerful framework to obtain a cryptographic group action. The orientation provided by the Frobenius on supersingular curves over $\mathbf{F}_p$ is the most prominent incarnation of this framework, and the heart of the CSIDH key exchange (see Section 2.2.2.2).

One downside of this framework is that the structure of the acting group $\mathrm{Cl}(\mathcal{O})$ is, in general, hard to compute. "Computing the structure" means computing a complete set of generators and relations, but even computing the order of the group is a difficult task in general. The best known algorithm is the *index-calculus* algorithm [HM89], running in subexponential expected time $L_{|\operatorname{disc}(\mathcal{O})|}(1/2)$. For security, the discriminant must be chosen large enough for the cost of solving $\mathcal{O}$-Vect to be prohibitively high; the fastest known *quantum* algorithm itself has complexity about $L_{|\operatorname{disc}(\mathcal{O})|}(1/2)$ (Theorem 2.20). Therefore, at first glance, it looks like (quantumly) breaking the security has about the same cost as computing the structure $\mathrm{Cl}(\mathcal{O})$. In other words, it seems hard to have the two following properties simultaneously:

- the $\mathcal{O}$-Vect problem is hard (even for a quantum computer), and
- the structure of $\mathrm{Cl}(\mathcal{O})$ is known.

This account of the situation is, of course, not fully accurate. The hidden constants in both $L_{|\operatorname{disc}(\mathcal{O})|}(1/2)$ complexities are not the same, and it appears that computing the class group is substantially faster than the quantum algorithm for vectorization (and, of course, has the advantage of not requiring a quantum computer). The CSI-FiSh scheme [BKV19] demonstrated that fact: they computed the structure of the class group for the first security level of CSIDH (namely CSIDH-512, where $|\operatorname{disc}(\mathcal{O})|$ is a 512-bit integer). The computation took an estimated effort of 52 core years. Applying this approach to higher security levels is believed to be infeasible.

2.5.1. SCALLOP. In ⟦DFK+23⟧ we explore another route and introduce SCALLOP, a scalable framework for the action of class groups with known structure. We observe that the case of CSIDH is quite unfavorable: the corresponding order $\mathcal{O} = \mathbf{Z}[\pi] = \mathbf{Z}[\sqrt{-p}]$ is a maximal order (or of index 2 in a maximal order). For such orders, the index-calculus algorithm is the best we can do, and we are stuck with the complexity $L_{|\operatorname{disc}(\mathcal{O})|}(1/2)$. The class group of non-maximal orders can be easier to compute; and in interesting cases, the corresponding vectorization problem remains hard. In SCALLOP, we consider the class group of a quadratic order of large prime conductor inside an imaginary quadratic field of small discriminant. This family of quadratic orders lets us easily determine the size of the class group, and, by carefully choosing the conductor, even exercise significant control on it. We analyse the security of this construction, and show how to efficiently represent the corresponding orientations.

One of the motivations was to be able to evaluate the action of arbitrary ideals — at a time where no polynomial-time algorithm was known. SCALLOP was the first to achieve

that goal for a security level equivalent to CSIDH-1024, a parameter firmly out of reach of the index-calculus approach of CSI-FiSh. Asymptotically, the evaluation algorithm described in SCALLOP is still subexponential. Today, that motivation is challenged by the polynomial time algorithm of Clapotis [PR23] to evaluate the action of arbitrary ideals (Clapotis is asymptotically more efficient, but its practicality is still under investigation). Still, the approach of SCALLOP remains the only framework in which the structure of the acting group can be computed in polynomial time. It even unlocks an action for which the discrete logarithm problem on the acting group is easy (it would be easy for a quantum adversary anyway, but this power can now be given to the good guys, for constructive applications).

2.5.2. Class groups of non-maximal orders. As mentioned in the previous section, non-maximal orders can have easy-to-compute class groups. For the moment, assume that the size of the class group $\mathrm{Cl}(\mathcal{O})$ is an accurate metric for the hardness of $\mathcal{O}$-Vect. We simply wish to find large class groups whose structure can be computed efficiently.

Consider an imaginary quadratic number field K, with maximal order $\mathcal{O}_K$. Recall that any order in K is of the form $\mathcal{O} = \mathbf{Z} + f\mathcal{O}_K$, where $f \in \mathbf{Z}_{>0}$ is the conductor. We start from two observations:

(1) The class group $\mathrm{Cl}(\mathcal{O})$ is a well-understood combination of $\mathrm{Cl}(\mathcal{O}_K)$ and other simpler groups.
(2) The size of the class group $\mathrm{Cl}(\mathcal{O})$ grows linearly with the conductor f.

Therefore, it is possible to construct large class groups $\mathrm{Cl}(\mathcal{O})$ as follows: consider a field of small discriminant (so the class group $\mathrm{Cl}(\mathcal{O}_K)$ is easy to compute), and let $\mathcal{O} = \mathbf{Z} + f\mathcal{O}_K$ be an order with large conductor f. The structure of $\mathrm{Cl}(\mathcal{O})$ can be deduced from $\mathrm{Cl}(\mathcal{O}_K)$ thanks to the classical exact sequence

$$1 \to \mathcal{O}_K^\times/\mathcal{O}^\times \to (\mathcal{O}_K/f\mathcal{O}_K)^\times/(\mathcal{O}/f\mathcal{O}_K)^\times \to \mathrm{Cl}(\mathcal{O}) \to \mathrm{Cl}(\mathcal{O}_K) \to 1.$$

Let us look at each group in that sequence. The group $\mathcal{O}_K^\times/\mathcal{O}^\times$ is the simplest. Indeed, for all but two imaginary quadratic number fields, we have $\mathcal{O}_K^\times = \{\pm 1\}$, hence $\mathcal{O}_K^\times/\mathcal{O}^\times$ is trivial. The two exceptions are the Gaussian integers and the Eisenstein integers, for which $\mathcal{O}_K^\times/\mathcal{O}^\times$ has order at most 3.

To compute the group $\mathrm{Cl}(\mathcal{O}_K)$, we choose a field K of small discriminant. In SCALLOP, we push this choice to the extreme, choosing a field for which $\mathrm{Cl}(\mathcal{O}_K)$ is trivial, like $\mathbf{Q}(\sqrt{-1})$.

To obtain the structure of $\mathrm{Cl}(\mathcal{O})$, it remains to compute $(\mathcal{O}_K/f\mathcal{O}_K)^\times/(\mathcal{O}/f\mathcal{O}_K)^\times$. It can be computed efficiently from the factorization of f. For SCALLOP, we only care about the case where f is a large (hence unramified) prime number — for security reasons, see Section 2.5.3. When f is an unramified prime, there is an explicit isomorphism

$$(\mathcal{O}_K/f\mathcal{O}_K)^\times/(\mathcal{O}/f\mathcal{O}_K)^\times \cong \begin{cases} \mathbf{F}_f^\times \text{ if } f \text{ splits in } K, \text{ or} \\ \mathbf{F}_{f^2}^\times/\mathbf{F}_f^\times \text{ if } f \text{ is inert in } K. \end{cases}$$

Example 2.27. Let $K = \mathbf{Q}(\sqrt{-1})$ and $\mathcal{O}_K = \mathbf{Z}[\sqrt{-1}]$. Then, $\mathcal{O}_K^\times/\mathcal{O}^\times$ has order 2, and $\mathrm{Cl}(\mathcal{O}_K)$ is trivial. Let f be a prime number such that $f \equiv 1 \bmod 4$, and let $\mathcal{O} = \mathbf{Z} + f\mathcal{O}_K$. Then, ℓ is inert in K, hence $(\mathcal{O}_K/f\mathcal{O}_K)^\times/(\mathcal{O}/f\mathcal{O}_K)^\times \cong \mathbf{F}_f^\times$. Through this isomorphism, the group $\mathcal{O}_K^\times/\mathcal{O}^\times$ maps to $\{\pm 1\}$ (the only subgroup or order 2). We deduce that

$$\mathrm{Cl}(\mathcal{O}) \cong \mathbf{F}_f^\times/\{\pm 1\} \cong \mathbf{Z}/\left(\frac{f-1}{2}\right)\mathbf{Z}.$$

This is it: we have an order $\mathcal{O}$ with arbitrarily large class group $\mathrm{Cl}(\mathcal{O})$, and an explicit description of the "structure" of $\mathrm{Cl}(\mathcal{O})$. It is cyclic of order $(f-1)/2$. Now, one could ask for more, like being able to compute discrete logarithms in $\mathrm{Cl}(\mathcal{O})$. In general, solving

discrete logarithms in $\mathbf{F}_f^\times$ cannot be done in polynomial time. But if it has smooth order, it is easy. One could craft the prime conductor f such that $f-1$ is smooth (say, $f = 2^n + 1$).

Once a suitable order $\mathcal{O}$ has been found, it remains to find an initial $\mathcal{O}$-oriented curve $(E, \iota) \in \mathcal{E}\ell\ell_p(\mathcal{O})$. This can be done in polynomial time via Theorem 2.19. A large part of the article [[DFK+23]] is concerned with making these theoretical algorithms as practical as possible. These considerations have since largely been surpassed, for instance by SCALLOP-HD [CLP23]. Leveraging the isogeny interpolation algorithm (Theorem 1.29), this "HD" version is both simpler and faster than the original construction.

2.5.3. Security analysis. Does replacing maximal orders (as in CSIDH) with non-maximal orders (in SCALLOP) jeopardize the security? Theorem 2.23 makes one thing clear: the conductor of the order should not be smooth, otherwise an attacker could "ascend the volcano", reducing the security to a case where the discriminant is small. Precursors of that theorem motivated the choice in SCALLOP for the conductor to be one large prime f. With this precaution, the fastest known attacks are essentially the meet-in-the-middle (classical) and Kuperberg (quantum) algorithms, analyzed in Section 2.4.

In [[DFK+23]], we discuss another conceivable attack strategy. Let us focus on the $\mathcal{O}$-EndRing problem (which is essentially equivalent to $\mathcal{O}$-Vect, thanks to Theorem 2.18). We are give some $(E_1, \iota_1) \in \mathcal{E}\ell\ell_p(\mathcal{O})$, and wish to compute $\mathrm{End}(E_1)$. In SCALLOP, the order $\mathcal{O}$ is of the form $\mathbf{Z} + f\mathcal{O}_K$ where the maximal order $\mathcal{O}_K$ has trivial class group. One can compute the (essentially) unique $(E_0, \iota_0) \in \mathcal{E}\ell\ell_p(\mathcal{O}_K)$ together with its endomorphism ring $\mathrm{End}(E_0)$. There exists a unique descending isogeny

$$\varphi : (E_0, \iota_0) \longrightarrow (E_1, \iota_1),$$

which has degree f. To compute $\mathrm{End}(E_1)$, one could try the following:

(1) Design an algorithm to find an efficient representation of φ.
(2) Using this representation, convert φ to its corresponding left $\mathrm{End}(E_0)$-ideal I_φ.
(3) Deduce $\mathrm{End}(E_1)$ as the right-order of I_φ.

Step 3 can be done with the methods presented in Section 1.3. Step 2 is already more challenging, but this "isogeny to ideal" problem for large prime degree has recently been solved in quantum polynomial time [CII+23]. Now, the security of SCALLOP entirely hangs on Step 1. The isogeny φ is uniquely determined by all the information available to an attacker, but there is no known method to find it, other than solving $\mathcal{O}$-EndRing in the first place with generic methods.

3

Ideal lattices

In this chapter, we present contributions related to ideal lattices, and their applications in cryptography and computational number theory. We study random walks in the space of ideal lattices up to isometry: the Arakelov class group. These random walks are a powerful tool to study the hardness of computational problems in ideal lattices, and to design rigorous algorithms for algebraic number theory.

This chapter is built around the presentation of the articles (in order of appearance):

[[BDPW20]] Koen de Boer, Léo Ducas, Alice Pellet-Mary, and Benjamin Wesolowski. Random self-reducibility of Ideal-SVP via Arakelov random walks. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 243–273. Springer, 2020.

[[FPSW23]] Joël Felderhoff, Alice Pellet-Mary, Damien Stehlé, and Benjamin Wesolowski. Ideal-SVP is hard for small-norm uniform prime ideals. In Guy N. Rothblum and Hoeteck Wee, editors, *Theory of Cryptography – 21st International Conference, TCC 2023*, volume 14372 of *Lecture Notes in Computer Science*, pages 63–92. Springer, 2023.

[[BPW24]] Koen de Boer, Alice Pellet-Mary, and Benjamin Wesolowski. Rigorous methods for computational number theory. Preprint available on demand, 2024.

Part of the results of [[BPW24]] also appeared in the PhD dissertation of de Boer [Boe22].

3.1. Introduction

A lattice is a discrete subgroup in a Euclidean vector space. A lattice Λ is generally represented by a *basis*: a collection $(b_1, \dots, b_m)$ of linearly independent vectors with $\Lambda = b_1\mathbf{Z} + \cdots + b_m\mathbf{Z}$. The integer m is the *rank* of the lattice. As a group, the lattice is isomorphic to $\mathbf{Z}^m$. The Euclidean norm on the ambient space makes for a much richer playground, inducing a notion of length on the vectors of the lattice. One can then ask geometric questions (what is the shortest possible length of a non-zero vector in the lattice?), or their computational counterparts (can one find a non-zero lattice vector with shortest possible length?).

Lattices have been a standard object in number theory since Minkowski's seminal work *Geometrie der Zahlen* published in 1896. This number-theoretic perspective on lattices

came to be known as the field of *geometry of numbers* in reference to this work. Most famously, Minkowski proved that any lattice contains a vector of relatively short norm (*Minkowski's bound* on the shortest vector). Lattices appear naturally in algebraic number theory: the ring of integers of a number field is a lattice, and so is any (fractional) ideal. We call them *ideal lattices*. Considering ideals as lattices and applying Minkowski's bound is a classical approach to prove one of the most fundamental theorems in algebraic number theory: the finiteness of the class group.

The *shortest vector problem* (SVP) is the following computational problem: given a lattice Λ (described by a basis), find a non-zero vector in Λ with shortest norm. The shortest vector problem (or its approximated version, Approx-SVP) is a central hard problem in complexity theory. It is presumed to be hard even for quantum algorithms, and thanks to the worst-case to average-case reductions of Ajtai [Ajt99] and Regev [Reg09], it has become the theoretical foundation for many post-quantum cryptographic constructions.

The idea is the following: the private key would be a "good" basis of a lattice Λ, consisting of n *short* vectors, and the public key would be a "bad" basis of the same lattice, from which it is hard to recover short vectors. The public basis, while "bad", still enables some simple operations: sampling random lattice points (far from the origin), or encoding a message as a lattice point $P \in \Lambda$. Adding a small "error" ε to this point (i.e., shifting it to obtain a close point outside the lattice), only the secret "good" basis allows to recover the original point P from the noisy point $P+\varepsilon$. Hence the security of lattice-based cryptography relies on the presumed hardness of finding short vectors (to recover a short basis from a bad one), and on closely related problems such as *learning with errors* (LWE) and *short integer solution* (SIS).

The main disadvantage of such plain lattice-based cryptosystems is their heavy memory and bandwidth footprint: the public and secret keys both are a lattice basis, an $n \times n$ matrix, where the dimension n is in the order of hundreds. This issue can be addressed by using lattices with more structure. Ideal lattices (or the more general notion of *module lattices*) thereby took a central place in lattice-based cryptography. The shortest vector problem in ideal lattices is the Id-SVP problem. The Ring-SIS [MicO7, LMO6, PRO6] and Ring-LWE [SSTX09, LPR13, PRS17] problems were introduced and shown to be at least as hard as worst-case instances of Id-SVP. Both Ring-SIS and Ring-LWE have proved very versatile problems for building efficient cryptographic schemes.

The additional algebraic structure, while practically useful to build efficient cryptosystems, also opens more cryptanalytic avenues. How hard is Id-SVP? While still far from being "solved", we do know much better algorithms for some versions of Id-SVP than for the unstructured case [CGS14, CDPR16],⟦CDW17, CDW21⟧. Id-SVP has proved a fertile cryptanalytic playground to test the impact of adding "structure" to lattice problems.

This chapter revolves around the development of a new tool and its consequences in cryptography and in computational number theory: random walks in the space of ideal lattices. The rapid-equidistribution properties of these random walks bring new insights on the (average) hardness of computational problems in ideal lattices, and unlock new algorithms for fundamental problems in computational number theory.

3.1.1. Contributions and organisation of the chapter. The chapter is organized as follows.

Ideal lattices and the Arakelov class group. We open this chapter with a few preliminaries on ideal lattices and the Arakelov class group in Section 3.2. After fixing some notation and recalling the main definitions, we explore the classical connection between the two notions: the Arakelov class group can be viewed as the group of all ideal lattices up to isometry.

This connection, long known by number theorists [Bay99, Sch08], had been absent from the cryptographic literature until the article ⟦BDPW20⟧. Remarkably, the Arakelov class group is a combination of two groups that had already led to significant cryptanalytic advances: the unit torus [CGS14, CDPR16] and the class group ⟦CDW17, CDW21⟧.

Random walks in the Arakelov class group. In Section 3.3, we present the main result of our article ⟦BDPW20⟧: random walks in the Arakelov class group equidistribute rapidly, assuming the Generalized Riemann Hypothesis. Contrary to the random walks in previous chapters, we are now walking in an infinite, continuous space. We thus consider random walks which combine two kinds of steps: "continuous" steps, akin to a Brownian motion, and "discrete" steps, capable of jumping between connected components.

Average hardness of ideal lattices. The motivation for the equidistribution theorem is the following. Elements of the Arakelov class group are, essentially, ideal lattices. From a starting ideal lattice, a random walk rapidly reaches a uniformly random ideal lattice. Through a (short) random walk, one can transfer a solution of Id-SVP (i.e., a short vector) from the target to the source, with a small loss on the approximation factor. In other words: if one can solve Id-SVP for uniformly random ideal lattices, then one can solve Id-SVP for any ideal lattice. It is a worst-case to average-case reduction.

We explore this idea in Section 3.4. In the article ⟦BDPW20⟧, we follow this strategy to obtain a worst-case to average-case reduction where "average" means uniform in the Arakelov class group (in the sense of the Haar measure). There is a critical difficulty: the Arakelov class group being continuous, this computational result requires rounding, resulting in ideals of rather large norm.

We refine this result in the article ⟦FPSW23⟧. Combining the Arakelov random walks with reductions of Gentry [Gen09, Gen10], we obtain a worst-case to average-case reduction where "average" means *uniformly random prime ideal of small norm*. Using the reduction from Pellet-Mary and Stehlé [PS21], this notably leads to the first distribution for instances of the NTRU cryptosystem with a polynomial modulus whose hardness is supported by a worst-case lattice problem.

A rigorous tool for algorithmic number theory. Long before finding a place in cryptography, ideal lattices have been a central object of interest in number theory. They naturally arise in many computational number theoretic questions, like the computation of the class group and unit group of a number field, or perhaps more surprisingly, the factorisation of integers.

Many algorithms for these fundamental problems (like the computation of class groups in subexponential time) are not fully understood: their analysis resorts to heuristic assumptions. This persistent need for heuristic assumptions often stems from a step of this form: given an ideal class $[\mathfrak{a}]$ of a number field, find a representative $\mathfrak{b} \in [\mathfrak{a}]$ belonging to a particular family $\mathcal{S}$ of ideals (for instance, the family of prime ideals, or smooth ideals). It is relatively simple to design an algorithm for this task: sample a random $\mathfrak{b} \in [\mathfrak{a}]$, and hope that it belongs to the desired family $\mathcal{S}$. One then heuristically argues that the probability that $\mathfrak{b} \in \mathcal{S}$ should be proportional to the density of $\mathcal{S}$. For instance, the subexponential density of smooth ideals heuristically implies that one can find smooth representatives in subexponential time. This is the heart of state-of-the-art algorithms to compute class groups, unit groups, or generators of principal ideals in number fields [BF14, Buc88, LL93], and has long constituted a theoretical obstacle overcome only by heuristic arguments (with the exception of quadratic fields [HM89]).

In Section 3.5, we present the result of the article ⟦BPW24⟧: a general strategy to solve these ideal sampling tasks rigorously and efficiently. We remove the need for heuristic assumptions by leveraging the rigorous randomization properties of walks in the Arakelov

class group. We illustrate the power of this technique by presenting the first algorithm for computing class groups and unit groups of arbitrary number fields that provably runs in probabilistic subexponential time.

3.2. Ideal lattices and the Arakelov class group

In this section, we define the notions of ideal lattice and Arakelov class groups, and explain their connection. A more detailed account can be found in the main inspiration for this section: the article [Sch08].

3.2.1. Ideal lattices. Fix a number field K of degree $n = [K : \mathbf{Q}]$ and discriminant Δ_K. The field K is a vector space of dimension n over $\mathbf{Q}$, and it has a canonical $\mathbf{R}$-valued inner product. In algebraic terms, the inner product is defined as $\langle \alpha, \beta \rangle = \mathrm{Tr}(\alpha\beta^*)$ where $-^*$ is the canonical involution of the étale $\mathbf{R}$-algebra $K \otimes \mathbf{R}$. The Minkowski embedding provides a more explicit description as follows. The number field K has n field embeddings into $\mathbf{C}$, which are divided into $n_{\mathbf{R}}$ real embeddings and $n_{\mathbf{C}}$ conjugate pairs of complex embeddings, with $n = n_{\mathbf{R}} + 2n_{\mathbf{C}}$. Let

$$K_{\mathbf{R}} = \left\{ (x_\sigma)_\sigma \in \bigoplus_{\sigma : K \to \mathbf{C}} \mathbf{C} \;\middle|\; \overline{x_\sigma} = x_{\overline{\sigma}} \right\} \cong K \otimes \mathbf{R},$$

where the sum is over all field embeddings $\sigma : K \to \mathbf{C}$. We consider the Euclidean norm $\|(x_\sigma)_\sigma\| = \left(\sum_\sigma |x_\sigma|^2\right)^{1/2}$ on $K_{\mathbf{R}}$. The Minkowski embedding is the map $\Psi : K \to K_{\mathbf{R}} : \alpha \mapsto (\sigma(\alpha))_\sigma$. The field K inherits the Euclidean structure of $K_{\mathbf{R}}$ through this embedding. More explicitly, for $\alpha \in K$, we have $\|\alpha\|^2 = \sum_\sigma |\sigma(\alpha)|^2$. Abusing notation, we treat Ψ simply as an inclusion $K \subset K_{\mathbf{R}}$.

Let $\mathcal{O}_K$ be the ring of integers of K. It is a discrete additive subgroup of rank n, hence a lattice in the vector space K. A fractional ideal in K is any subset of K of the form $\alpha\mathfrak{a}$ where $\alpha \in K^\times$ and $\mathfrak{a}$ a non-zero ideal in $\mathcal{O}_K$ We denote by $\mathscr{I}_K$ the group of fractional ideals of K. Any fractional ideal is also a lattice in K. This is essentially what we define as an ideal lattice in K, with one last generalization: we extend the scalars to $\mathbf{R}$, leading to the following definition.

Definition 3.1. An *ideal lattice* over K is a lattice in $K_{\mathbf{R}}$ of the form $x\mathfrak{a}$, where $x \in K_{\mathbf{R}}^\times$ and $\mathfrak{a}$ is an ideal in $\mathcal{O}_K \subset K \subset K_{\mathbf{R}}$. Equivalently, an ideal lattice over K is a lattice in $K_{\mathbf{R}}$ of rank n which is also an $\mathcal{O}_K$-submodule.

We denote by IdLat_K the set of all ideal lattice over K. Note that it forms an abelian group, with $x\mathfrak{a} \cdot y\mathfrak{b} = xy\mathfrak{a}\mathfrak{b}$, and neutral element $\mathcal{O}_K$. The (co)volume of an ideal lattice is

$$\mathrm{Vol}(x\mathfrak{a}) = |\Delta_K|^{1/2} N(\mathfrak{a}) \prod_\sigma |x_\sigma|.$$

Our problems of interest are invariant under rescaling. We thus define the subgroup

$$\mathrm{IdLat}_K^0 = \{ L \in \mathrm{IdLat}_K \,|\, \mathrm{Vol}(L) = \mathrm{Vol}(\mathcal{O}_K) \}.$$

Definition 3.2. An *isometry* between two ideal lattices $L_1, L_2 \subset K_{\mathbf{R}}$ is an isomorphism $\varphi : L_1 \to L_2$ of $\mathcal{O}_K$-modules such that $\|\varphi(z)\| = \|z\|$ for all $z \in L_1$. Equivalently, it is a map of the form $\varphi_\xi : z \mapsto \xi z$ for some $\xi = (\xi_\sigma)_\sigma \in K_{\mathbf{R}}$ with $|\xi_\sigma| = 1$ for all σ. If two ideal lattices L_1 and L_2 are isometric, we write $L_1 \cong L_2$.

The set $\mathrm{Iso}_K = \{ L \in \mathrm{IdLat}_K \mid L \cong \mathcal{O}_K \}$ is a subgroup of IdLat_K^0, and the quotient

$$\mathrm{IdLat}_K^0 / \mathrm{Iso}_K = \mathrm{IdLat}_K^0 / \cong$$

forms the group of (normalised) *ideal lattices up to isometry*. This is *almost* the Arakelov class group: the latter is defined through a different formalism, but the result is isomorphic.

3.2.2. The Arakelov class group. In this section, we define the Arakelov class group Pic^0_K, and outline the proof that it is isomorphic to $\mathrm{IdLat}^0_K / \cong$, the group of ideal lattices up to isometry.

A *finite place* of K is a (non-zero) prime ideal in $\mathcal{O}_K$. An *infinite place* of K is a field embedding $K \to \mathbf{C}$ up to complex conjugation; in other words, it is either a real embedding $\nu : K \to \mathbf{R}$, or a pair $\nu = \{\sigma, \overline{\sigma}\}$ of complex embeddings. Given an embedding σ, we write ν_σ for the corresponding place; reciprocally, for any place ν, we let σ_ν be one of the corresponding embeddings. There are exactly $n_{\mathbf{R}} + n_{\mathbf{C}} \leq n$ infinite places. The group of *Arakelov divisors* is

$$\mathrm{Div}_K = \bigoplus_{\mathfrak{p}} \mathbf{Z} \times \bigoplus_{\nu} \mathbf{R}.$$

We denote the canonical basis elements with the symbols $(\!|\mathfrak{p}|\!)$ and $(\!|\nu|\!)$ (the divisor with value 1 at $\mathfrak{p}$ or ν respectively, and 0 everywhere else). Then, an arbitrary divisor can be written as

$$\mathbf{a} = \sum_{\mathfrak{p}} n_{\mathfrak{p}} \cdot (\!|\mathfrak{p}|\!) + \sum_{\nu} x_\nu \cdot (\!|\nu|\!).$$

The *degree* of an Arakelov divisor is given by the group homomorphism

$$\begin{aligned} \deg : \mathrm{Div}_K &\longrightarrow \mathbf{R} \\ \sum_{\mathfrak{p}} n_{\mathfrak{p}} \cdot (\!|\mathfrak{p}|\!) + \sum_{\nu} x_\nu \cdot (\!|\nu|\!) &\longmapsto \sum_{\mathfrak{p}} n_{\mathfrak{p}} \log(N(\mathfrak{p})) + \sum_{\nu \text{ real}} x_\nu + \sum_{\nu \text{ complex}} 2x_\nu. \end{aligned}$$

The kernel of this map is the group Div^0_K of degree-zero divisors.

Given any element $\alpha \in K^\times$, one can construct an Arakelov divisor via the map

$$\mathrm{div} : K^\times \longmapsto \mathrm{Div}_K : \alpha \longmapsto \sum_{\mathfrak{p}} \mathrm{ord}_{\mathfrak{p}}(\alpha) \cdot (\!|\mathfrak{p}|\!) - \sum_{\nu} \log |\sigma_\nu(\alpha)| \cdot (\!|\nu|\!).$$

Divisors of the form $\mathrm{div}(\alpha)$ are called *principal divisors*, and the *product formula* states that principal divisors have degree zero, i.e., $\mathrm{div}(K^\times) \subset \mathrm{Div}^0_K$.

Definition 3.3. The *Arakelov class group* of K is the group $\mathrm{Pic}^0_K = \mathrm{Div}^0_K / \mathrm{div}(K^\times)$. For any Arakelov divisor $\mathbf{a}$, we denote by $[\mathbf{a}]$ its class in Pic^0_K.

3.2.3. Arakelov divisors and ideal lattices. In this section, we explain the connection between Arakelov divisors and ideal lattices, culminating in the following theorem.

Theorem 3.4. *The "exponential map" induces an isomorphism* $\mathrm{Pic}^0_K \to \mathrm{IdLat}^0_K / \cong$.

In other words, the Arakelov class group encodes the collection of ideal lattices up to isomorphism and rescaling. As the "exponential" terminology suggests, the Arakelov class group can informally be thought of as a "logarithmic space" for ideal lattices.

The connection between Arakelov divisors and ideal lattices $L = x\mathfrak{a}$ (where $x \in K_{\mathbf{R}}^\times$ and $\mathfrak{a}$ is a (fractional) ideal) can be understood by separating the "finite" part and the "infinite" part. First, the "finite" part $\bigoplus_{\mathfrak{p}} \mathbf{Z}$ accounts for the $\mathfrak{a}$-component: the unique factorisation of fractional ideals into prime ideals is a bijection between $\bigoplus_{\mathfrak{p}} \mathbf{Z}$ and the collection of fractional ideals. We thus have a bijection between divisors and fractional ideals:

$$\mathrm{Exp}_{\mathrm{fin}} : \bigoplus_{\mathfrak{p}} \mathbf{Z} \longrightarrow \mathscr{I}_K : \sum_{\mathfrak{p}} n_{\mathfrak{p}} \cdot (\!|\mathfrak{p}|\!) \longmapsto \prod_{\mathfrak{p}} \mathfrak{p}^{n_{\mathfrak{p}}}.$$

Its inverse is the factorisation map

$$\mathrm{Log}_{\mathrm{fin}} : \mathscr{I}_K \longrightarrow \bigoplus_{\mathfrak{p}} \mathbf{Z} : \mathfrak{a} \longmapsto \sum_{\mathfrak{p}} \mathrm{ord}_{\mathfrak{p}}(\mathfrak{a}) \cdot (\!|\mathfrak{p}|\!).$$

Second, the "infinite" part $\bigoplus_\nu \mathbf{R}$ somehow accounts for the continuous part of an ideal lattice: the x-component. That may not be as clear: how does $\bigoplus_\nu \mathbf{R}$ compare with $K_{\mathbf{R}}^\times$? We at least have an injective group homomorphism

$$\mathrm{Exp}_\infty : \bigoplus_\nu \mathbf{R} \longrightarrow K_{\mathbf{R}}^\times : \sum_\nu x_\nu \cdot (\!|\nu|\!) \longmapsto (e^{x_{\nu\sigma}})_\sigma.$$

It is not surjective, so does not have an inverse, but it does have a retraction:

$$\mathrm{Log}_\infty : K_{\mathbf{R}}^\times \longrightarrow \bigoplus_\nu \mathbf{R} : (y_\sigma)_\sigma \longmapsto \sum_\nu \log |y_{\sigma_\nu}| \cdot (\!|\nu|\!).$$

Remark 3.5. The composition of the inclusion $K \to K_{\mathbf{R}}$ with Log_∞ is the map $\mathrm{Log} : K^\times \to \bigoplus_\nu \mathbf{R}$ often referred to as the *Logarithmic embedding* of K, which plays an important role in the cryptanalysis of ideal lattices [CGS14, CDPR16], [[CDW21]].

Regrouping the "finite" and "infinite" parts, we obtain a map

$$\mathrm{Exp} = \mathrm{Exp}_{\mathrm{fin}} \cdot \mathrm{Exp}_\infty : \mathrm{Div}_K \longrightarrow \mathrm{IdLat}_K .$$

It is not surjective, because Exp_∞ is not. However, the induced map to the quotient $\mathrm{IdLat}_K / \cong$ is surjective. Indeed, for any ideal lattice $x\mathfrak{a}$, we have an isomorphism $x\mathfrak{a} \to (\xi x)\mathfrak{a} : z \mapsto \xi x$ where $\xi = (|x_\sigma|/x_\sigma)_\sigma$, and the element $\xi x = (|x_\sigma|)_\sigma \in K_{\mathbf{R}}^\times$ is in the image of Exp_∞. In particular, $\mathrm{IdLat}_K / \cong$ is isomorphic to a quotient of Div_K. It can be verified that the kernel of Exp is the group $\mathrm{div}(K^\times)$ of principal divisors, hence $\mathrm{Div}_K / \mathrm{div}(K^\times) \cong \mathrm{IdLat}_K / \cong$. The last step to obtain Theorem 3.4 is to prove that the subgroup of normalized ideal lattices IdLat_K^0 corresponds to the subgroup of degree-zero divisors Div_K^0. This follows from the formula

$$\mathrm{Vol}(\mathrm{Exp}(-)) = \mathrm{Vol}(\mathcal{O}_K) e^{\deg(-)}.$$

In particular, the exponential map restricts and co-restricts to

$$\mathrm{Exp}^0 : \mathrm{Div}_K^0 \longrightarrow \mathrm{IdLat}_K^0,$$

and we obtain Theorem 3.4.

Remark 3.6. Note that if one wishes to work with ideal lattices up to rescaling, but *not* up to isomorphism, one could consider the *oriented* Arakelov class group [Sch08].

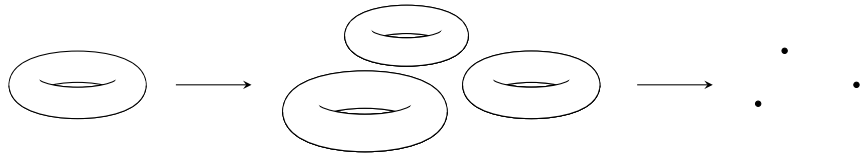
3.2.4. Structure of the Arakelov class group. The Arakelov class group Pic_K^0 provides a convenient formalism to study the space of ideal lattices (up to isomorphism and scaling): it is naturally a compact topological group. In this section, we discuss the structure of this group, summarized in the short exact sequence

$$0 \longrightarrow T_K \longrightarrow \mathrm{Pic}_K^0 \longrightarrow \mathrm{Cl}(K) \longrightarrow 0, \tag{3.1}$$

where $\mathrm{Cl}(K)$ is the class group of K, and T_K is a torus (the quotient of a real vector space by a lattice), the so-called *Log-unit torus* of K. Geometrically, the Arakelov class group is thus a finite union of copies of the torus T_K: one copy for each ideal class. We illustrate this structure in Figure 3.1.

Let us discuss this short exact sequence in greater detail. The map $\mathrm{Pic}_K^0 \longrightarrow \mathrm{Cl}(K)$ is the most straightforward. Consider the natural morphism

$$\begin{aligned} \mathrm{Div}_K^0 &\longrightarrow \mathscr{I}_K \\ \sum_{\mathfrak{p}} n_{\mathfrak{p}} \cdot (\!|\mathfrak{p}|\!) + \sum_\nu x_\nu \cdot (\!|\nu|\!) &\longmapsto \prod_{\mathfrak{p}} \mathfrak{p}^{n_{\mathfrak{p}}}. \end{aligned}$$



Log-unit torus T Arakelov class group Pic^0_K Class group $\mathrm{Cl}(K)$

FIGURE 3.1. Schematic representation of the Arakelov class group, a combination of the class group and the Log-unit torus.

Evidently, principal divisors are sent to principal ideals, thereby inducing the morphism $\mathrm{Pic}^0_K \to \mathrm{Cl}(K)$ in the second half of the sequence (3.1).

It remains to understand the first half of (3.1). The Log-unit torus T_K is defined as follows. Consider the real vector space

$$H = \left\{ (x_\nu)_\nu \in \bigoplus_\nu \mathbf{R} \;\middle|\; \sum_{\nu \text{ real}} x_\nu + \sum_{\nu \text{ complex}} 2x_\nu = 0 \right\} \subset \bigoplus_\nu \mathbf{R}.$$

This vector space H is generated by the image of $\mathcal{O}_K^\times$ through the logarithmic embedding $\mathrm{Log} : K^\times \to \bigoplus_\nu \mathbf{R}$ (see Remark 3.5), and $\mathrm{Log}(\mathcal{O}_K^\times)$ is a full-rank lattice in H. The Log-unit torus is the quotient $T_K = H/\mathrm{Log}(\mathcal{O}_K^\times)$. The map $T_K \to \mathrm{Pic}^0_K$ in the exact sequence (3.1) is induced by the obvious map

$$H \longrightarrow \mathrm{Div}^0_K : (x_\nu)_\nu \longmapsto \sum_\nu x_\nu \cdot (\!|\nu|\!).$$

In summary, we have the following commutative diagram, where each line is exact, and each vertical arrow is the natural quotient map:

$$\begin{array}{ccccccccc} 0 & \longrightarrow & H & \longrightarrow & \mathrm{Div}^0_K & \longrightarrow & \mathscr{I}_K & \longrightarrow & 0 \\ & & \downarrow & & \downarrow & & \downarrow & & \\ 0 & \longrightarrow & T_K & \longrightarrow & \mathrm{Pic}^0_K & \longrightarrow & \mathrm{Cl}_K & \longrightarrow & 0. \end{array}$$

3.3. Random walks in Arakelov class groups

This section presents the new versatile tool developed in [[BDPW20]]: random walks in the Arakelov class group, and their rapid equidistribution property. The goal is to design a process which, starting with an arbitrary ideal lattice, applies a sequence of small random modifications. Each modification, a step in the walk, affects the shape of the lattice in a well-controlled manner. The result is a random ideal lattice, closely related to the original one, but well distributed in the space of ideal lattices.

3.3.1. The random walk. Before stating the equidistribution theorem, we must define the random walks under consideration. The design of the walk is motivated by the structure of the Arakelov class group discussed in Section 3.2.4. The group is a combination of a continuous part, the unit torus T_K, and a discrete part, the class group $\mathrm{Cl}(K)$. The walk is thus a combination of a discrete walk and a continuous walk.

The random walk on the Arakelov class group Pic^0_K is best described as a walk on Div^0_K, projected to Pic^0_K. For any probability distribution $\mathcal{D}$ on Div^0_K, we write $[\mathcal{D}]$ for the corresponding quotient distribution on Pic^0_K.

The discrete walk. A discrete step is essentially a step of this form: from an arbitrary Arakelov divisor $\mathbf{a}$, sample a random prime ideal $\mathfrak{p}$, and go to $\mathbf{a} + (\!|\mathfrak{p}|\!)$. The new divisor $\mathbf{a} + (\!|\mathfrak{p}|\!)$ does not have degree 0, so we normalize it to

$$\mathbf{a} + (\!|\mathfrak{p}|\!) - \sum_\nu \frac{\log(N(\mathfrak{p}))}{n} \cdot (\!|\nu|\!).$$

We also write it as $\mathbf{a} + d_0(\mathfrak{p})$, using the map

$$d_0 : \mathscr{I}_K \longrightarrow \mathrm{Div}^0_K : \mathfrak{a} \longmapsto \sum_\mathfrak{p} \mathrm{ord}_\mathfrak{p}(\mathfrak{a}) \cdot (\!|\mathfrak{p}|\!) - \sum_\nu \frac{\log(N(\mathfrak{a}))}{n} \cdot (\!|\nu|\!).$$

One needs to fix a distribution for $\mathfrak{p}$. Fix a bound B, and let $\mathscr{P}_K(B)$ be the set of all prime ideals in $\mathcal{O}_K$ of norm at most B. We consider $\mathfrak{p}$ uniformly distributed in $\mathscr{P}_K(B)$.

Definition 3.7 (Discrete step, discrete walk)**.** Let $\mathcal{D}$ be any probability distribution on Div^0_K. We denote by $W_B(\mathcal{D})$ the distribution of $\mathbf{a} + d_0(\mathfrak{p})$, where $\mathbf{a}$ is sampled with distribution $\mathcal{D}$, and $\mathfrak{p} \in \mathscr{P}_K(B)$ is uniform. In other words, $W_B(\mathcal{D})$ is the distribution obtained by sampling from $\mathcal{D}$, then taking a *discrete step.* A *discrete walk* of length k with initial distribution $\mathcal{D}$ consists in sampling from $\mathcal{D}$ then taking k discrete steps sequentially. The endpoint of a discrete walk of length k has distribution $W_B^k(\mathcal{D})$.

On the side of ideal lattices, a discrete step consists in replacing the ideal lattice $x\mathfrak{a}$ with the random sublattice $x\mathfrak{a}\mathfrak{p} \subset x\mathfrak{a}$, and rescaling it to normalise the volume.

A motivation for this choice of step comes from previous work on random walks in class groups: the equidistribution result of Jao, Miller and Venkatesan [JMV09], presented in Section 2.4.2. They define a similar random step in the class group: from a starting ideal class $[\mathfrak{a}]$, go to the class $[\mathfrak{a}\mathfrak{p}]$ for some uniformly random prime $\mathfrak{p} \in \mathscr{P}_K(B)$. Assuming GRH, the main theorem of [JMV09] states that for some $B = O_\varepsilon((n \log \Delta_K)^{2+\varepsilon})$, applying this step k times (i.e., a length k random walk) converges to the uniform distribution on $\mathrm{Cl}(K)$ at an exponential rate in k.

In particular, we readily obtain that our discrete walk in the Arakelov class group rapidly equidistributes among the $\#\,\mathrm{Cl}(K)$ connected components. The discrete walk "jumps" between connected components, and very effectively so.

The continuous walk. Taking discrete steps from any starting point, no matter how long the walk, the distribution of the endpoint will remain discrete. To hope for a convergence to the uniform distribution in the sense of the total variation distance (i.e., convergence in $L^1(\mathrm{Pic}^0_K)$), we need to "smooth it out". This is where the continuous walk comes in, as a kind of "blurring" step.

Geometrically, a starting point $[\mathbf{a}]$ in the Arakelov class group is a point on a torus, a copy of the unit torus T_K. A continuous walk consists in replacing $[\mathbf{a}]$ with a random nearby point on the torus: we consider a Gaussian distribution centered at $[\mathbf{a}]$ and with somewhat small standard deviation.

Definition 3.8 (Continuous walk)**.** Let $\mathcal{D}$ be any probability distribution on Div^0_K. We denote by $W^s_\infty(\mathcal{D})$ the distribution of $\mathbf{a} + x$, where $\mathbf{a}$ is sampled with distribution $\mathcal{D}$, and $x \in H \subset \bigoplus_\nu \mathbf{R} \subset \mathrm{Div}^0_K$ is sampled from the Gaussian distribution on H with standard deviation s centered at the origin.

On the side of ideal lattices, this continuous walk consists in replacing an ideal lattice $x\mathfrak{a}$ with a slight deformation of it, i.e., a lattice $\delta x\mathfrak{a}$ with $\delta \in K^\times_\mathbf{R}$ close to the identity.

The Arakelov random walk. Any probability distribution on the Arakelov class group Pic_K^0 is of the form $[\mathcal{D}]$ for some distribution $\mathcal{D}$ on Div_K^0. We thus define the Arakelov random walk as follows.

Definition 3.9 (Arakelov random walk)**.** Let $B, k, s > 0$ be parameters. Let $[\mathcal{D}]$ be a probability distribution on the Arakelov class group Pic_K^0. The Arakelov random walk with initial distribution $[\mathcal{D}]$, prime bound B, length k, and standard deviation s is the following process: sample a starting point $[\mathbf{a}]$ from $[\mathcal{D}]$, then apply the continuous walk with standard deviation s, and k discrete steps with prime bound B. The distribution of the endpoint is $[W_B^k(W_\infty^s(\mathcal{D}))]$.

3.3.2. Rapid equidistribution. The main tool developed in ⟦BDPW20⟧ is the following theorem, which states that Arakelov random walks rapidly converge to the uniform distribution (for the Haar measure).

Theorem 3.10 (Simplified form of ⟦BDPW20, Theorem 3.3⟧)**.** *Let $\varepsilon > 0$ and $s > 0$. Assuming the Generalized Riemann Hypothesis, there are bounds*

$$B = \mathrm{poly}(n, \log(\Delta_K), \log\log(1/\varepsilon), \log(1/s)), \text{ and}$$
$$\kappa = \log(\Delta_K) \cdot \mathrm{poly}(\log n, \log\log(\Delta_K), \log(1/\varepsilon), \log(1/s))$$

such that for any initial distribution $[\mathcal{D}]$ on Pic_K^0, and any $k \geq \kappa$, the endpoint of the Arakelov random walk with prime bound B, length k, and standard deviation s is at total variation distance at most ε from the uniform distribution (for the Haar measure on Pic_K^0).

Remark 3.11. Instead of the discrete walk being a succession of small steps, one could take one large leap: multiplication by a single prime ideal of larger norm. One obtains a result similar to Theorem 3.10, with $\kappa = 1$ and $B = \mathrm{poly}(n^n, \Delta_K)$. More generally, a condition of the form $B^\kappa = \mathrm{poly}(n^n, \Delta_K)$ suffices for rapid equidistribution.

Sketch of the proof. The distribution of the endpoint of the random walk is $[W_B^k(W_\infty^s(\mathcal{D}))]$. It is obtained by sampling $[\mathbf{a}]$ from $[\mathcal{D}]$, then applying the continuous walk and k discrete steps with starting point $[\mathbf{a}]$. Observe that it is also the distribution of $[\mathbf{a}] + [\mathbf{b}]$ where $[\mathbf{a}]$ is sampled from $[\mathcal{D}]$, and $[\mathbf{b}]$ is sampled for the distribution $W_B^k(W_\infty^s(\mathbf{1}_0))$ (the endpoint of a random walk with starting point $[0]$). To prove that $[W_B^k(W_\infty^s(\mathcal{D}))]$ is close to uniform, it is thus sufficient to prove that $[W_B^k(W_\infty^s(\mathbf{1}_0))]$ is close to uniform.

The distribution $\mathcal{G}_s = W_\infty^s(\mathbf{1}_0)$ is the Gaussian distribution on H with standard deviation s centered at the origin.

The discrete step can be seen as an operator $W_B : L^2(\mathrm{Pic}_K^0) \to L^2(\mathrm{Pic}_K^0)$, sending any function $f : \mathrm{Pic}_K^0 \to \mathbf{C}$ to

$$W_B(f) : \mathbf{a} \longmapsto \frac{1}{\#\mathscr{P}_K(B)} \sum_{\mathfrak{p} \in \mathscr{P}_K(B)} f(\mathbf{a} - d_0(\mathfrak{p})),$$

averaging over all "incoming neighbors" of $\mathbf{a}$. The proof proceeds with a spectral analysis of W_B, and decomposing $[\mathcal{G}_s]$ as a sum of eigenfunctions. Showing that all the eigenvalues are small except for the constant eigenfunction, all terms in the decomposition of $[W_B^k(\mathcal{G}_s)]$ vanish at an exponential rate in k — except the constant term, which corresponds to the uniform distribution.

The discrete walk operator W_B is in fact a Hecke operator, whose spectral properties are well studied. It can easily be verified that the characters $\chi \in \widehat{\mathrm{Pic}_K^0}$ are eigenfunctions with eigenvalue

$$\lambda_\chi = \frac{1}{\#\mathscr{P}_K(B)} \sum_{\mathfrak{p} \in \mathscr{P}_K(B)} \overline{\chi}(d_0(\mathfrak{p})),$$

where $\overline{\chi}$ is the complex conjugate of χ. In other words, for any $\chi \in \widehat{\mathrm{Pic}_K^0}$, we have $W_B(\chi) = \lambda_\chi \chi$. In particular, the constant function $\mathbf{1}$ has eigenvalue $\lambda_{\mathbf{1}} = 1$, and all other eigenvalues satisfy $|\lambda_\chi| \leq 1$. Decomposing $[\mathcal{G}_s]$ as a sum of eigenfunctions, we get

$$[\mathcal{G}_s] = \frac{\mathbf{1}}{\mathrm{Vol}(\mathrm{Pic}_K^0)} + \sum_{\chi \in \widehat{\mathrm{Pic}_K^0}} c_\chi \chi,$$

where $c_\chi \in \mathbf{C}$ are the Fourier coefficients of $[\mathcal{G}_s]$. Applying the discrete walk operator, we get

$$[W_B^k(\mathcal{G}_s)] = \frac{\mathbf{1}}{\mathrm{Vol}(\mathrm{Pic}_K^0)} + \sum_{\chi \in \widehat{\mathrm{Pic}_K^0}} \lambda_\chi^k c_\chi \chi.$$

As soon as $|\lambda_\chi| < 1$, the terms $\lambda_\chi^k c_\chi$ vanish as k grows. But they do not vanish *uniformly*, as eigenvalues λ_χ could be arbitrarily close to 1. To proceed, we distinguish two kinds of terms according to the *analytic conductor* $\mathfrak{q}_\infty(\chi)$:

- *Low-frequency characters.* When the analytic conductor $\mathfrak{q}_\infty(\chi)$ is small, we can think of χ as a *low frequency* character. In that case, the eigenvalue λ_χ is reasonably small. Indeed, applying classical bounds from analytic number theory [IK04, Theorem 5.15], we have

$$|\lambda_\chi| \leq \frac{1}{B^{1/2}} \cdot \mathrm{poly}(n, \log(\Delta_K), \log(B), \mathfrak{q}_\infty(\chi)), \tag{3.2}$$

  which can be made arbitrarily small by choosing B large enough. We can enforce a clear, uniform gap $|\lambda_\chi| < 1/2 < 1$, and the corresponding terms $\lambda_\chi^k c_\chi \chi$ vanish rapidly and uniformly.
- *High-frequency characters.* When the analytic conductor $\mathfrak{q}_\infty(\chi)$ is large, we can think of χ as a *high frequency* character. In such a case, the above bound (3.2) is vacuous: the right-hand side could be larger than 1, while we know that $\lambda_\chi \leq 1$. This is where our choice of the Gaussian distribution $\mathcal{G}_s$ comes in handy: its Fourier coefficients c_χ vanish rapidly for large "frequencies" $\mathfrak{q}_\infty(\chi)$. In other words, while the decaying rate of $\lambda_\chi^k c_\chi \chi$ may not be so rapid in k, the coefficient c_χ is vanishingly small to begin with, when $\mathfrak{q}_\infty(\chi)$ is large.

In summary, we have

$$[W_B^k(\mathcal{G}_s)] = \underbrace{\frac{\mathbf{1}}{\mathrm{Vol}(\mathrm{Pic}_K^0)}}_{\substack{\text{The uniform} \\ \text{distribution}}} + \underbrace{\sum_{\substack{\chi \text{ of low} \\ \text{frequency}}} \lambda_\chi^k c_\chi \chi}_{\text{Decays rapidly with } k} + \underbrace{\sum_{\substack{\chi \text{ of high} \\ \text{frequency}}} \lambda_\chi^k c_\chi \chi}_{\substack{\text{Small from the start,} \\ \text{and cannot increase}}},$$

which means that $[W_B^k(\mathcal{G}_s)]$ rapidly approaches the uniform distribution. □

3.4. Average hardness of ideal lattices

Our original motivation for the study of random walks in the Arakelov class group was to study the average hardness of computational problems in ideal lattices. We consider the following version of the shortest vector problem for ideal lattices.

Problem 3.12 (Id-HSVP$_\gamma$)**.** The *ideal Hermite shortest vector problem with approximation factor* γ is the following computational problem. Given a number field K of degree n and a fractional ideal lattice Λ, find $x \in \Lambda$ such that $0 < \|x\| \leq \gamma \cdot \mathrm{Vol}(\Lambda)^{1/n}$.

Remark 3.13. We restrict the problem to *fractional* ideal lattices (i.e., fractional ideals seen as ideal lattices through the Minkowski embedding) to avoid dealing with real numbers in the input of the problem. Any ideal lattice can be approximated to arbitrary precision by

a fractional ideal (fractional ideals are dense in IdLat_K), and fractional ideals are much easier to manipulate. We consider them to be represented by a basis in Hermite Normal Form (with respect to a well-chosen basis of K).

An *average-case* version of this problem requires a probability distribution on instances of Id-HSVP$_\gamma$. At this point of the chapter, the reader may suspect a natural candidate: the uniform distribution for the Haar measure. For computational purposes, we would rather have a discrete distribution supported on fractional ideal lattices (or even integral — we can always rescale a fractional ideal to an integral one). For complexity-theoretic or cryptographic applications, one looks for distributions with certain valuable properties, like being efficiently sampleable, easy to analyse, or supported on integral ideals of "small" norm. Three distributions for Id-HSVP$_\gamma$ instances have been considered.

(1) *Inverse-of-prime distribution.* In [Gen09, Gen10], Gentry considers ideal lattices sampled as *the inverse of a uniformly random prime ideal with norm in a prescribed interval* $[A, B]$. We refer to the corresponding average-case problem as $\mathfrak{P}^{-1}$-Id-HSVP$_\gamma$.
(2) *Discrete Haar distribution.* In the article ⟦BDPW20⟧, we consider a discretization of the uniform distribution for the Haar measure. We refer to the corresponding average-case problem as Haar-Id-HSVP$_\gamma$.
(3) *Prime distribution.* In the article ⟦FPSW23⟧, we consider ideal lattices sampled as *a uniformly random prime ideal with norm in a prescribed interval* $[A, B]$ (similar to [Gen09, Gen10], but without the inversion). We refer to the corresponding average-case problem as $\mathfrak{P}$-Id-HSVP$_\gamma$.

These articles prove worst-case to average-case reductions for Id-HSVP$_\gamma$ for each of these distributions (with a bounded loss in the approximation factor).

The "inverse of prime" distribution of [Gen09] may be surprising. This choice is explained by the fact that [Gen09] focuses on the *bounded distance decoding* problem (BDD), a problem *dual* to SVP. It is a convenient choice for studying BDD, but ill-suited for SVP. Indeed, recall that we value distributions supported on integral ideals of "small" norm. Gentry's reduction allows for interval boundaries A and B as small as $\Delta_K^{O(1)} \cdot n^{O(n)}$. Sampling $\mathfrak{p}$ of norm at most B, inverting it, then rescaling it to an integral ideal yields an instance of the form $N(\mathfrak{p})/\mathfrak{p}$, with norm of the order of $\Delta_K^{O(n)} \cdot n^{O(n^2)}$. This is too large for certain applications. Also note that the worst-case to average-case reduction of Gentry (and the algorithm to sample from this distribution) requires access to a factoring oracle (and otherwise runs in classical polynomial time).

The "discrete Haar distribution" is perhaps the most natural. It is mathematically very convenient, thanks to the properties of the Haar distribution, such as the rapid equidistribution of random walks, Theorem 3.10. The worst-case to average-case reduction (and the algorithm to sample from this distribution) from ⟦BDPW20⟧ runs in classical polynomial time, with no need for a factoring oracle. However, to obtain a "good enough" approximation of the (continuous) Haar distribution, we define a discretization supported on ideals of rather large norm, up to $\Delta_K^{O(1)} \cdot 2^{O(n^2)}$. We discuss this case in greater detail in Section 3.4.1.

Finally, the "prime distribution" is supported on integral ideals of much smaller norm: the bounds A and B can be as small as $\Delta_K^{O(1)} \cdot n^{O(n)}$. It is the first distribution supported on integral ideals of norm sufficiently small for certain applications. The worst-case to average-case reduction proved in ⟦FPSW23⟧ unlocks the first distribution on instances for the NTRU cryptosystem with a polynomial modulus whose hardness is supported by a worst-case lattice problem. The analysis of this distribution combines the results of [Gen09] with the techniques of ⟦BDPW20⟧ (random walks on Arakelov class groups). Like the reduction

of [Gen09], it requires a factoring oracle (and otherwise runs in classical polynomial time). We discuss this case in greater detail in Section 3.4.2.

3.4.1. Hardness for the discrete Haar distribution. The random walks studied in Section 3.3 hint at a natural strategy for a worst-case to average-case reduction, in the same vein as Section 1.2.3. Suppose we have an oracle for Haar-Id-HSVP$_\gamma$, i.e., solving with good probability for "uniformly random" ideals. Let Λ be an arbitrary ideal lattice (a worst-case instance). We apply the following steps:

(1) *Randomization:* from the starting point Λ, one can generate a random walk following Section 3.3.1. It first consists in a discrete walk, replacing Λ with the sublattice $\Lambda' = (\prod_i \mathfrak{p}_i)\Lambda$, for a random sequence of prime ideals $(\mathfrak{p}_i)_i$. Then, the continuous walk replaces Λ' with $\Lambda'' = \delta\Lambda'$ where $\delta \in K_{\mathbf{R}}^\times$ is a random "distorsion" close to the identity.
(2) *Solving the average case:* call the Haar-Id-HSVP$_\gamma$ oracle for the random ideal Λ''. Since Λ'' is close to uniform, the average-case oracle succeeds with good probability. It returns an element $x'' \in \Lambda''$ such that $0 < \|x''\| \leq \gamma \cdot \mathrm{Vol}(\Lambda'')^{1/n}$.
(3) *Pulling back the solution:* Return $x = \delta^{-1}x'' \in \delta^{-1}\Lambda'' = \Lambda' \subseteq \Lambda$.

As required, we get $x \in \Lambda$. Let us estimate the approximation factor achieved by this strategy. Since δ is close to the identity, we have $\|x\| \approx \|x''\|$. Let $\mathfrak{w} = \prod_i \mathfrak{p}_i$ be the ideal corresponding to the discrete walk. We obtain

$$\|x\| \approx \|x''\| \leq \gamma \cdot \mathrm{Vol}(\Lambda'')^{\frac{1}{n}} \approx \gamma \cdot \mathrm{Vol}(\Lambda')^{\frac{1}{n}} = (\gamma N(\mathfrak{w})^{\frac{1}{n}}) \cdot \mathrm{Vol}(\Lambda)^{\frac{1}{n}}.$$

Using the bounds B and κ from Theorem 3.10, we have $N(\mathfrak{w})^{1/n} \leq B^{\kappa/n}$. We obtain a reduction from worst-case Id-HSVP$_{\gamma'}$ to average-case Haar-Id-HSVP$_\gamma$ with a loss of $\gamma'/\gamma \approx B^{\kappa/n}$ in the approximation factor. This loss is polynomial in the degree n and the root-discriminant $\Delta_K^{1/n}$. More precisely, we prove in ⟦BDPW20, Theorem 4.5⟧ that we can achieve a loss of

$$\gamma'/\gamma = O(B^{\kappa/n}) \leq \begin{cases} \widetilde{O}\left(n^{1/2}\right) & \text{for prime-power cyclotomic fields, assuming } h_K^+ \leq (\log n)^n \\ \widetilde{O}\left(n^{1-n\mathrm{c}/n} \cdot \Delta_K^{1/(2n)}\right) & \text{for arbitrary number fields.} \end{cases}$$

with h_K^+ the class number of the maximal totally real subfield.

Discretization. The above discussion totally abstracts away the (significant) trouble of discretization. In ⟦BDPW20⟧ we introduce a rounding procedure, which takes as input an arbitrary ideal lattice, and returns a good approximation by a fractional ideal. The *actual* average-case distribution for Haar-Id-HSVP$_\gamma$ is the result of sampling uniformly at random for the Haar measure, then applying the rounding procedure. The rounding procedure is itself randomized, and has the following properties:

- The output is a good approximation of the input — its geometry is almost the same.
- Given an input ideal lattice, the distribution on the output depends only on its isomorphism class — not on a representative of the class.
- As a fractional ideal, the numerator and denominator of the output have bounded norms.

When the input is a fractional lattice, there is an efficient algorithm to apply this rounding procedure. In particular, one can efficiently sample from the discretized Haar distribution.

3.4.2. Hardness for random small-norm prime ideals. In the article ⟦FPSW23⟧, we consider the average hardness of $\mathfrak{P}$-Id-HSVP$_\gamma$, for ideal lattices sampled as a uniformly random prime ideal with norm in a prescribed interval $[A, B]$.

Despite the similarity with $\mathfrak{P}^{-1}$-Id-HSVP$_\gamma$, mimicking the technique of [Gen09] does not work. The reduction in [Gen09] utilises the fact that $\mathcal{O} \subset \mathfrak{p}^{-1}$, readily providing some reasonably small vectors like $1 \in \mathfrak{p}^{-1}$. There is no exploitable analog for $\mathfrak{p}$.

However, we can still exploit the main result of [Gen09]: the worst-case to average-case reduction from Id-HSVP$_{\gamma''}$ to $\mathfrak{P}^{-1}$-Id-HSVP$_{\gamma'}$ (for some approximation factors γ' and γ''). We prove in ⟦FPSW23⟧ that $\mathfrak{P}^{-1}$-Id-HSVP$_{\gamma'}$ reduces to $\mathfrak{P}$-Id-HSVP$_\gamma$, and deduce the average-case hardness of $\mathfrak{P}$-Id-HSVP$_\gamma$ by composition of reductions:

$$\text{Id-HSVP}_{\gamma''} \underset{\text{[Gen09]}}{\leq} \mathfrak{P}^{-1}\text{-Id-HSVP}_{\gamma'} \underset{\text{⟦FPSW23⟧}}{\leq} \mathfrak{P}\text{-Id-HSVP}_{\gamma}.$$

The last reduction works as follows. Suppose we have an oracle for $\mathfrak{P}$-Id-HSVP$_\gamma$, and we are given an instance $\mathfrak{p}^{-1}$ of $\mathfrak{P}^{-1}$-Id-HSVP$_{\gamma'}$. We proceed as follows:

(1) Call the oracle on $\mathfrak{p}$, which finds (with good probability) a "small" element $x_\mathfrak{p} \in \mathfrak{p}$ such that $0 < \|x_\mathfrak{p}\| \leq \gamma \cdot \mathrm{Vol}(\mathfrak{p})^{1/n}$.
(2) Sample a pair $(\mathfrak{b}, y)$ such that $\mathfrak{b}$ is a uniformly random integral ideal of bounded norm, and $y \in (\mathfrak{bp})^{-1}$ is small. This is some kind of *trapdoor generation*: first generating an ideal $\mathfrak{b}$, then finding a small element y would be infeasible, but generating both "simultaneously" can be done. It is not obvious how to perform such a sampling task. For the moment, consider that this is feasible thanks to the knowledge of a small element $x_\mathfrak{p} \in \mathfrak{p}$, and using a factoring oracle.
(3) If $\mathfrak{b}$ is not prime, try again from the start.
(4) Now that we have ensured that $\mathfrak{b}$ is a random *prime* ideal, we can call the oracle (for the second time) on $\mathfrak{b}$, which finds a "small" element $x_\mathfrak{b} \in \mathfrak{b}$ such that $0 < \|x_\mathfrak{b}\| \leq \gamma \cdot \mathrm{Vol}(\mathfrak{b})^{1/n}$.
(5) Return $x = x_\mathfrak{b} \cdot y \in \mathfrak{b} \cdot (\mathfrak{bp})^{-1} = \mathfrak{p}^{-1}$.

We indeed obtain $x \in \mathfrak{p}^{-1}$ as required, and x is a product of "small" elements, so it is itself "small". It is a solution of $\mathfrak{P}^{-1}$-Id-HSVP$_{\gamma'}$ for some approximation factor γ'. We prove in ⟦FPSW23, Corollary 5.3⟧ that this method achieves a loss in the approximation factor of $\gamma'/\gamma = O(n\Delta_K^{1/n})$.

The trapdoor generation of Step 2. Step 2 is the most subtle. It requires a combination of ingredients from [Gen09, Gen10] and from the upcoming Section 3.5.1 on ideal sampling. Here, we only explain the ingredient from [Gen09, Gen10], which solves a simpler problem of the same type: sampling a random ideal $\mathfrak{a}$ together with a small element $\alpha \in \mathfrak{a}$.

From a random ideal, it may be hard to extract a small element, but one could proceed the other way around: first sample a small element $\alpha \in \mathcal{O}_K$, then let $\mathfrak{a} = \alpha\mathcal{O}_K$. By construction, α is small, and $\alpha \in \mathfrak{a}$. The resulting ideal $\mathfrak{a}$ may be random, but it is far from any natural notion of uniformity: it is always a principal ideal.

To correct this obvious bias, one could replace the ideal $\alpha\mathcal{O}_K$ with a random ideal containing α — in other words, a random ideal dividing $\alpha\mathcal{O}_K$. This is where the need for a factoring oracle arises: compute the factorization $\alpha\mathcal{O}_K = \prod_i \mathfrak{q}_i^{e_i}$ into distinct prime ideals $\mathfrak{q}_i$. One can then sample one of the ideals $\mathfrak{q}_i$ uniformly at random, and return $(\alpha, \mathfrak{q}_i)$. Three obvious biases remain:

(1) First, the distribution is now supported on prime ideals; this is not an issue.
(2) Second, *small* prime ideals are significantly overrepresented. This bias can be corrected by ignoring ideals below some bound A, and some rejection-sampling.
(3) Third, *large principal* prime ideals are overrepresented. Indeed, if $\alpha\mathcal{O}_K$ happens to be prime, it is returned unchanged. More generally, there is a bias towards large primes whose inverse class contains a small ideal. This bias can be corrected by ignoring ideals above some bound B.

One can in fact show that for some appropriate bounds A and B, the above procedure returns a pair $(\alpha, \mathfrak{q})$ where $\alpha \in \mathfrak{q}$ is small and $\mathfrak{q}$ is close to uniformly distributed among prime ideals with norm in the interval $[A, B]$.

3.5. A rigorous tool for algorithmic number theory

In this final section, we shift our attention away from cryptography, and towards some of the most fundamental problems in computational number theory: the computation of class groups, unit groups, or a variety of other problems related to the manipulation of ideals in number fields.

Many number theoretic algorithms resort to heuristic assumptions for their analysis. This issue concerns even these field-defining problems. This persistent need for heuristic assumptions often stems from a step of this form: given an ideal class $[\mathfrak{a}]$ of a number field K, find a representative $\mathfrak{b} \in [\mathfrak{a}]$ belonging to a particular family $\mathcal{S}$ of ideals (for instance, the family of smooth ideals). It is relatively simple to design an algorithm for this task: sample a random representative $\mathfrak{b} \in [\mathfrak{a}]$, and hope that it belongs to the desired family $\mathcal{S}$. One then heuristically argues that the probability that $\mathfrak{b} \in \mathcal{S}$ should be proportional to the density of $\mathcal{S}$. For instance, the subexponential density of smooth ideals heuristically implies that one can find smooth representatives in subexponential time. This is the heart of state-of-the-art algorithms to compute class groups, unit groups, or generators of principal ideals in number fields [Buc88, LL93, BF14], and has long constituted a theoretical obstacle overcome only by heuristic arguments (with the exception of quadratic fields [HM89]).

In the paper [[BPW24, Part 1]], we propose a general strategy to solve these ideal sampling tasks rigorously and efficiently, assuming only the Generalized Riemann Hypothesis. Illustrating the power of this technique, we present in [[BPW24, Part 2]] the first algorithm for computing class groups and unit groups of arbitrary number fields that provably runs in probabilistic subexponential time, assuming GRH.

3.5.1. Sampling ideals in a class. Let $\mathcal{S}$ be an arbitrary family of ideals, and $\mathcal{S}_B$ the family of B-smooth ideals (i.e., products of prime ideals of norm at most B). In [[BPW24, Part 1]], we describe an efficient algorithm that samples $\mathfrak{b} \in [\mathfrak{a}]$ such that $\mathfrak{b} \in \mathcal{S} \cdot \mathcal{S}_B$ with probability proportional to the density of $\mathcal{S}$. The set $\mathcal{S}_B$ is used to randomize the input (it corresponds to the discrete walk of Section 3.3), and B can be chosen as small as $(\log \Delta_K)^{O(1)}$. In greater generality, we prove this result for arbitrary *ray* class groups, and when the set $\mathcal{S}_B$ is restricted to ideals whose prime factors fall in a prescribed subgroup. For concreteness, Theorem 3.14 below is a specialization of the main theorem of [[BPW24, Part 1]] to the simplest case, without ray nor subgroups. Here, the quantity $\delta_{\mathcal{S}}[r^n]$ is the *local density* of $\mathcal{S}$, i.e., the proportion of ideals of norm at most r^n that belong to $\mathcal{S}$. It is essentially $\delta_{\mathcal{S}}[r^n] \approx \frac{\#\{\mathfrak{b} \in \mathcal{S} | N(\mathfrak{b}) < r^n\}}{\#\{\mathfrak{b} | N(\mathfrak{b}) < r^n\}}$, and tends to the *natural density* as $r \to \infty$.

Theorem 3.14. *Assuming the Generalized Riemann Hypothesis, there is a randomized algorithm $\mathcal{A}$ such that the following holds. Let K be a number field, with degree n, discriminant Δ_K, and ring of integers $\mathcal{O}_K$. Let $\mathfrak{a} \subseteq \mathcal{O}_K$ be an integral ideal. Let $\varepsilon \in \mathbf{R}_{>0}$, let $\mathsf{b} \geq 2$ be an integer, and let $r \geq 16 \cdot \mathsf{b}^{2n/\mathsf{b}} \cdot n^{7/2} \cdot |\Delta_K|^{3/(2n)}$.*

Given the above data, the algorithm $\mathcal{A}$ outputs $\beta \in \mathfrak{a}$ such that $\beta\mathfrak{a}^{-1} \in \mathcal{S} \cdot \mathcal{S}_B$ with probability at least $\delta_{\mathcal{S}}[r^n]/3 - \varepsilon$, for some smoothness bound $B = (\log |\Delta_K| + \log\log(1/\varepsilon))^{O(1)}$ and for any set $\mathcal{S}$ of integral ideals. Furthermore, the algorithm runs in expected polynomial time in $\log |\Delta_K|$, $\log(N(\mathfrak{a}))$, $\log(1/\varepsilon)$, b^{b}, and in the length of the input.

Remark 3.15. Note that the algorithm is described in a slightly different way than the above discussion: given $\mathfrak{a}$, we find $\beta \in \mathfrak{a}$ such that $\beta\mathfrak{a}^{-1} \in \mathcal{S} \cdot \mathcal{S}_B$. The ideal $\beta\mathfrak{a}^{-1}$ is in the

inverse class of $\mathfrak{a}$, so up to an inversion, this problem is equivalent to the ideal sampling problem discussed above.

Overview of the technique. The folklore strategy to solve ideal sampling tasks is the following. The input ideal $\mathfrak{a}$ is seen as a lattice, via the Minkowski embedding. One may find a reasonably short basis of $\mathfrak{a}$ (for instance, by means of LLL [LLL82]), which then allows one to sample a reasonably short random element $\beta \in \mathfrak{a}$, and hope that the ideal $\mathfrak{b} = \beta\mathfrak{a}^{-1}$ belongs to the desired family $\mathcal{S}$. One then typically argues (heuristically!) that the probability of success is proportional to the density of $\mathcal{S}$.

To obtain a rigorous sampling algorithm, we proceed in two steps. First, we prove that a fairly straightforward strategy as above indeed has the desired probability of success *when the input $\mathfrak{a}$ is treated as a random ideal lattice with uniformly random Arakelov class.* More precisely, we prove that there is a reasonably small "ball" $r\mathcal{B}$ (in the embedding space $K_{\mathbf{R}}$) such that the *expected* density of elements $\beta \in \mathfrak{a} \cap r\mathcal{B}$ such that $\beta\mathfrak{a}^{-1} \in \mathcal{S}$ is proportional to the density of $\mathcal{S}$.

Second, we deal with arbitrary input $\mathfrak{a}$ by randomizing its Arakelov class via a random walk, following Section 3.3. Concretely, the input $\mathfrak{a}$ is multiplied by random ideals of small prime norm (the discrete part of the random walk), and is randomly distorted according to some Gaussian distribution (the continuous part of the random walk). Theorem 3.10 ensures that the result is uniformly distributed in the Arakelov class group. The discrete part of the random walk introduces small prime factors, hence our method samples ideals in $\mathcal{S}\cdot\mathcal{S}_B$ instead of $\mathcal{S}$. In all applications we are aware of, we have $\mathcal{S} = \mathcal{S}\cdot\mathcal{S}_B$.

The randomization step is a straightforward application of Section 3.3, so in the rest of this section, we explain the main novelty of ⟦BPW24, Part 1⟧: proving that the folklore sampling method works for random ideals whose Arakelov classes are uniformly distributed.

Sampling in a fixed ideal lattice. Let $\Lambda = x\mathfrak{a} \in \mathrm{IdLat}_K^0$ be a normalized ideal lattice. We consider the following sampling procedure. It takes as input the ideal lattice Λ, and a parameter $r > 0$ controlling how large we want the sampled element to be.

(1) Consider a "ball" $\mathcal{B}$ in the embedding space $K_{\mathbf{R}}$. This is the shape we are going to sample from. The reader may think of $\mathcal{B}$ as a unit ball in $K_{\mathbf{R}}$ centered at the origin. The conventional ℓ^2-ball is not the most convenient choice. In ⟦BPW24⟧, we consider the unit ball for the ℓ^∞-norm. In ⟦FPSW23⟧ we achieve finer properties by crafting a stranger shape.
(2) Sample $\beta \leftarrow \Lambda \cap r\mathcal{B}$ uniformly at random. This finite set is the intersection of a lattice with a "ball" of radius r. There is an efficient algorithm sampling from this distribution, at least when r is not too small.
(3) Return the integral ideal $\beta\Lambda^{-1}$.

Let us analyse the distribution of $\beta\Lambda^{-1}$. For the moment, we consider Λ to be fixed, and the source of randomness is the uniform sampling $\beta \leftarrow \Lambda \cap r\mathcal{B}$. Consider the probability for the output to be a particular ideal $\mathfrak{b}$:

(3.3)
$$\Pr_{\beta\leftarrow\Lambda\cap r\mathcal{B}}[\beta\Lambda^{-1} = \mathfrak{b}] = \Pr_{\beta\leftarrow\Lambda\cap r\mathcal{B}}[\beta \text{ generates the } \mathcal{O}_K\text{-module } \mathfrak{b}\Lambda] = \frac{\#(\mathrm{gen}(\mathfrak{b}\Lambda)\cap r\mathcal{B})}{\#(\Lambda\cap r\mathcal{B})},$$

where for any ideal lattice Λ, we denote by $\mathrm{gen}(\Lambda) = \{x \in \Lambda \mid \Lambda = x\mathcal{O}_K\}$ the set of $\mathcal{O}_K$-generators (which may well be empty). The denominator $\#(\Lambda \cap r\mathcal{B})$ is easy to estimate: assuming that $r\mathcal{B}$ is large enough (compared to the covolume of Λ — or more accurately, to its *covering radius*), then $\#(\Lambda \cap r\mathcal{B})$ is well-approximated by the volume-to-covolume

ratio

$$\#(\Lambda \cap r\mathcal{B}) \approx \frac{\mathrm{Vol}(r\mathcal{B})}{\mathrm{Vol}(\Lambda)} = r^n \frac{\mathrm{Vol}(\mathcal{B})}{|\Delta_K|^{1/2}}. \tag{3.4}$$

The numerator $\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B})$ presents a more delicate challenge. Clearly, it is zero if $\mathfrak{b}\Lambda$ has no generator. If there exists a generator $g \in \mathrm{gen}(\mathfrak{b}\Lambda)$, then $\mathrm{gen}(\mathfrak{b}\Lambda) = g\mathcal{O}_K^\times$. Recall the logarithm map

$$\mathrm{Log}_\infty : K_{\mathbf{R}}^\times \longrightarrow \bigoplus_\nu \mathbf{R} : (y_\sigma)_\sigma \longmapsto \sum_\nu \log|y_{\sigma_\nu}| \cdot (\!|\nu|\!).$$

Writing $\alpha = \mathrm{Log}_\infty(g)$, we have $\mathrm{Log}_\infty(\mathrm{gen}(\mathfrak{b}\Lambda)) = \alpha + \mathrm{Log}(\mathcal{O}_K^\times)$. Since the kernel of $\mathrm{Log}|_{\mathcal{O}_K^\times}$ consists of the roots of unity μ_K, we obtain that

$$\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B}) = \#\mu_K \cdot \#(\underbrace{(\alpha + \mathrm{Log}(\mathcal{O}_K^\times))}_{\text{A translated lattice}} \cap \underbrace{\mathrm{Log}_\infty(r\mathcal{B})}_{\text{A ``logarithmic ball''}}).$$

Now, we have reduced the problem of estimating $\Pr[\beta\Lambda^{-1} = \mathfrak{b}]$ to the problem of counting the number of points in the intersection of a translated lattice $(\alpha + \mathrm{Log}(\mathcal{O}_K^\times))$ with a kind of "logarithmic ball" $\mathrm{Log}_\infty(r\mathcal{B})$. While not quite a ball, it is helpful to think of $\mathrm{Log}_\infty(r\mathcal{B})$ as a ball of radius $\log r$.

One may be tempted to apply the same method as for the numerator: the intersection should be approximately the volume-to-covolume ratio $\mathrm{Vol}(\mathrm{Log}_\infty(r\mathcal{B}))/\mathrm{Vol}(\mathrm{Log}(\mathcal{O}_K^\times))$. There are two issues with that. The first is that the lattice $\mathrm{Log}(\mathcal{O}_K^\times)$ does not have full rank in the ambient space: it spans the hyperplane H. Therefore, instead of the "ball" $\mathrm{Log}_\infty(r\mathcal{B})$, one should consider the volume of a "slice"

$$\mathscr{S}(r, N(\mathfrak{b})) = (\alpha + H) \cap \mathrm{Log}_\infty(r\mathcal{B}).$$

As the notation suggests, it does not depend on a choice of α: the shifted hyperplane $\alpha + H = \mathrm{Log}_\infty(N(\mathfrak{b})^{1/n}) + H$ depends only on $N(\mathfrak{b})$. We should then expect

$$\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B}) \overset{?}{\approx} \#\mu_K \cdot \frac{\mathrm{Vol}(\mathscr{S}(r, N(\mathfrak{b}))}{\mathrm{Vol}(\mathrm{Log}(\mathcal{O}_K^\times))}. \tag{3.5}$$

The second issue is more fundamental: the volume-to-covolume approximation for lattice points requires the volume of the ball to be significantly larger than the covolume of the lattice. The volume of $\mathrm{Log}(\mathcal{O}_K^\times)$ is, up to proper normalization, the regulator of the field, and can grow exponentially in n and $\log|\Delta_K|$. Applying the volume-to-covolume approximation would require r to be unreasonably large.

Average sampling for uniform Arakelov classes. This is where randomization comes in. While the estimation (3.5) is very inaccurate for any particular Λ, it does hold *on average*. Suppose $\Lambda \in \mathrm{IdLat}_K^0$ is a random (normalized) ideal lattice, following a distribution $\mathcal{D}$ such that the Arakelov class of Λ is uniformly distributed.

We make two observations. First, the quantity $\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B})$ depends only on the *Arakelov class* (i.e., isometry class) of the ideal lattice Λ. Indeed, if Λ' is isometric to Λ, then so are $\mathfrak{b}\Lambda$ and $\mathfrak{b}\Lambda'$, and the isometry gives a bijection between $\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B}$ and $\mathrm{gen}(\mathfrak{b}\Lambda') \cap r\mathcal{B}$. This is not exactly true for any choice of $r\mathcal{B}$: we need that $r\mathcal{B}$ is somewhat preserved by isometries (like an actual ℓ^2-ball).

Second, the quantity $\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B})$ is non-zero only if the ideal class of $\mathfrak{b}\Lambda$ is trivial, which means that its Arakelov class is in the image of $T_K \to \mathrm{Pic}_K^0$. Recall that $T_K = H/\mathrm{Log}(\mathcal{O}_K^\times)$ is the Log-unit torus. Let $F \subset H$ be a fundamental domain for $\mathrm{Log}(\mathcal{O}_K^\times)$ (i.e., the projection $F \to T_K$ is a bijection). The isometry class of $\mathfrak{b}\Lambda$ is uniform

among lattices of volume $\mathrm{Vol}(\mathfrak{b}) = N(\mathfrak{b})|\Delta_K|^{1/2}$. In particular, when non-empty, the translated lattice $\mathrm{Log}_\infty(\mathrm{gen}(\mathfrak{b}\Lambda))$ is uniformly random among the translated lattices

$$\beta + \mathrm{Log}_\infty(N(\mathfrak{b})^{1/n}) + \mathrm{Log}(\mathcal{O}_K^\times) \subset \mathrm{Log}_\infty(N(\mathfrak{b})^{1/n}) + H,$$

with $\beta \in F$. It is well-known that for any (full-rank) lattice $L \subset V$ and measurable set $B \subset V$, the average of $\#((\alpha + L) \cap B)$ over all possible translates $\alpha + L$ is equal to $\mathrm{Vol}(B)/\mathrm{Vol}(L)$. We deduce that the estimation (3.5) holds on average in the following sense:

$$\begin{aligned}
&\mathbf{E}_{\Lambda \leftarrow \mathcal{D}}\left[\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B})\right] \\
&= \frac{\#\mu_K}{\#\,\mathrm{Cl}(K)} \cdot \mathbf{E}_{\beta \leftarrow F}\left[\#((\beta + \mathrm{Log}_\infty(N(\mathfrak{b})^{1/n}) + \mathrm{Log}(\mathcal{O}_K^\times)) \cap \mathrm{Log}_\infty(r\mathcal{B}))\right] \\
&= \frac{\#\mu_K}{\#\,\mathrm{Cl}(K)} \cdot \frac{\mathrm{Vol}((\mathrm{Log}_\infty(N(\mathfrak{b})^{1/n}) + H) \cap \mathrm{Log}_\infty(r\mathcal{B}))}{\mathrm{Vol}(\mathrm{Log}(\mathcal{O}_K^\times))} \\
&= \frac{\#\mu_K}{\#\,\mathrm{Cl}(K)\,\mathrm{Vol}(\mathrm{Log}(\mathcal{O}_K^\times))} \cdot \mathrm{Vol}(\mathscr{S}(r, N(\mathfrak{b}))). \qquad (3.6)
\end{aligned}$$

We are ready to go back to estimating $\Pr[\beta\Lambda^{-1} = \mathfrak{b}]$, this time with a random ideal lattice Λ. We have

$$\begin{aligned}
\Pr_{\substack{\Lambda \leftarrow \mathcal{D} \\ \beta \leftarrow \Lambda \cap r\mathcal{B}}}[\beta\Lambda^{-1} = \mathfrak{b}] &\overset{(3.3)}{=} \mathbf{E}_{\Lambda \leftarrow \mathcal{D}}\left[\frac{\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B})}{\#(\Lambda \cap r\mathcal{B})}\right] \\
&\overset{(3.4)}{\approx} \frac{|\Delta_K|^{1/2} \cdot \mathbf{E}_{\Lambda \leftarrow \mathcal{D}}\left[\#(\mathrm{gen}(\mathfrak{b}\Lambda) \cap r\mathcal{B})\right]}{r^n\,\mathrm{Vol}(\mathcal{B})} \\
&\overset{(3.6)}{=} \frac{|\Delta_K|^{1/2} \cdot \#\mu_K}{\#\,\mathrm{Cl}(K)\,\mathrm{Vol}(\mathrm{Log}(\mathcal{O}_K^\times))} \cdot \frac{\mathrm{Vol}(\mathscr{S}(r, N(\mathfrak{b})))}{r^n\,\mathrm{Vol}(\mathcal{B})}.
\end{aligned}$$

In summary, there is a constant C_K, depending only on the field K, such that

$$\Pr_{\substack{\Lambda \leftarrow \mathcal{D} \\ \beta \leftarrow \Lambda \cap r\mathcal{B}}}[\beta\Lambda^{-1} = \mathfrak{b}] \approx C_K \cdot \frac{\mathrm{Vol}(\mathscr{S}(r, N(\mathfrak{b})))}{r^n\,\mathrm{Vol}(\mathcal{B})}.$$

In particular, the probability for the sampler to hit $\mathfrak{b}$ depends only on $N(\mathfrak{b})$. With an appropriate choice of $\mathcal{B}$, one can ensure that the volume of the slice $\mathscr{S}(r, N(\mathfrak{b}))$ is essentially constant for $N(\mathfrak{b}) < r^n$, and suddenly drops to zero when $N(\mathfrak{b}) > r^n$. On other words, as a function of $\mathfrak{b}$, the probability $\Pr[\beta\Lambda^{-1} = \mathfrak{b}]$ is proportional to the indicator function $\mathbf{1}_{N(\mathfrak{b}) < r^n}$.

Needless to say, all the approximations we have made in this discussion require much more careful consideration. Still, we hope that we have given enough elements to convince the reader of the following informal lemma.

Lemma 3.16 (informal)**.** *When the input Λ has uniformly random Arakelov class, the sampler outputs an ideal $\beta\Lambda^{-1}$ which is* almost *uniformly distributed among ideals of norm at most r^n.*

Sketch of the proof of Theorem 3.14. The above informal lemma unlocks the proof of Theorem 3.14. On input an ideal $\mathfrak{a}$, first generate a random walk to $\Lambda = \delta\mathfrak{w}\mathfrak{a}$ (where $\delta \in K_{\mathbf{R}}^\times$ is the continuous walk and the B-smooth ideal $\mathfrak{w} \in \mathcal{S}_B$ is the discrete walk). Applying Theorem 3.10, the ideal lattice Λ is uniformly random in the Arakelov class group. Sampling $\beta_0 \leftarrow \Lambda \cap r\mathcal{B}$, Lemma 3.16 ensures that $\beta_0\Lambda^{-1}$ is essentially uniform among integral ideals of norm at most r^n. We get

$$\Pr_{\substack{\Lambda \leftarrow \mathcal{D} \\ \beta_0 \leftarrow \Lambda \cap r\mathcal{B}}}[\beta_0\Lambda^{-1} \in \mathcal{S}] = \sum_{\mathfrak{b} \in \mathcal{S}} \Pr_{\substack{\Lambda \leftarrow \mathcal{D} \\ \beta \leftarrow \Lambda \cap r\mathcal{B}}}[\beta_0\Lambda^{-1} = \mathfrak{b}] \approx \frac{\#\{\mathfrak{b} \in \mathcal{S} \mid N(\mathfrak{b}) < r^n\}}{\#\{\mathfrak{b} \mid N(\mathfrak{b}) < r^n\}} \approx \delta_{\mathcal{S}}[r^n],$$

where $\delta_{\mathcal{S}}[r^n]$ is the "local density" of $\mathcal{S}$, i.e., essentially the proportion of integral ideals of norm at most r^n that belong to the family $\mathcal{S}$.

Finally, output $\beta = \delta^{-1}\beta_0 \in \mathfrak{w}\mathfrak{a} \subset \mathfrak{a}$. We have

$$\beta\mathfrak{a}^{-1} = \beta_0\Lambda^{-1}\mathfrak{w} \in (\beta_0\Lambda^{-1}) \cdot \mathcal{S}_B,$$

which falls in $\mathcal{S}\cdot\mathcal{S}_B$ with probability at least (approximately) $\delta_{\mathcal{S}}[r^n]$, proving Theorem 3.14. □

3.5.2. Rigorous computation of class groups and unit groups. In ⟦BPW24, Part 2⟧, we illustrate the power of the ideal-sampling technique of ⟦BPW24, Part 1⟧ by applying it to the computation of class groups and unit groups.

Let K be a number field of degree n and discriminant Δ_K. The determination of the structure of its class group $\mathrm{Cl}(K)$, together with a system of fundamental units, is one of the main problems of computational number theory [Coh93, p. 217]. It has long been believed that this task can be solved in probabilistic subexponential time. Such algorithms have been described and analyzed under a variety of heuristic assumptions [Buc88, BF14]. Despite decades of investigation, only imaginary quadratic fields have been amenable to a rigorous analysis [HM89], assuming GRH. In ⟦BPW24, Part 2⟧, we present the first general algorithm for this problem that provably runs in probabilistic subexponential time, assuming GRH. Recall the classical L-notation

$$L_x(\alpha) = \exp\left(O(\log x)^{\alpha}(\log\log x)^{1-\alpha}\right).$$

We prove the following theorem.

Theorem 3.17. *Assuming the Generalized Riemann Hypothesis, there is a probabilistic algorithm which, on input a number field K of degree n and discriminant Δ_K, computes its ideal class group and a compact representation of a fundamental system of units, and runs in expected time polynomial in the length of the input, in $L_{|\Delta_K|}(1/2)$, in $L_{n^n}(2/3)$, and in $\min(\rho_K, L_{|\Delta_K|}(2/3+o(1)))$, where ρ_K is the residue at 1 of the Dedekind zeta function ζ_K.*

It has been conjectured since Buchmann's 1988 heuristic algorithm [Buc88] that this problem can be solved in subexponential time $L_{|\Delta_K|}(1/2)$ for any family of fields of fixed degree. Theorem 3.17 implies this conjecture, assuming GRH.

Then, it was conjectured by Biasse and Fieker's 2014 algorithm [BF14] that this problem can be solved in subexponential time even for varying degree. Again, Theorem 3.17 implies this conjecture, assuming GRH. However, Biasse and Fieker conjectured a complexity as in Theorem 3.17 where the quantity ρ_K is replaced with $L_{n^n}(2/3)$. In our analysis, the quantity ρ_K arises from the best known estimates on the density of bounded smooth ideals. It seems ρ_K should appear in the same way in the heuristic complexity of [BF14], unless one expects a better bound on the density of smooth ideals.

Blueprint of the algorithm. The algorithm in Theorem 3.17 follows the classical strategy for the computation of the class group. Consider a bound $B > 0$, and let $\mathscr{P}_K(B)$ be the set of prime ideals of norm at most B. For B large enough, this set $\mathscr{P}_K(B)$ generates the class group [Bac90]. "Computing the class group" consists in finding a basis of the kernel of the map

$$\Phi : \bigoplus_{\mathfrak{p}\in\mathscr{P}_K(B)} \mathbf{Z} \longrightarrow \mathrm{Cl}(K) : (n_{\mathfrak{p}})_{\mathfrak{p}} \longmapsto \left[\prod_{\mathfrak{p}} \mathfrak{p}^{n_{\mathfrak{p}}}\right].$$

This kernel is a lattice. An element $(n_{\mathfrak{p}})_{\mathfrak{p}} \in \ker\Phi$ is called a *relation*. All we need to do is generate many random relations, until they generate $\ker\Phi$. We thus need a *relation generator*. The standard approach consists in generating a somewhat random B-smooth

ideal $\mathfrak{s} = \prod_{\mathfrak{p}} \mathfrak{p}^{a_{\mathfrak{p}}} \in \mathcal{S}_B$, then sample random elements $\beta \in \mathfrak{s}$ until $\beta\mathfrak{s}^{-1} = \prod_{\mathfrak{p}} \mathfrak{p}^{b_{\mathfrak{p}}}$ is *also* B-smooth. We obtain a "random" relation $(a_{\mathfrak{p}} + b_{\mathfrak{p}})_{\mathfrak{p}} \in \ker \Phi$, since $[\prod_{\mathfrak{p}} \mathfrak{p}^{a_{\mathfrak{p}}+b_{\mathfrak{p}}}] = [\beta\mathcal{O}_K] = 1$. This approach typically necessitates two heuristic assumptions:

(1) *Efficiency of the relation generator:* the probability for $\beta\mathfrak{s}^{-1} = \prod_{\mathfrak{p}} \mathfrak{p}^{b_{\mathfrak{p}}}$ to be B-smooth is expected to be proportional to the density of B-smooth ideals.
(2) *Random relations are well-distributed:* the resulting relation $(a_{\mathfrak{p}} + b_{\mathfrak{p}})_{\mathfrak{p}} \in \ker \Phi$ should be "random enough": a few samples should rapidly generate $\ker \Phi$.

These are two serious challenges for a heuristic-free algorithm. While it is clear that the new ideal-sampling technique, Theorem 3.14, solves the first challenge, the second requires more work. The main idea is that, for some appropriate distribution on the input ideal $\mathfrak{s} = \prod_{\mathfrak{p}} \mathfrak{p}^{a_{\mathfrak{p}}}$, the sampler returns a relation $(a_{\mathfrak{p}} + b_{\mathfrak{p}})_{\mathfrak{p}} \in \ker \Phi$ which follows a *discrete Gaussian* distribution on $\ker \Phi$. Samples from a discrete Gaussian distribution rapidly generate the lattice on which it is supported.

Computing S-units. While the main result Theorem 3.17 is stated as an algorithm for computing units and class groups, our algorithm actually does slightly more than that: it computes the so-called Log-S-unit lattice for any set S of prime ideals. It is well known that such an algorithm for S-units can be used to compute the class group and the unit group. The same method can also be used to solve other algorithmic problems, such as the principal ideal problem (decide whether an ideal is principal, and when it is, find a generator), or the class group discrete logarithm problem.

3.5.3. Further applications. Sampling smooth ideals is a task that regularly arises in computational number theory. In ⟦BPW24, Part 2⟧, we focus on the problem of class group computation, but it is more generally a common component of index-calculus algorithms, like the general number field sieve for integer factorization [LL93] or the computation of discrete logarithms in finite fields. This direction has not been investigated yet.

Applying the ideal-sampling method of ⟦BPW24, Part 1⟧ to the case where $\mathcal{S}$ is the set of prime ideals allows one to sample in the family $\mathcal{S} \cdot \mathcal{S}_B$ of near-prime ideals, of particular interest in that it constitutes a dense family of efficiently factorable ideals. Therefore, our sampling method provides a rigorous way to transform any ideal $\mathfrak{a}$ into an equivalent ideal $\mathfrak{b}$ of known factorization. Obtaining such factorable ideals (or elements) is a key step in algorithms to compute power residue symbols. Specifically, it allows one to perform the "principalization step" in [BP17, §5.2] efficiently. De Boer has developed this idea in his PhD dissertation [Boe22], applying the main result of ⟦BPW24, Part 1⟧ to construct the first polynomial time algorithm to compute power residue symbols, assuming GRH.

BIBLIOGRAPHY

[ACD+23] Sarah Arpin, James Clements, Pierrick Dartois, Jonathan Komada Eriksen, Péter Kutas, and Benjamin Wesolowski. Finding orientations of supersingular elliptic curves and quaternion orders. Cryptology ePrint Archive, Paper 2023/1268, 2023. https://eprint.iacr.org/2023/1268.

[ACL+23] Sarah Arpin, Mingjie Chen, Kristin E Lauter, Renate Scheidler, Katherine E Stange, and Ha TN Tran. Orienteering with one endomorphism. *La Matematica*, 2(3):523–582, 2023.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming – ICALP 1999*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9, 1999.

[Arp23] Sarah Arpin. Adding level structure to supersingular elliptic curve isogeny graphs. Preprint arXiv:2203.03531, 2023. https://arxiv.org/abs/2203.03531.

[Bac90] Eric Bach. Explicit bounds for primality testing and related problems. *Mathematics of Computation*, 55(191):355–380, 1990.

[Bay99] Eva Bayer. Lattices and number fields. *Contemp. Math.*, 241, 1999.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. *to appear in Advances in Cryptology – CRYPTO 2018*, 2018.

[BCC+23] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14005 of *Lecture Notes in Computer Science*, pages 405–437. Springer, 2023.

[BDD+24] Andrea Basso, Luca De Feo, Pierrick Dartois, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D-West: The Fast, the Small, and the Safer. Cryptology ePrint Archive, Paper 2024/760, 2024. https://eprint.iacr.org/2024/760.

[BDFLS20] Daniel J Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020.

[BDPW20] Koen de Boer, Léo Ducas, Alice Pellet-Mary, and Benjamin Wesolowski. Random self-reducibility of Ideal-SVP via Arakelov random walks. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 243–273. Springer, 2020.

[BdQL+19] Paul Bottinelli, Victoria de Quehen, Chris Leonardi, Anton Mosunov, Filip Pawlega, and Milap Sheth. The dark sidh of isogenies. Cryptology ePrint Archive, Paper 2019/1333, 2019. https://eprint.iacr.org/2019/1333.

[BF14] J.-F. Biasse and C. Fieker. Subexponential class group and unit group computation in large degree number fields. *LMS Journal of Computation and Mathematics*, 17:385–403, 1 2014.

[BFH+24] Alex Biryukov, Ben Fisch, Gottfried Herold, Dmitry Khovratovich, Gaëtan Leurent, María Naya-Plasencia, and Benjamin Wesolowski. Cryptanalysis of algebraic verifiable delay functions. *To appear in Advances in Cryptology – CRYPTO 2024*, 2024.

[BGJT14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.

[BJW17] Ernest Hunter Brooks, Dimitar Jetchev, and Benjamin Wesolowski. Isogeny graphs of ordinary abelian varieties. *Research in Number Theory*, 3(1):28, 2017.

[BKSW23] Karim Belabas, Thorsten Kleinjung, Antonio Sanso, and Benjamin Wesolowski. A note on the low order assumption in class group of an imaginary quadratic number fields. *Mathematical Cryptology*, 3:44–51, Jul. 2023.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247. Springer, 2019.

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security*, volume 12492 of *Lecture Notes in Computer Science*, pages 520–550. Springer, 2020.

[BMP23] Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: fast encryption from supersingular torsion attacks. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, volume 14444 of *Lecture Notes in Computer Science*, pages 98–126. Springer, 2023.

[Boe22] Koen de Boer. *Random Walks on Arakelov Class Groups.* PhD thesis, Leiden University, 2022.

[BP17] Koen de Boer and Carlo Pagano. Calculating the power residue symbol and ibeta. In *ISSAC*, volume 68, pages 923–934, 2017.

[BPW24] Koen de Boer, Alice Pellet-Mary, and Benjamin Wesolowski. Rigorous methods for computational number theory. Preprint available on demand, 2024.

[Buc88] Johannes Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. *Séminaire de théorie des nombres, Paris*, 1989:28–41, 1988.

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *Advances in cryptology – EUROCRYPT 2023. Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447. Springer, 2023.

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585. Springer, 2016.

[CDW17] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short Stickelberger class relations and application to Ideal-SVP. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, volume 10210 of *Lecture Notes in Computer Science*, pages 324–348. Springer, 2017.

[CDW21] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Mildly short vectors in cyclotomic ideal lattices in quantum polynomial time. *Journal of the ACM*, 68(2):8:1–8:26, 2021.

[CGS14] Peter Campbell, Michael Groves, and Dan Shepherd. Soliloquy: A cautionary tale. ETSI 2nd Quantum-Safe Crypto Workshop, 2014. Available at `http://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf`.

[CGS20] Perlas Caranay, Matthew Greenberg, and Renate Scheidler. Computing modular polynomials and isogenies of rank two drinfeld modules over finite fields. In *75 Years of Mathematics of Computation: Symposium on Celebrating 75 Years of Mathematics of Computation, November 1-3, 2018, the Institute for Computational and Experimental Research in Mathematics (ICERM)*, volume 754, page 283. American Mathematical Soc., 2020.

[CHVW22] Wouter Castryck, Marc Houben, Frederik Vercauteren, and Benjamin Wesolowski. On the decisional Diffie-Hellman problem for class group actions on oriented elliptic curves. *Research in Number Theory*, 8(4):99, 2022. Proceedings of the Fifteenth Algorithmic Number Theory Symposium – ANTS-XV.

[CII+23] Mingjie Chen, Muhammad Imran, Gábor Ivanyos, Péter Kutas, Antonin Leroux, and Christophe Petit. Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of psidh. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 99–130. Springer, 2023.

[CJS14] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

[CK20] Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, 14(1):414–437, 2020.

[CL23] Giulio Codogni and Guido Lido. Spectral theory of isogeny graphs. Preprint arXiv:2308.13913, 2023. https://arxiv.org/abs/2308.13913.

[CLG09] Denis X. Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, Jan 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 D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, 2018.

[CLP23] Mingjie Chen, Antonin Leroux, and Lorenz Panny. Scallop-hd: group action from 2-dimensional isogenies. Cryptology ePrint Archive, Paper 2023/1488, 2023. https://eprint.iacr.org/2023/1488.

[Coh93] Henri Cohen. *A course in computational algebraic number theory*, volume 8. Springer-Verlag Berlin, 1993.

[Cos20] Craig Costello. B-SIDH: supersingular isogeny diffie-hellman using twisted torsion. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463. Springer, 2020.

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

[CPV20] Wouter Castryck, Lorenz Panny, and Frederik Vercauteren. Rational isogenies from irrational endomorphisms. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 12106 of *Lecture Notes in Computer Science*, pages 523–548. Springer, 2020.

[DDF+21] Luca De Feo, Cyprien Delpech 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*, volume 13093 of *Lecture Notes in Computer Science*, pages 249–278. Springer, 2021.

[DFK+23] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography – PKC 2023*, volume 13940 of *Lecture Notes in Computer Science*, pages 345–375. Springer, 2023.

[DG19] Luca De Feo and Steven D. Galbraith. Seasign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 11478 of *Lecture Notes in Computer Science*, pages 759–789. Springer, 2019.

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

[DKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.

[DKPS19] Cyprien Delpech de Saint Guilhem, Péter Kutas, Christophe Petit, and Javier Silva. Séta: Supersingular encryption from torsion attacks. IACR Cryptology ePrint Archive, Paper 2019/1291, 2019. https://eprint.iacr.org/2019/1291.

[DLLW23] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the Deuring correspondence - towards practical and secure SQISign signatures. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 659–690. Springer, 2023.

[DLRW24] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: new dimensions in cryptography. *To appear in Advances in Cryptology – EUROCRYPT 2024*, 2024.

[DMPR23] Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert. An algorithmic approach to $(2, 2)$-isogenies in the theta model and applications to isogeny-based cryptography. Cryptology ePrint Archive, Paper 2023/1747, 2023. https://eprint.iacr.org/2023/1747.

[DMPS19] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security*, volume 11921 of *Lecture Notes in Computer Science*, pages 248–277. Springer, 2019.

[DPW19] Léo Ducas, Maxime Plançon, and Benjamin Wesolowski. On the shortness of vectors to be found by the Ideal-SVP quantum algorithm. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, volume 11692 of *Lecture Notes in Computer Science*, pages 322–351. Springer, 2019.

[EHL+18] Kirsten Eisenträger, Sean Hallgren, Kristin 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*, pages 329–368, Cham, 2018. Springer International Publishing.

[EL24] Jonathan Komada Eriksen and Antonin Leroux. Computing orientations from the endomorphism ring of supersingular curves and applications. Cryptology ePrint Archive, Paper 2024/146, 2024. https://eprint.iacr.org/2024/146.

[FKMT22] Tako Boris Fouotsa, Péter Kutas, Simon-Philipp Merz, and Yan Bo Ti. On the isogeny problem with torsion point information. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *Public-Key Cryptography – PKC 2022, Part I*, volume 13177 of *Lecture Notes in Computer Science*, pages 142–161. Springer, 2022.

[FM02] M. Fouquet and F. Morain. Isogeny volcanoes and the SEA algorithm. In Claus Fieker and David R. Kohel, editors, *Algorithmic number theory, 5th International Symposium – ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 276–291. Springer, Berlin, 2002.

[FPSW23] Joël Felderhoff, Alice Pellet-Mary, Damien Stehlé, and Benjamin Wesolowski. Ideal-SVP is hard for small-norm uniform prime ideals. In Guy N. Rothblum and Hoeteck Wee, editors, *Theory of Cryptography – 21st International Conference, TCC 2023*, volume 14372 of *Lecture Notes in Computer Science*, pages 63–92. Springer, 2023.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[Gen09] C. Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2009.

[Gen10] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 116–137. Springer, 2010.

[GHS02] Steven D. Galbraith, Florian Hess, and Nigel P. Smart. Extending the GHS Weil descent attack. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 29–44. Springer, 2002.

[GKL+21] Robert Granger, Thorsten Kleinjung, Arjen K. Lenstra, Benjamin Wesolowski, and Jens Zumbrägel. Computation of a 30750-bit binary field discrete logarithm. *Mathematics of Computation*, 90(332):2997–3022, 2021.

[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, 2020.

[GW17] Alexandre Gélin and Benjamin Wesolowski. Loop-abort faults on supersingular isogeny cryptosystems. In Tanja Lange and Tsuyoshi Takagi, editors, *International Workshop on Post-Quantum Cryptography – PQCrypto 2017*, pages 93–106. Springer, 2017.

[HM89] James L Hafner and Kevin S McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American mathematical society*, 2(4):837–850, 1989.

[HW23] Arthur Herlédan Le Merdy and Benjamin Wesolowski. The supersingular endomorphism ring problem given one endomorphism. Cryptology ePrint Archive, Paper 2023/1448, 2023. https://eprint.iacr.org/2023/1448.

[IK04] H. Iwaniec and E. Kowalski. *Analytic Number Theory*. Number v. 53 in American Mathematical Society Colloquium Publications. American Mathematical Society, 2004.

[JAC+17] David Jao, Reza Azarderakhsh, Matt Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalili, Brian Koziel, Brian Lamacchia, Patrick Longa, et al. Sike: Supersingular isogeny key encapsulation. 2017.

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

[JMKR23] David Jacquemin, Anisha Mukherjee, Péter Kutas, and Sujoy SINHA ROY. Ready to sqi? safety first! towards a constant-time implementation of isogeny-based signature, sqisign. Cryptology ePrint Archive, Paper 2023/807, 2023. https://eprint.iacr.org/2023/807.

[JMV09] D. Jao, S. D. Miller, and R. Venkatesan. Expander graphs based on GRH with an application to elliptic curve cryptography. *Journal of Number Theory*, 129(6):1491–1504, 2009.

[JN19] Antoine Joux and Anand Kumar Narayanan. Drinfeld modules may not be for isogeny based cryptography. Cryptology ePrint Archive, Report 2019/1329, 2019. https://ia.cr/2019/1329.

[JW19] Dimitar Jetchev and Benjamin Wesolowski. Horizontal isogeny graphs of ordinary abelian varieties and the discrete logarithm problem. *Acta Arithmetica*, 187:381–404, 2019.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.

[KMP+21] Péter Kutas, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Weak instances of SIDH variants under improved torsion-point attacks. In *to appear in Advances in Cryptology - CRYPTO 2021*, Lecture Notes in Computer Science, 2021.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[Koh96] David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.

[Kun22] Sabrina Kunzweiler. Efficient computation of $(2^n, 2^n)$-isogenies. Cryptology ePrint Archive, Paper 2022/990, 2022. https://eprint.iacr.org/2022/990.

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

[KV10] Markus Kirschmer and John Voight. Algorithmic enumeration of ideal classes for quaternion orders. *SIAM Journal on Computing*, 39(5):1714–1747, 2010.

[KW19] Thorsten Kleinjung and Benjamin Wesolowski. A new perspective on the powers of two descent for discrete logarithms in finite fields. In *Proceedings of the Thirteenth Algorithmic Number Theory Symposium – ANTS-XIII*, volume 2, pages 343–352. The Open Book Series, Mathematical Sciences Publishers, 2019.

[KW22] Thorsten Kleinjung and Benjamin Wesolowski. Discrete logarithms in quasi-polynomial time in finite fields of fixed characteristic. *Journal of the American Mathematical Society*, 35(2):581–624, 2022.

[Ler23] Antonin Leroux. Verifiable random function from the deuring correspondence and higher dimensional isogenies. Cryptology ePrint Archive, Paper 2023/1251, 2023. https://eprint.iacr.org/2023/1251.

[LL93] Arjen K. Lenstra and Hendrik W. Lenstra. *The development of the number field sieve*, volume 1554. Springer Science & Business Media, 1993.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming – ICALP 2006 (part II)*, pages 144–155, 2006.

[LMO79] J.C. Lagarias, H.L. Montgomery, and A.M. Odlyzko. A bound for the least prime ideal in the Chebotarev density theorem. *Inventiones mathematicae*, 54:271–296, 1979.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60(6):43:1–43:35, 2013. Preliminary version in EUROCRYPT 2010.

[LR12] David Lubicz and Damien Robert. Computing isogenies between abelian varieties. *Compositio Mathematica*, 148(5):1483–1515, 2012.

[LS22] Antoine Leudière and Pierre-Jean Spaenlehauer. Hard homogeneous spaces from the class field theory of imaginary hyperelliptic function fields. Cryptology ePrint Archive, Report 2022/349, 2022. https://ia.cr/2022/349.

[LW17] Arjen K. Lenstra and Benjamin Wesolowski. Trustworthy public randomness with sloth, unicorn, and trx. *International Journal of Applied Cryptography*, 3(4):330–343, 2017.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Preliminary version in FOCS 2002.

[Mil86] V. S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology – CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1986.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023.

[Onu21] Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields and Their Applications*, 69:101777, 2021.

[Pet17] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017*, volume 10625 of *Lecture Notes in Computer Science*, pages 330–353. Springer, 2017.

[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: Fast-Fourier lattice-based compact signatures over NTRU. *NIST Post-Quantum Cryptography Project*, 2017.

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

[Pom87] Carl Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In David S. Johnson, Takao Nishizeki, Akihiro Nozaki, and Herbert S. Wilf, editors, *Discrete Algorithms and Complexity*, pages 119–143. Academic Press, 1987.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography Conference – TCC*, pages 145–166, 2006.

[PR23] Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Paper 2023/1766, 2023. https://eprint.iacr.org/2023/1766.

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing – STOC 2017*, pages 461–473. ACM, 2017.

[PS21] Alice Pellet-Mary and Damien Stehlé. On the hardness of the NTRU problem. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 3–35. Springer, 2021.

[PW18] Cécile Pierrot and Benjamin Wesolowski. Malleability of the blockchain's entropy. *Cryptography and Communications*, 10(1):211–233, 2018.

[PW24] Aurel Page and Benjamin Wesolowski. The supersingular endomorphism ring and one endomorphism problems are equivalent. *To appear in Advances in Cryptology – EUROCRYPT 2024*, 2024.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005.

[RK24] Farzin Renan and Péter Kutas. Sqiasignhd: Sqisignhd adaptor signature. Cryptology ePrint Archive, Paper 2024/561, 2024. https://eprint.iacr.org/2024/561.

[Rob22] Damien Robert. Some applications of higher dimensional isogenies to elliptic curves (overview of results). Cryptology ePrint Archive, Paper 2022/1704, 2022. https://eprint.iacr.org/2022/1704.

[Rob23] Damien Robert. Breaking SIDH in polynomial time. In *Advances in cryptology – EUROCRYPT 2023. Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.

[Rón92] Lajos Rónyai. Algorithmic properties of maximal orders in simple algebras over $\mathbf{Q}$. *Computational Complexity*, 2(3):225–243, 1992.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. IACR Cryptology ePrint Archive, Paper 2006/145, 2006. https://eprint.iacr.org/2006/145.

[Sch08] René Schoof. Computing arakelov class groups. *Algorithmic number theory: lattices, number fields, curves and cryptography*, 44:447–495, 2008.

[SEMR23] Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders. AprèsSQI: Extra fast verification for SQIsign using extension-field signing. Cryptology ePrint Archive, Paper 2023/1559, 2023. https://eprint.iacr.org/2023/1559.

[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, 1997.

[Sil86] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Gradute Texts in Mathematics*. Springer-Verlag, 1986.

[Sim05] Denis Simon. Solving quadratic equations using reduced unimodular quadratic forms. *Mathematics of Computation*, 74:1531–1543, 2005.

[Sim06] Denis Simon. Quadratic equations in dimensions 4, 5 and more. Preprint, 2006. See [Wat13] for a published review.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009.

[Vél71] J. Vélu. Isogénies entre courbes elliptiques. *Comptes rendus de l'Académie des Sciences, Séries A-B*, 273:A238–A241, 1971.

[Voi21] John Voight. *Quaternion Algebras*. Springer International Publishing, 2021. Graduate Texts in Mathematics, No. 288.

[Wat13] Mark Watkins. Some comments about indefinite LLL. *Diophantine Methods, Lattices, and Arithmetic Theory of Quadratic Forms*, 587(233):32, 2013.

[Wes19a] Benjamin Wesolowski. Efficient verifiable delay functions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 379–407. Springer, 2019.

[Wes19b] Benjamin Wesolowski. Generating subgroups of ray class groups with small prime ideals. In *Proceedings of the Thirteenth Algorithmic Number Theory Symposium – ANTS-XIII*, volume 2, pages 461–478. The Open Book Series, Mathematical Sciences Publishers, 2019.

[Wes20] Benjamin Wesolowski. Efficient verifiable delay functions. *Journal of Cryptology*, 33(4):2113–2147, 2020.

[Wes21] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *62nd IEEE Annual Symposium on Foundations of Computer Science – FOCS 2021*, pages 1100–1111. IEEE, 2021.

[Wes22a] Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, volume 13277 of *Lecture Notes in Computer Science*, pages 345–371. Springer, 2022.

[Wes22b] Benjamin Wesolowski. Understanding and improving the Castryck–Decru attack on SIDH. Archive ouverte HAL, Report hal-04557845, 2022. `https://hal.science/hal-04557845`.

[Wes24] Benjamin Wesolowski. Computing isogenies between finite Drinfeld modules. *To appear in IACR Communications in Cryptology*, 2024.

[WJ15] Benjamin Wesolowski and Pascal Junod. Ciphertext-policy attribute-based broadcast encryption with small keys. In Soonhak Kwon and Aaram Yun, editors, *Information Security and Cryptology – ICISC 2015*, volume 9558 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2015.

[WW20] Ryan Williams and Benjamin Wesolowski. Lower bounds for the depth of modular squaring. IACR Cryptology ePrint Archive, Report 2020/1461, 2020. `https://eprint.iacr.org/2020/1461`.

[YAJ+17] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security – 21st International Conference, FC 2017*, volume 10322 of *Lecture Notes in Computer Science*, pages 163–181. Springer, 2017.