New algorithms for the Deuring correspondence: SQISign twice as fast

Luca De Feo[1], Antonin Leroux[2,3], and Benjamin Wesolowski[4]

[1] IBM Research Europe, Zürich, Switzerland
[2] DGA
[3] LIX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris
[4] Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400, Talence, France
INRIA, IMB, UMR 5251, F-33400, Talence, France

Abstract. The Deuring correspondence defines a bijection between isogenies of supersingular elliptic curves and ideals of maximal orders in a quaternion algebra. We present a new algorithm to translate ideals of prime-power norm to their corresponding isogenies — a central task of the effective Deuring correspondence. The new method improves upon the algorithm introduced in 2021 by De Feo, Kohel, Leroux, Petit and Wesolowski as a building-block of the SQISign signature scheme. SQISign is the most compact post-quantum signature scheme currently known, but is several orders of magnitude slower than competitors, the main bottleneck of the computation being the ideal-to-isogeny translation. We implement the new algorithm and apply it to SQISign, achieving a more than twofold speed-up in key generation and signing. Verification time is not directly impacted by the change, however we also achieve a twofold speed-up through various other improvements.

In a second part of the article, we advance cryptanalysis by showing a very simple distinguisher against one of the assumptions used in SQISign. We present a way to impede the distinguisher through a few changes to the generic KLPT algorithm. We formulate a new assumption capturing these changes, and provide an analysis together with experimental evidence for its validity.

1 Introduction

Isogeny-based cryptography is one of the active areas of post-quantum cryptography. Protocols constructed from isogenies between supersingular curves are generally very compact (in particular with respect to key sizes) but less efficient than other families of schemes. A good illustration of this situation is the recent signature scheme SQISign of De Feo, Kohel, Leroux, Petit and Wesolowski [7,8]. It is, by a decent margin, the most compact post-quantum signature scheme, but signing takes a couple of seconds, which is several orders of magnitude slower than other solutions. In a way reminiscent of Galbraith, Petit and Silva [13], SQISign makes a constructive use of the Deuring correspondence, a mathematical equivalence between supersingular elliptic curves (and isogenies connecting them) and maximal orders in a quaternion algebra (and ideals connecting them). This correspondence was first introduced to isogeny-based cryptography for cryptanalytic ends [15,11,9,22], but it has since revealed its potential as a constructive tool: for signatures [13,7], for encryption schemes [6], and for key exchange [16]. These applications exploit the following idea: certain problems involving elliptic curves and isogenies are hard to solve, but their quaternion counterparts are easy. A trapdoor can be used to translate between both worlds, letting the secret holder solve problems that would otherwise be hard.

Better algorithms for the Deuring correspondence therefore have both constructive and destructive applications. The main technical contribution of [7] is a pair of algorithms to solve two of the major tasks of the computational Deuring correspondence: *translating ideals to isogenies*, and finding *quaternion ℓ-isogeny paths*. The efficiency of SQISign is mostly governed by the ideal-to-isogeny translation, while its security strongly depends on properties of the quaternion-path-finding algorithm. In this work, we improve both.

Translation ideals to isogenies. Polynomial time algorithms to translate ideals to isogenies have been known since at least 2016 [13,11], however these were hardly practical, and certainly too slow for cryptographic purposes. One of the main contributions in SQISign [7] is the design and implementation of a new practical algorithm for this task. Despite this considerable improvement, the ideal-to-isogeny translation remains the main bottleneck in SQISign.

Our first contribution is a new algorithm to translate ideals to isogenies when the norm of the ideal is a power of a small prime ℓ ($\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$, Algorithm 5). The new algorithm proves to be more efficient than the one in [7], as we demonstrate by applying it to SQISign.

One important building block, here, is an algorithm to solve norm equations inside any maximal order ($\mathsf{SpecialEichlerNorm}$, Algorithm 3), which may be of independent interest.

Security of SQISign. In [7], SQISign was proven existentially unforgeable under several computational assumptions, among which an *ad hoc* assumption on the distribution of the outputs of the quaternion-path-finding algorithm. We show that this assumption does not hold, by presenting a simple and efficient distinguisher. Although we are unable to derive a complete attack, this shows that the security proof of SQISign is invalid.

We explain how to modify the path-finding algorithm so that our distinguisher does not work anymore. We formulate a computational assumption for the modified algorithm, and analyze it via a precise study of ideals and isogenies derived from solutions of norm equations over maximal orders.

Plan. This article is organized as follows. After a brief technical overview, we introduce in Section 2 the fundamental mathematical notions and notations. In Section 3 we focus on solving norm equations inside Eichler orders and introduce our new algorithm $\mathsf{SpecialEichlerNorm}$. In Section 4, we present in full detail our new ideal-to-isogeny algorithm. The application of our method to SQISign and the associated C implementation are discussed in Section 5. Finally, in Section 6, we study the security of SQISign.

1.1 Technical overview

We now give a succinct outline of our technical contributions.

Translating ideals to isogenies. The main bottleneck in SQISign is the following task: given a maximal order $\mathcal{O}$ corresponding to the endomorphism ring of some curve E defined over a finite field $\mathbb{F}_{p^2}$, given an ideal I of norm a prime power ℓ^e corresponding to an isogeny $\varphi_I : E \to E'$ of the same degree ℓ^e, compute an *ℓ-isogeny walk* for φ_I (i.e., a sequence of isogenies of degree ℓ whose composition is φ_I).

Following [13,11], this is achieved by decomposing the isogeny $\varphi_I = \varphi_m \circ \cdots \circ \varphi_1$ into isogenies $\varphi_i : E_i \to E_{i+1}$ of smaller degree ℓ^f, where f is a system parameter depending on p. Such decomposition requires computing the endomorphism rings $\mathcal{O}_i$ of each intermediate curve E_i, a task for which SQISign (see [8, Algorithm 9]) employs a variant of the KLPT algorithm [15]. Our main technical contribution consists in replacing the full endomorphism ring $\mathcal{O}_i$ by a single well-chosen endomorphism ω_i, computed by $\mathsf{SpecialEichlerNorm}$ (Algorithm 3), a new algorithm to solve norm equations inside any maximal order.

$\mathsf{SpecialEichlerNorm}$ is not, *per se*, faster than KLPT: the true performance gain happens further down the line. Indeed, KLPT produces a representation of $\mathcal{O}_i$ by endomorphisms of degree T coprime to ℓ, where $T \approx p^{3/2}$ is another fixed system parameter. In contrast, the degree of the endomorphism ω_i output by $\mathsf{SpecialEichlerNorm}$ is only $T \approx p^{5/4}$. These endomorphisms then need to be evaluated on the torsion subgroup $E_i[T]$, something that can only be done efficiently when T is a smooth integer and $E_i[T]$ is defined over a small degree extension of $\mathbb{F}_{p^2}$.

All these facts combined create a strong constraint $\ell^f T | (p^{2d} - 1)$ for some small integer d, and in fact SQISign even forces $d = 1$, for maximum efficiency. Primes p such that $p^2 - 1$ has such a large smooth factor are extremely difficult to find, and thus the overall efficiency of SQISign comes from a balancing act between f, the smoothness of T, and the computational resources available to search for p. In this light, it is clear that moving from $T \approx p^{3/2}$ to $T \approx p^{5/4}$ constitutes a big improvement as one may hope to find better "SQISign-friendly" primes, as we do here. In fact, even using the same prime p as in [7], our new algorithm leads to a (smaller) improvement because we can ignore some factors of T and use a smaller endomorphism degree $T' | T$.

Security of SQISign. The SQISign signature scheme is obtained by applying the Fiat–Shamir transform [12] to an interactive identification scheme. While it is straightforward to prove that the identification scheme is a 2-special sound proof of knowledge of an endomorphism (a statement closely related to the knowledge of the endomorphism ring [21,1]), proving zero-knowledge turns out to be much more difficult.

Indeed, De Feo *et al.* could not construct a statistically indistinguishable simulator, and had to resort instead to a computational assumption [8, Problem 2] stating that the the ideals output by the quaternion-path-finding algorithm SigningKLPT [8, Algorithm 5] are indistinguishable from uniformly random ideals of the same norm. They provided evidence for the assumption by showing that the output of SigningKLPT is uniformly distributed in an exponentially large set whose size does not depend on the secret.

We show that their assumption does not hold. Indeed, we prove that the ideal I output by SigningKLPT is contained into an ideal of norm 2 that is not uniformly distributed. This condition that can be easily checked by looking at the first step of the associated 2-isogeny walk φ_I, immediately implying that SQISign signatures can be distinguished with non-negligible advantage from random 2-isogeny walks of fixed length.

This bias is due to the fact that RepresentInteger, a sub-algorithm of SigningKLPT, solves norm equations inside a suborder of a *special maximal order* $\mathcal{O}_0$ (see definition in Section 2.1). We present in Section 3.1 a variant of RepresentInteger fixing the bias, then we provide both theoretical and empirical evidence that the newly defined distribution cannot be distinguished by intersecting with ideals of norm 2^k for small k.

2 Preliminaries

Throughout this work, p is a prime number and $\mathbb{F}_{p^2}$ is a finite field of size p^2.

A negligible function $f : \mathbb{Z}_{>0} \to \mathbb{R}_{>0}$ is a function whose growth is bounded by $O(x^{-n})$ for all $n > 0$. In the analysis of a probabilistic algorithm, we say that an event happens with *overwhelming probability* if its probability of failure is a negligible function of the length of the input.

We say that a distinguishing problem is hard when any probabilistic polynomial-time distinguisher has a negligible advantage with respect to the length of the instance. Two distributions are computationally indistinguishable if their associated distinguishing problem is hard.

2.1 Mathematical background on the Deuring Correspondence

We now briefly present mathematical notions used in this article.

Elliptic curves, isogenies and endomorphisms Elliptic curves are abelian varieties of dimension 1, and isogenies are non-constant morphisms between them. The degree of an isogeny is its degree as a rational map. An isogeny is *separable* if its degree is equal to the size of its kernel. Let E be an elliptic curve. To any finite subgroup G of E, one can associate a separable isogeny $\varphi : E \to E/G$ with kernel $\ker \varphi = G$, and this isogeny is unique up to an isomorphism of the target. Isogenies can be computed from their kernels with

Vélu's formula [19]. An isogeny from a curve E to itself is an endomorphism of E. In positive characteristic, the ring $\mathrm{End}(E)$ of all endomorphisms of E is isomorphic either to an order in a quadratic imaginary field or a maximal order in a quaternion algebra. In the first case, the curve is said to be *ordinary* and otherwise it is *supersingular*. We focus on the supersingular case in this article. The book from J. Silverman [18] is a good reference for more details on elliptic curves and isogenies.

Supersingular elliptic curves over $\overline{\mathbb{F}}_p$ always have a model defined over $\mathbb{F}_{p^2}$. Furthermore, this model can always be chosen so that all its endomorphisms are also defined over $\mathbb{F}_{p^2}$. This property is preserved by the $\mathbb{F}_{p^2}$-isogeny class, and in this article, we work in one such class.

Quaternion algebras, orders and ideals. The endomorphism rings of supersingular elliptic curves over $\mathbb{F}_{p^2}$ are isomorphic to maximal orders of $B_{p,\infty}$, the quaternion algebra ramified at p and ∞. We fix a basis $1, i, j, k$ of $B_{p,\infty}$, satisfying $i^2 = -q$, $j^2 = -p$ and $k = ij = -ji$ for some integer q. The canonical involution of conjugation sends an element $\alpha = a + ib + jc + kd$ to $\overline{\alpha} = a - (ib + jc + kd)$. A *fractional ideal* I is a $\mathbb{Z}$-lattice of rank four. We denote by $n(I)$ the *norm* of I as the largest rational number such that $n(\alpha) \in n(I)\mathbb{Z}$ for any $\alpha \in I$. Given fractional ideals I and J, if $J \subseteq I$ then the index $[I : J]$ is defined to be the order of the finite quotient group I/J. We define the ideal conjugate $\overline{I} = \{\overline{\alpha}, \alpha \in I\}$. An order $\mathcal{O}$ is a subring of $B_{p,\infty}$ that is also a fractional ideal. An order is called *maximal* when it is not contained in any other larger order. The left order of a fractional ideal is defined as $\mathcal{O}_L(I) = \{\alpha \in B_{p,\infty} \mid \alpha I \subset I\}$ and similarly for the right order $\mathcal{O}_R(I)$. Then I is said to be an $(\mathcal{O}_L(I), \mathcal{O}_R(I))$-ideal or a left $\mathcal{O}_L(I)$-ideal. A fractional ideal is *integral* if it is contained in its left order, or equivalently in its right order; we refer to integral ideals hereafter as ideals. An ideal can be written as $I = \mathcal{O}_L(I)\alpha + \mathcal{O}_L(I)n(I) = \mathcal{O}_L(I)\langle\alpha, n(I)\rangle$ for some $\alpha \in \mathcal{O}_L(I)$. Two left $\mathcal{O}$-ideals I and J are equivalent if there exists $\beta \in B_{p,\infty}^\times$, such that $I = J\beta$. For a given $\mathcal{O}$, this defines equivalences classes of left $\mathcal{O}$-ideals, and we denote the set of such classes by $\mathrm{Cl}(\mathcal{O})$. We will reuse the following notation from [7]: for any ideal K and any $\alpha \in B_{p,\infty}^\times$, we write $\chi_K(\alpha) = K\overline{\alpha}/n(K)$. Ideals equivalent to K are precisely the ideals $\chi_K(\alpha)$ with $\alpha \in K \setminus \{0\}$.

The Deuring Correspondence. In [10], Deuring made the link between elliptic curves and quaternion algebras over $\mathbb{Q}$ by showing that the endomorphism ring of a supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$ is isomorphic to a maximal order in $B_{p,\infty}$. Fix a supersingular elliptic curve E_0, and an order $\mathcal{O}_0 \simeq \mathrm{End}(E_0)$. The curve/order correspondence allows one to associate each outgoing isogeny $\varphi : E_0 \to E_1$ to an integral left $\mathcal{O}_0$-ideal, and every such ideal arises in this way (see [14] for instance). Through this correspondence, the ring $\mathrm{End}(E_1)$ is isomorphic to the right order of this ideal. This isogeny/ideal correspondence is defined in [20], and in the separable case, it is explicitly given as follows.

Definition 1. *Given I an integral left $\mathcal{O}_0$-ideal coprime to p, we define the I-torsion $E_0[I] = \{P \in E_0(\overline{\mathbb{F}}_{p^2}) : \alpha(P) = 0$ for all $\alpha \in I\}$. To I, we associate the separable isogeny φ_I of kernel $E_0[I]$. Conversely given an isogeny φ, the corresponding ideal is defined as $I_\varphi = \{\alpha \in \mathcal{O}_0 : \alpha(P) = 0$ for all $P \in \ker(\varphi)\}$.*

We summarize properties of the Deuring correspondence in Table 1, borrowed from [7].

Supersingular j-invariants over $\mathbb{F}_{p^2}$ $j(E)$ (up to galois conjugacy)	Maximal orders in $B_{p,\infty}$ $\mathcal{O} \cong \mathrm{End}(E)$ (up to isomorpshim)
(E_1, φ) with $\varphi : E \to E_1$	I_φ integral left $\mathcal{O}$-ideal and right $\mathcal{O}_1$-ideal
$\theta \in \mathrm{End}(E_0)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$n(I_\varphi)$
$\hat{\varphi}$	$\overline{I_\varphi}$
$\varphi : E \to E_1, \psi : E \to E_1$	Equivalent Ideals $I_\varphi \sim I_\psi$
Supersingular j-invariants over $\mathbb{F}_{p^2}$	$\mathrm{Cl}(\mathcal{O})$
$\tau \circ \rho : E \to E_1 \to E_2$	$I_{\tau\circ\rho} = I_\rho \cdot I_\tau$
N-isogenies (up to isomorphism)	$\mathrm{Cl}(\mathfrak{O})$, with Eichler order $\mathfrak{O}$ of level N

Table 1: The Deuring correspondence, a summary [7].

Special extremal order. A *special extremal order* is an order $\mathcal{O}_0$ in $B_{p,\infty}$ which contains a suborder of the form $R + jR$, where $R = \mathbb{Z}[\omega] \subset \mathbb{Q}(i)$ is a quadratic order and ω has minimal discriminant. When $p \equiv 3 \bmod 4$, we have the special extremal order $\mathcal{O}_0 = \left\langle 1, \frac{i+j}{2}, \frac{1+k}{2} \right\rangle$, with $i^2 = -1$. It is isomorphic to the elliptic curve E_0 of j-invariant 1728. For the rest of the paper, we fix this special extremal order $\mathcal{O}_0$, with subring $\mathbb{Z}[\omega]$, and the corresponding elliptic curve E_0.

2.2 The SQISign protocol

We now present SQISign [7], the main target for applying the present work. The signature scheme is based on an interactive identification protocol, made non-interactive through the classic Fiat–Shamir transform. The initial setup and key generation are as follows.

$\mathsf{setup} : \lambda \mapsto \mathsf{param}$ Pick a prime number p and a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$, with known special extremal endomorphism ring $\mathcal{O}_0$. Select an odd smooth number D_c of λ bits and $D = 2^e$ where e is larger than the diameter of the supersingular 2-isogeny graph.

$\mathsf{keygen} : \mathsf{param} \mapsto (\mathsf{pk} = E_A, \mathsf{sk} = \tau)$ Pick a random isogeny walk $\tau : E_0 \to E_A$, leading to a random elliptic curve E_A. The public key is E_A, and the secret key is the isogeny τ.

To prove knowledge of the secret τ, the prover engages in the following Σ-protocol with the verifier.

Commitment The prover generates a random (secret) isogeny walk $\psi : E_0 \to E_1$, and sends E_1 to the verifier.

Challenge The verifier sends the description of a cyclic isogeny $\varphi : E_1 \to E_2$ of degree D_c to the prover.

Response From the isogeny $\varphi \circ \psi \circ \hat{\tau} : E_A \to E_2$, the prover constructs a new isogeny $\sigma : E_A \to E_2$ of degree D such that $\hat{\varphi} \circ \sigma$ is cyclic, and sends σ to the verifier.

Verification The verifier accepts if σ is an isogeny of degree D from E_A to E_2 and $\hat{\varphi} \circ \sigma$ is cyclic. They reject otherwise.

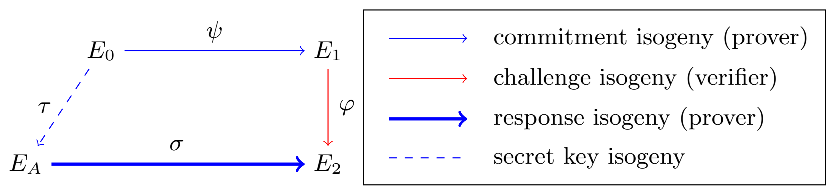


Fig. 1: A picture of the identification protocol

2.3 Algorithms from previous works

We will rely upon or mention several algorithms existing in the literature. In the interest of conciseness, we will use the algorithms below without describing them. The interested reader will find pseudo-code for most of them in [7,8], the others are standard:

- $\mathsf{Cornacchia}(M)$: either find x, y such that $x^2 + Y^2 = M$ or output $\perp$.
- $\mathsf{RepresentInteger}_{\mathcal{O}_0}(M)$, given $M \in \mathbb{N}$ with $M > p$, finds $\gamma \in \mathcal{O}_0$ of norm dividing M.
- $\mathsf{EquivalentPrimeIdeal}(\mathsf{I})$, given a left $\mathcal{O}_0$-ideal I, finds the smallest equivalent left $\mathcal{O}_0$-ideal of prime norm.
- $\mathsf{EichlerModConstraint}(I, \gamma)$, given a left $\mathcal{O}_0$-ideal I of norm N, and $\gamma \in \mathcal{O}_0$ of norm n coprime with N, finds $(C_0 : D_0) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ such that $\mu_0 = j(C_0 + \omega D_0)$ satisfies $\gamma\mu_0 \in \mathbb{Z} + I$.
- $\mathsf{StrongApproximation}_{\mathcal{N}}(N, C_0, D_0)$, given a prime N and $C_0, D_0 \in \mathbb{Z}$, finds $\mu = \lambda\mu_0 + N\mu_1 \in \mathcal{O}_0$ of smallest norm in $\mathcal{N}$, with $\mu_0 = j(C_0 + \omega D_0)$ and $\mu_1 \in \mathcal{O}_0$. When $\mathcal{N} = \{d \in \mathbb{N}, d|D\}$ for some $D \in \mathbb{N}$, we simply write $\mathsf{StrongApproximation}_D$. We will also use the notation $\ell^\bullet = \{\ell^e, e \in \mathbb{N}\}$.

Remark 1. Variants of RepresentInteger and StrongApproximation (denoted as FullXxx) will be presented as Algorithms 1 and 2 in Section 3. Their formulations differ only slightly from the one introduced in [7], but we will argue these modifications are necessary.

Remark 2. The algorithm EquivalentPrimeIdeal above finds the smallest possible solution. We sometimes use its randomized version (written RandomEquivalent-PrimeIdeal) where we choose a random output among a set of solutions of small norm.

3 Solving norm equations inside maximal orders

In this section, we consider the following problem: given a maximal order $\mathcal{O}$ of $B_{p,\infty}$, and a set of integers $\mathcal{N}$, find and element $\beta \in \mathcal{O}$ with $n(\beta) \in \mathcal{N}$. The relevant case for our application is the following: we fix an integer T, and $\mathcal{N}$ is the set of divisors of T^2. Algorithms to solve this task are presented in [8, Section 5.1], but they find solutions that are not well distributed in $\mathcal{O}$: they always fall in a particular sublattice, inducing a bias that affects both the efficiency and security of its applications. We present how to eliminate this bias.

For ease of exposition, we fix $p \equiv 3 \bmod 4$, and the special extremal order $\mathcal{O}_0 = \langle 1, i, \frac{i+j}{2}, \frac{1+k}{2} \rangle$, with $i^2 = -1$ and $\omega = i$. Most of what follows remains true for other primes and special extremal orders under small adjustments.

The method underlying Algorithm 3 follows the blueprint introduced in [8, Section 5.1]: find an Eichler order of small prime level embedded inside both $\mathcal{O}$ and the special extremal order $\mathcal{O}_0$ (considered as an implicit parameter of the algorithm below) and solve the norm equation inside this Eichler order. As a first step, we study in Section 3.1 the problem of solving norm equations in the full maximal order $\mathcal{O}_0$ (rather than the convenient suborder $\mathbb{Z}[i,j]$ as in [15,7]). This study, and the resulting new algorithms, will prove useful for Algorithm 3 (as pointed out in Remark 4) and also prevents a simple distinguisher against a problem relating to the zero-knowledge property of SQISign; the latter point is further investigated in Section 6.

3.1 Special extremal order case: exploiting the full order

We first deal with norm equations in the special extremal order $\mathcal{O}_0$. In this case, algorithms from [15,7] only find solutions in the suborder $\mathbb{Z}[i,j]$, exploiting the orthogonal basis $\langle 1, i, j, k \rangle$. This suborder has index 4 inside $\mathcal{O}_0$, so many potential solutions are excluded, a source of complications for some applications. In this section, we describe how to heuristically obtain well-distributed solutions in $\mathcal{O}_0$.

The norm form of $\langle 1, i, j, k \rangle$ is $f : (x, y, z, t) \mapsto x^2 + y^2 + p(z^2 + t^2)$ and the usual way to find a representation of a given integer M (a method common to both RepresentInteger and StrongApproximation) is to choose z, t (possibly with some additional conditions) until $M - p(z^2+t^2)$ is a prime represented by x^2+y^2, then use Cornacchia's algorithm [3] to solve $x^2 + y^2 = M - p(z^2 + t^2)$. Solutions in the full order $\mathcal{O}_0$ can be found from solutions in $\mathbb{Z}[i,j]$ thanks to Lemma 1. Let $g : (x, y, z, t) \mapsto (x + t/2)^2 + (y + z/2)^2 + p((z/2)^2 + (t/2)^2)$ be the norm form of $\mathcal{O}_0 = \langle 1, i, \frac{i+j}{2}, \frac{1+k}{2} \rangle$.

Lemma 1. *An integer M is represented by g if and only if $4M$ is represented by f with $x = t \mod 2$ and $y = z \mod 2$.*

Proof. If we have $M = (x+t/2)^2 + (y+z/2)^2 + p((z/2)^2 + (t/2)^2)$, we have that $4M = (2x+t)^2 + (2y+z)^2 + p(z^2+t^2)$. Thus, an integer M is represented by g (with solution (x, y, z, t)) if and only if $4M$ is represented by f with a solution $(x', y', z', t') = (2x+t, 2y+z, z, t)$ satisfying $x' = t' \mod 2$ and $y' = z' \mod 2$.

Algorithm 1 $\mathsf{FullRepresentInteger}_{\mathcal{O}_0}(M)$

Input: $M \in \mathbb{Z}$ such that $M > p$
Output: $\gamma = x + yi + z\frac{i+j}{2} + t\frac{1+k}{2}$ with $n(\gamma)$ dividing M.
1: Set $m' = \lfloor \sqrt{\frac{4M}{p}} \rfloor$ and sample a random integer $z' \in [-m', m']$.
2: Set $m'' = \lfloor \sqrt{\frac{4M}{p} - z'^2} \rfloor$ and take a random t' inside $[-m'', m'']$. Set $M' = 4M - pf(z', t')$.
3: If $\mathsf{Cornacchia}(M') = \perp$ go back to the previous step. Otherwise set $x', y' = \mathsf{Cornacchia}(M')$.
4: If $x' \neq t' \mod 2$ or $z' \neq y' \mod 2$ then go back to Step 2.
5: Set $\gamma = (x + iy + jz + kt)/2$ and repeat $\gamma = \gamma/2$ until $\gamma/2 \notin \mathcal{O}_0$.
6: **return** γ.

From Lemma 1 and the original algorithm for RepresentInteger from [15], we derive FullRepresentInteger in Algorithm 1. Just as RepresentInteger is heuristically believed to return well-distributed solutions in $\mathbb{Z}[i,j]$, the variant FullRepresentInteger is believed to return well-distributed solutions in $\mathcal{O}_0$, thanks to Lemma 1. A rigorous study of the distribution of the output appears to be hard because of the Cornacchia subroutine whose success depends on the factorisation pattern of its input. This question is further investigated in Section 6, with heuristic and experimental evidence.

The running time of FullRepresentInteger is essentially the same as the running time of RepresentInteger, divided by the success probability of the condition $\gamma/2 \in \mathcal{O}_0$. Heuristically, this constant is $2/3$: the solutions $(x', y', z', t') \mod 2$ of the equation $x'^2 + y'^2 + p(z'^2 + t'^2) = 0 \mod 4$ are $(0,0,0,0)$, $(1,1,1,1)$, $(1,0,0,1)$, $(0,1,1,0)$, $(1,0,1,0)$, and $(0,1,0,1)$. Among these 6, there are 2 that do not lead to $\gamma/2 \in \mathcal{O}_0$: the solutions $(1,0,1,0)$ and $(0,1,0,1)$. The heuristic analysis is corroborated by the behaviour of our implementation.

Remark 3. One might wonder why we do not propose to swap x' and y' when the constraint is not satisfied. Undeniably, this would be a good way to ensure that each set of values x', y', z', t' leads to a solution. However, this introduces a distinguishable bias, precisely of the kind investigated in Section 6.

The StrongApproximation algorithm can also be modified to find solutions in the full order $\mathcal{O}_0$ with Lemma 1. In Algorithm 2, we present FullStrongApproximation as a generic reduction to StrongApproximation. Thanks to Lemma 1, properties of the distribution of the output of FullStrongApproximation directly follow from properties of the distribution of StrongApproximation. As in the case of FullRepresentInteger, we expect the running time of FullStrongApproximation to be equal to the running time of StrongApproximation multiplied by 3/2.

Algorithm 2 $\mathsf{FullStrongApproximation}_{\mathcal{N}}$

Input: A prime number N, two values $C, D \in \mathbb{Z}$.
Output: $\mu \in \mathcal{O}_0$ such that $2\mu = \lambda\mu_0 + N\mu_1$ with $\mu_0 = jC + kD$, $\mu_1 \in \mathcal{O}_0$, and $n(\mu) \in \mathcal{N}$.

1: Let $4\mathcal{N} = \{4n \mid n \in \mathcal{N}\}$.
2: Set $\mu' = \mathsf{StrongApproximation}_{4\mathcal{N}}(N, C, D)$.
3: If $\mu' \notin 2\mathcal{O}_0$, go back to Step 2.
4: **return** $\mu = \mu'/2$.

3.2 Norm equations in generic maximal order: the algorithm

We are now ready to describe an algorithm to solve equations inside generic maximal orders. For simplicity, we restrict the description to the case that will be useful for our new variant of *ideal to isogeny translation* (see Section 4). Thus, we require that the algorithm outputs elements of norm dividing T^2 for some parameter T and that the solution β satisfies the following constraint: given the additional input K, a left $\mathcal{O}$-ideal of norm ℓ coprime to T, we need that $\beta \notin \mathbb{Z} + K$ (see Step 7 in Algorithm 3). A justification for this constraint is provided in Section 4.1.

Algorithm 3 $\mathsf{SpecialEichlerNorm}_T(\mathcal{O}, K)$

Input: $\mathcal{O}$ a maximal order and K a left $\mathcal{O}$-ideal of norm ℓ.
Output: $\beta \in \mathcal{O} \setminus (\mathbb{Z} + K)$ of norm dividing T^2.

1: Compute $I = I(\mathcal{O}_0, \mathcal{O})$.
2: Set $L = \mathsf{RandomEquivalentPrimeIdeal}(I)$, $N = n(L)$ and compute α s.t $L = I\alpha$.
3: Compute $K' = \alpha^{-1}K\alpha$
4: Compute $(C : D) = \mathsf{EichlerModConstraint}(L, 1)$.
5: Enumerate all possible solutions of $\mu = \mathsf{FullStrongApproximation}_{T^2}(N, C, D)$ until $\mu \notin \mathbb{Z} + K'$. If it fails go back to Step 2.
6: **return** $\beta = \alpha\mu\alpha^{-1}$.

Proposition 1. *Under plausible heuristics, the algorithm* $\mathsf{SpecialEichlerNorm}_T$ *is correct and terminates with constant probability when* $T > p^{5/4}$.

Proof. Under the heuristics from [15], we know that the value $N = n(L)$ has size approximately $p^{1/2}$ when L is the output of $\mathsf{RandomEquivalentPrimeIdeal}$. Then, it was proven in [7] that $\mathsf{EichlerModConstraint}$ is correct and terminates. We argued correctness and termination with constant probability for $\mathsf{FullStrongApproximation}$ in Section 3.1. Now, we introduce the following heuristic assumption: the output μ of $\mathsf{FullStrongApproximation}$ satisfies $\mu \notin \mathbb{Z} + K'$ with probability approximately $[\mathcal{O}_0 : \mathbb{Z} + K']^{-1}$ (which is the probability one would get if $\mu \in \mathcal{O}_0$ were drawn uniformly in a large enough ball). Even though the precise distribution of μ appears difficult to analyse, this heuristic is plausible since the algorithm $\mathsf{FullStrongApproximation}$ seems to constrain possible values of μ only locally at N and T, both coprime with ℓ. The proof is concluded by the fact that $\mathsf{FullStrongApproximation}_{T^2}(N, \cdot)$ finds at least one solution with constant probability when $T^2 > pN^3 \approx p^{5/2}$ (see [8, Section 5.3]). Thus, we have proven heuristic termination. For correctness, it is easy to see that $n(\mu) = n(\beta)$ and so the correctness of $\mathsf{FullStrongApproximation}$ proves that $n(\beta)|T^2$. Since $\mu \in \mathbb{Z} + L = \mathcal{O}_0 \cap \mathcal{O}_R(L)$ where $L = I\alpha$, we can show that $\mathcal{O} = \alpha\mathcal{O}_R(L)\alpha^{-1}$. Since $\mu \notin \mathbb{Z} + K'$, then $\alpha\mu\alpha^{-1} \notin \mathbb{Z} + \alpha K \alpha^{-1} = \mathbb{Z} + K$.

Remark 4. Note that the new heuristic introduced in the proof of Proposition 1 would not have held if we had used the $\mathsf{StrongApproximation}$ from [15]. Indeed, the solutions of $\mathsf{StrongApproximation}$ lie in $\mathbb{Z}\langle 1, i, j, k\rangle$ which is contained in the Eichler order $(\mathbb{Z}+\mathcal{O}_0\langle 1+i, 2\rangle)$. Thus, when $K \cap J = \mathcal{O}_0\langle 1+i, 2\rangle \cap J$, the condition $\mu \notin \mathbb{Z} + K$ can never be satisfied. This is why it is important to use our new variant $\mathsf{FullStrongApproximation}$.

Failures. Algorithm 3 may fail when the heuristics used in the proof of Proposition 1 are not accurate. In particular, the problematic case is when the size of the output of $\mathsf{RandomEquivalentPrimeIdeal}(I)$ is bigger than expected. This situation occurs when there exists a representative in the ideal class of I with norm considerably smaller than $p^{1/2}$ (see the bounds on the norm of elements in a Minkowski-reduced basis of a lattice from [15, Section 3.1]). There are only a negligible number of problematic maximal orders but we still need to handle those few bad cases. The simplest solution to avoid that problem altogether is to increase the size of T. We have the absolute bound $N < p$ and so we can ensure termination by taking $T > p^2$. However, we want the bound of T to be as tight as possible and so this is not a suitable solution for us. There is a way to handle the bad cases without increasing T but it does not always work. Let us assume for the rest of this paragraph that there exists $J \sim I$ with $n((J) \ll p^{1/2}$. $\mathsf{FullStrongApproximation}(M, \cdot)$ does not strictly require its input M to be prime (see [8, Section 6.3]) and so $\mathsf{FullStrongApproximation}$ can be modified to work with $n(J)$ in input instead of N. We can also run $\mathsf{EichlerModConstraint}$ with J instead of L. Since $n(J) \ll p^{1/2}$ it should be possible to complete the computation when $T \approx p^{5/4}$. However, we may be in trouble with the additional condition $\mu \notin \mathbb{Z} + K$. Indeed, if $J \subset K$, this constraint will never be satisfied because $\mu \in \mathbb{Z} + J$. If $n(J)$ is coprime with ℓ, this will not happen but it can occur when $\ell | n(J)$.

In summary, $\mathsf{SpecialEichlerNorm}_{T^2}$ cannot terminate on input $\mathcal{O}, K$ with $T \approx T^{5/4}$ when $\mathcal{O}$ is connected to $\mathcal{O}_0$ with an ideal of small norm included in K. We will explain in Section 4.2 how to overcome this obstacle.

4 A new algorithm for ideal to isogeny translation

The goal of this section is to introduce our new algorithm to perform the *ideal-to-isogeny translation* required in computations of the effective Deuring correspondence. We start with an informal overview of how our new method manages to be more efficient than previous ones. A more detailed cost analysis tailored to SQISign will be provided in Section 5.1.

The goal is, given as input an $\mathcal{O}$-ideal I of norm D and a curve E with $\mathrm{End}(E) \cong \mathcal{O}$, to compute the kernel of the D-isogeny $\varphi_I : E \to \star$. For this task, SQISign introduces [8, Algorithm 9], a generalization of [13, Algorithm 2]. Its principle is simple: evaluate the endomorphisms corresponding to elements of I on a basis of the D-torsion, then solve a discrete logarithm to find a generator of $\ker \varphi_I$.

For this algorithm to be efficient, it is necessary that the evaluation points are defined over an extension of $\mathbb{F}_p$ of small degree. In [8], this is solved by decomposing the ideal I as a chain of ideals I_i of smaller norm D_i: small enough that the D_i-torsion is defined over $\mathbb{F}_{p^2}$. The idea is then to apply the technique introduced in [13], enhanced with several tricks, to translate each I_i.

It is not obvious, however, how to evaluate endomorphisms of all the I_i at arbitrary points. This task is easy in special cases: for example, the explicit correspondence between the maximal order $\mathcal{O}_0 = \langle 1, i, (i+j)/2, (1+ij)/2 \rangle$ and the endomorphism ring of $E_0 : y^2 = x^3 + x$ was leveraged in [13]. Instead, for ideals I_i of a generic order $\mathcal{O}$, the ideal-to-isogeny transaltion of [8] first computes an isogeny walk ϕ_K of degree T, coprime to D_i, from a special curve E_0 to E (see [8, Algorithm 7]), then evaluates it at the points of order D_i. The repeated evaluation of such isogenies of large degree is the bottleneck of the computation, consequently the size and smoothness of T greatly affect performance. In SQISign, ϕ_K is computed using a variant of the KLPT algorithm [15], and thus $T > p^{3/2}$.

Here we introduce $\mathsf{IdealToIsogenyEichler}_D$, a new variant of $\mathsf{IdealToIsogeny}_D$ that only requires one well-chosen endomorphism of $\mathrm{End}(E)$ to perform the translation above. The endomorphism is computed by $\mathsf{SpecialEichlerNorm}_T$ and translated to an isogeny from E to itself. We will show in Lemma 2 that, assuming an additional property, the kernel of φ_I can be found via a single evaluation at a point of order D. Like in [8], we will use T-isogenies, with T coprime to D, and, thanks to Proposition 1, $T \approx p^{5/4}$. This reduction in size affords us a lot more flexibility in the choice of p. Several compromises can be made on the size and smoothness of T and D_i; in any case, our new method speeds up SQISign key generation and signing, as we will demonstrate in Section 5.3.

4.1 Ideal to isogeny translation

The goal of this section is to introduce a new $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ algorithm. The specifications are exactly the same as those of [8, Algorithm 9]. Our algorithm builds upon three sub-routines: $\mathsf{IdealToIsogeny}_D$ that is [13, Algorithm 2] (performing the ideal-to-isogeny translation on $\mathcal{O}_0$-ideals by performing operations on the D-torsion), $\mathsf{SpecialEichlerNorm}$ presented in Algorithm 3 (replacing KLPT) and $\mathsf{IdealToIsogenyEichler}_{\ell^f}$ that we introduce below in Algorithm 4 (and is analogous to [8, Algorithms 7 and 8]).

For the rest of this section, we fix the prime p and we take f as the largest exponent such that $\ell^f | (p^2-1)/2$. We also fix a parameter T coprime with ℓ dividing p^2-1 and assume that $T > p^{5/4}$. $\mathsf{IdealToIsogenyEichler}_{\ell^f}$ describes a way to translate $\mathcal{O}$-ideals of norm ℓ^f using one endomorphism evaluation. Intuitively, the idea is to choose an endomorphism θ such that $P, \theta(P)$ constitutes a basis of the ℓ^f-torsion for some point P. Then, a generator of the kernel of the desired isogeny can be obtained as a linear combination of $P, \theta(P)$ whose coefficients can be found using classical linear algebra. For efficiency, we will take θ of norm dividing T^2 so we can represent θ using two isogenies φ_1, φ_2 of degree n_1, n_2 dividing T such that $\theta = \varphi_2 \circ \hat{\varphi}_1$.

Algorithm 4 $\mathsf{IdealToIsogenyEichler}_{\ell^f}(\mathcal{O}, I, J, \varphi_J, P)$

Input: I a left $\mathcal{O}$-ideal of norm ℓ^f, an $(\mathcal{O}_0, \mathcal{O})$-ideal J of norm $\ell^\bullet$ and $\varphi_J : E_0 \to E$ the corresponding isogeny, the generator $P \in E[\ell^f]$ of $\ker \varphi_K$ s.t $\hat{\varphi}_J = \varphi_{K'} \circ \varphi_K$.
Output: φ_I of degree ℓ^f
1: Set $K = \overline{J} + \mathcal{O}\ell^f$.
2: Compute $\theta = \mathsf{SpecialEichlerNorm}_T(\mathcal{O}, K + \mathcal{O}\ell)$ of norm dividing T^2.
3: Select $\alpha \in I$ s.t $I = \mathcal{O}\langle \alpha, \ell^f \rangle$.
4: Compute C, D s.t. $\alpha \cdot (C + D\theta) \in K$ and $\gcd(C, D, \ell) = 1$ using linear algebra.
5: Take any $n_1 | T$ and $n_2 | T$ s.t $n_1 n_2 = n(\theta)$. Compute $H_1 = \mathcal{O}\langle \theta, n_1 \rangle$ and $H_2 = \mathcal{O}\langle \overline{\theta}, n_2 \rangle$.
6: Compute $L_i = [J]^* H_i$, and $\varphi_i = [\varphi_J]_* \mathsf{IdealToIsogeny}_{n_i}(L_i)$ for $i \in \{1, 2\}$.
7: Compute $Q = \hat{\varphi}_2 \circ \varphi_1(P)$.
8: Compute φ_I of kernel $\langle [C]P + [D]Q \rangle$.
9: **return** φ_I.

Before stating that Algorithm 4 is correct and terminates we need a preliminary lemma. For any $\mathcal{O}$-ideal K of degree ℓ^f we write $K_e = K + \ell^e \mathcal{O}$ for $1 \leq e \leq f$.

Lemma 2. *Let E be a supersingular curve and $\mathcal{O} \cong \mathrm{End}(E)$ be a maximal order. Let K and I be two $\mathcal{O}$-ideals of norm ℓ^f, let $\theta \in \mathcal{O} \setminus (\mathbb{Z} + K)$ have norm* *coprime to ℓ. Let $E[K] = \langle P \rangle$, then $E[I] = \langle [C]P + [D]\theta(P) \rangle$ iff* $\gcd(C, D, \ell) = 1$ *and $\alpha \circ (C + D\theta) \in K$ for any α s.t $I = \mathcal{O}\langle \alpha, \ell^f \rangle$.*

Proof. Let us take $Q = [C]P + [D]\theta(P)$ and assume that $E[I] = \langle Q \rangle$. Since Q has order ℓ^f, it is clear that $\gcd(C, D, \ell) = 1$. Let us take $\alpha \in I$ such that $I = \mathcal{O}\langle \alpha, \ell^f \rangle$. This condition is equivalent to $\ker \alpha \cap E[\ell^f] = E[I]$. We want to show that $\alpha \circ (C + D\theta) \in K$ i.e that $\alpha \circ (C + D\theta)(P) = 0$ which is straightforward since $E[I] = \langle [C]P + [D]\theta(P) \rangle$. Conversely, let us assume that $\gcd(C, D, \ell) = 1$ and $\alpha \circ (C + D\theta) \in K$ for any α s.t $I = \mathcal{O}\langle \alpha, \ell^f \rangle$. Taking such an α we get that $\alpha \circ (C + D\theta)(P) = 0$ which must imply that $[C]P + [D]\theta(P) = \lambda Q$ for some $\lambda \in \mathbb{Z}$ and Q such that $E[I] = \langle Q \rangle$. If we show that $\gcd(\lambda, \ell^f) = 1$ then we will have shown our result as P and $\theta(P)$ have order ℓ^f. Let us assume this is not the case. We have $\gcd(\lambda, \ell^f) = \ell^{e_0}$ for $e_0 > 0$. Then the point $P_0 = [\ell^{f-e_0}]$ of order ℓ^{e_0} satisfies $[D]\theta(P_0) = [-C]P_0$. Since $\gcd(C, D, \ell) = 1$, we must have $\gcd(D, \ell) = 1$ and so $\theta(P_0) = [\mu]P_0$ where $\mu = -C/D \mod \ell^{e_0}$. This proves that we have $\theta \in \mathbb{Z} + K_{e_0} \subset \mathbb{Z} + K_1$ which is a contradiction with our intitial assumption. Hence, $\gcd(\lambda, \ell^f) = 1$ and we have proven the result.

Proposition 2. *Under plausible heuristics,* $\mathsf{IdealToIsogenyEichler}_{\ell^f}$ *is correct and terminates with overwhelming probability.*

Proof. By Proposition 1, we have that $\mathsf{EichlerNormEq}$ is correct and terminates with overwhelming probability under plausible heuristics. Apart from the execution of $\mathsf{SpecialEichlerNorm}$, the only step that neeeds justification is Step 4. First, it is not clear that such a solution must always exist. In fact, the existence of such C, D follows from $\theta \notin \mathbb{Z} + (K + \ell\mathcal{O})$. This condition implies that $P, \theta(P)$ form a basis of $E[\ell^f]$, for otherwise we would have $[\ell^{f-1}]P = [\ell^{f-1}]\theta(P)$ and so $\theta \in \mathbb{Z} + (K + \ell\mathcal{O})$, since $E[K] = \langle P \rangle$. When it exists, a solution C, D can easily be found using linear algebra in a similar fashion to $\mathsf{EichlerModConstraint}$.

Correctness follows from Lemma 2. When we assimilate the endomorphisms α and $[C] + [D]\theta$ in $\mathrm{End}(E)$ with their image through the isomorphism between $\mathrm{End}(E)$ and $\mathcal{O}$, we get that the composition $\alpha \circ (C + D\theta)$ becomes the multiplication of the quaternion elements $\alpha \cdot (C + D\theta)$. Thus, by Lemma 2, the values C, D computed at Step 4 are such that $\ker \varphi_I = \langle [C] + [D]\theta(P) \rangle$. By definition of H_1, H_2, we have that $\theta = \hat{\varphi}_2 \circ \varphi_1$ and this concludes the proof that the output isogeny is indeed the one corresponding to I through the Deuring Correspondence.

Now we are ready for our full algorithm, which is basically made of sequential executions of $\mathsf{IdealToIsogenyEichler}_{\ell^f}$. For simplicity, we assume in Algorithm 5 that the ideal input to $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ has norm ℓ^e, where $e = fg$ for some $g \in \mathbb{N}$. The general case is easily derived.

Proposition 3. $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ *is correct and terminates with overwhelming probability.*

Proof. It is easily verified that the $\mathcal{O}_i, I_i, J_i, \varphi_J \circ \varphi_I, P_i$ are correct inputs to $\mathsf{IdealToIsogenyEichler}_{\ell^f}$. Thus, the result follows from Proposition 2

Algorithm 5 $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}(I, J, \varphi_J)$

Input: I a left $\mathcal{O}$-ideal of norm ℓ^e with $e = fg$, an $(\mathcal{O}_0, \mathcal{O})$-ideal J of norm $\ell^\bullet$ and $\varphi_J : E_0 \to E$ the corresponding isogeny
Output: φ_I of degree ℓ^e.
1: Set $J_i = J$, $I_i = I + \ell^f \mathcal{O}$, $I'_i = I_i^{-1} I$, $\mathcal{O}_i = \mathcal{O}$.
2: Set φ_i of degree ℓ^f as the isogeny such that $\hat{\varphi}_J = \varphi' \circ \varphi_i$
3: Set $\varphi_I = [1]_E$ and $E_i = E$.
4: **for** $i \in [1, g]$ **do**
5: Compute $P_i \in E_i[\ell^f]$ s.t $\ker \varphi_i = \langle P \rangle$.
6: Compute $\varphi_{I_i} = \mathsf{IdealToIsogenyEichler}_{\ell^f}(\mathcal{O}_i, I_i, J_i, \varphi_J \circ \varphi_I, P_i)$.
7: Set $\varphi_i = \hat{\varphi}_{I_i}$, $\varphi_I = \varphi_{I_i} \circ \varphi_I$ and E_i is the codomain of φ_{I_i}.
8: Set $J_i = J_i \cdot I_i$, $\mathcal{O}_i = \mathcal{O}_L(I'_i)$, $I_i = I'_i + \ell^f \mathcal{O}_i$ and $I'_i = I_i^{-1} I'_i$.
9: **end for**
10: **return** φ_I.

Below, we explain more precisely how to perform Step 7 of $\mathsf{IdealToIsogeny}$-$\mathsf{Eichler}_{\ell^f}$. The technical details were left out of the description in Algorithm 4 to clarify the explanations but they are important for an efficient implementation. Throughout this entire section, we have avoided the issues of potential failures of $\mathsf{SpecialEichlerNorm}$ that were mentioned at the end of Section 3.2. We will discuss in Section 4.2 how to perform the computation in this eventuality.

Endomorphism evaluation. For Step 7 of $\mathsf{IdealToIsogenyEichler}_{\ell^f}$ it is required to evaluate the endomorphisms $\theta = \hat{\varphi}_2 \circ \varphi_1$ after the two isogenies φ_1, φ_2 have been computed. One might assume that it suffices to push P through φ_1 and then do the same through $\hat{\varphi}_2$. This apparently simple algorithm is not so easy to implement. The first problem lies with signs. Efficient isogeny algorithms are using x-only arithmetic which imply that we can only evaluate isogenies up to signs. This is problematic as the ultimate goal is to compute $[C]P + [D]\theta(P)$. Solving this issue requires to evaluate several other points through φ_1, φ_2 and there does not seem to be another easy way to remove the ambiguity. The second issue is with the dual computation in itself. For an isogeny φ of degree T and kernel $\langle P \rangle$, computing $\hat{\varphi}(R)$ for some point R would first require to compute $\varphi(Q)$ where Q is of order T and orthogonal to P to get $\ker \hat{\varphi}$, before using this kernel to compute $\hat{\varphi}(R)$. In the context of SQISign where T-isogenies have kernel made of two points, this is already 2 T-isogeny computations and 3 evaluations (see Section 5.1 for a more detailed account on operation estimates). Together with the computation of $\varphi_1(P)$, we have a total of 3 T-isogeny computations and 4 evaluations and this is without whatever would be required to lift the sign ambiguity.

Targetting the application to $\mathsf{IdealToISogenyEichler}_{\ell^f}$, we present in Algorithm 6, a method that requires only 2 T-isogeny computations and 5 evaluations together with a few discrete logarithms to evaluate an endomorphism of the form $C + D\theta$ where $\theta = \hat{\varphi}_2 \circ \varphi_1$ on a point of power-smooth order (this performs both Step 7 and Step 8 of Algorithm 4 at once). Our method is entirely based on x-

only arithmetic and requires a basis P, Q of the ℓ^f-torsion. The main principle is to express $\varphi_1(P)$ as a linear combination of $\varphi_2(P), \varphi_2(Q)$ and see that $\hat{\varphi}_2 \circ \varphi_1(P)$ is a multiple of the linear combination of P, Q with the same coefficients. When dealing with x-only arithmetic we need also to compute $\varphi_2(P+Q)$ to perform the discrete log computations. Finally, to lift the ambiguity (the linear combination that we obtain is only up to sign) we use the trace of $\theta = \hat{\varphi}_2 \circ \varphi_1$ (which can be computed by expressing θ in the basis $\langle 1, i, j, k \rangle$). In the basis P, Q, the action of θ can be seen as a matrix of $\mathbb{M}_2(\mathbb{Z}/\ell^f\mathbb{Z})$. This matrix is essentially the one we obtain with the coefficient of the two discrete logarithms and so it suffices to check the value of the trace to lift any sign ambiguity.

In Algorithm 6 we assume a function $\mathsf{xBIDIM}_{\ell^f}(x(R), x(P), x(Q), x(P+Q))$ computes the scalars a, b such that $x(R) = x([a]P + [b]Q)$ for points P, Q of order ℓ^f. It has complexity $O(f)$.

Algorithm 6 $\mathsf{EndomorphismEvaluation}_{\ell^f}(\varphi_1, \varphi_2, C, D, t, P)$

Input: Two isogenies $\varphi_1, \varphi_2 : E \to E'$, scalars C, D, the trace $t = \mathrm{tr}(\hat{\varphi}_2 \circ \varphi_1)$ and a point P of order ℓ^f

Output: $[C]P + [D]\hat{\varphi}_2 \circ \varphi_1(P)$

1: Compute Q such that P, Q is a basis of $E[\ell^f]$ and compute $P + Q$.
2: Compute $x(\varphi_1(P)), x(\varphi_1(Q)), x(\varphi_2(P)), x(\varphi_2(Q)), x(\varphi_2(P+Q))$.
3: Compute $x_1, x_2 = \mathsf{xBIDIM}(x(\varphi_1(P)), x(\varphi_2(P)), x(\varphi_2(Q)), x(\varphi_2(P+Q))$ and $x_3, x_4 = \mathsf{xBIDIM}(x(\varphi_1(Q)), x(\varphi_2(P)), x(\varphi_2(Q)), x(\varphi_2(P+Q)))$.
4: Change the signs of $(x_1, x_2), (x_3, x_4)$ until $(x_1 + x_4)\deg\varphi_2 = t \mod \ell^f$.
5: Set $a = C + x_1 D$ and $b = x_2 D$.
6: Compute $R = [a]P + [b]Q$.
7: **return** R.

Remark 5. For the signature, one needs to compute a canonical representation of the isogeny obtained in output of $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$. The method described in [8, Section 8.5] with the algorithms Compression/Decompression it to compute deterministically a basis at each intermediate curve E_i and then use the linear combination leading to the kernel of the next step as a representation. Since one point can always be obtained as the kernel of the previous isogeny, it suffices to pick the second point in a deterministic manner. Typically, this would be done in Step 1 of EndomorphisEvaluation in the choice of Q and so the value a, b would be used to derive the final representation.

4.2 Handling special failure cases

In the analysis proposed at the end of Section 3.2, we explained that there are some inputs $\mathcal{O}, K$ for which the computation of $\mathsf{SpecialEichlerNorm}_T(\mathcal{O}, K)$ will fail if $T \approx p^{5/4}$. If one of the $\mathcal{O}_i$ is one of those bard orders, the execution of $\mathsf{IdealToIsogenyEichler}_{\ell^f}$ at the i-th iteration in $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ will fail.

Since we cannot afford to increase the size of T, we need another way to handle the failures. Short of finding a new (and better) methods, there are basically two options: revert to the method of [7] to perform the translation, or use a special extremal order other than $\mathcal{O}_0$ with $\mathsf{SpecialEichlerNorm}$.

Applying the $\mathsf{IdealToIsogeny}_{\ell f}$ from [7]. At first glance going back to the old method might seem like an odd thing to do. However, the failure cases for $\mathsf{SpecialEichlerNorm}$ are actually good cases for the method from [7] because there is an ideal of norm $M \ll p^{1/2}$ connecting $\mathcal{O}_0$ and $\mathcal{O}$. As we explained, this is only a bad thing for $\mathsf{SpecialEichlerNorm}$ because we have an additional constraint with the ideal K but $\mathsf{IdealToIsogeny}_{\ell f}$ does not suffer from the same limitation. $\mathsf{IdealToIsogeny}_{\ell f}$ relies on the KLPT algorithm that will succeed to find an element of norm T^2 if $T \approx pM$. Hence, when $M < p^{1/4}$, we can hope to make it work with $T \approx p^{5/4}$. However, there is an obvious range of degrees $p^{1/4} \ll M \ll p^{1/2}$ where this solution will not work. This is why in practice, we will use the second method described below.

Using another special extremal order. The bad property depends on the special extremal order $\mathcal{O}_0$ that we use. In practice, when $p = 3 \mod 4$, it is standard in the literature to use the maximal extremal order $\langle 1, i, \frac{1+k}{2}, \frac{i+j}{2} \rangle$ because it contains the quadratic order $\mathbb{Z}[i]$ (which is basically the best quadratic order we can hope for). However, this canonical example is not the only maximal order matching the definition of extremal orders given in [15]. We recall that a maximal order containing a given quadratic order $\mathfrak{O}$ exists when p is an inert prime in the quadratic imaginary field associated to $\mathfrak{O}$. Even if other quadratic orders will not be as efficient as $\mathbb{Z}[i]$, the complexity of $\mathsf{SpecialEichlerNorm}$ is logarithmic in disc $\mathfrak{O}$ and so we can expand the range of choices without affecting too much the performances. Thus, we can gather a small list of good candidates for $\mathcal{O}_0$ and enumerate through that list until we find one that does not have the bad property. To prove that this idea works, we need to make sure that a maximal order $\mathcal{O}$ will not have the bad property with all the extremal orders. Unfortunately, we do not have a definitive proof of this fact and or reduced to make it a heuristic assumption. Boneh and Love [17] showed that maximal quaternion orders admitting embeddings of small quadratic orders are far from one another. While this conveys the right idea, their bound in [17, Proposition 4.5] is too loose to help us. In practice, switching to another maximal order seems to work well in our implementation.

Meet-in-the-middle trick. A different idea is to replace the degree T^2 in $\mathsf{SpecialEichlerNorm}$ by $T^2\varepsilon$ where ε is a small smooth number. Then, θ will be decomposed as $\hat{\varphi}_2 \circ \eta \circ \varphi_1$ where η has degree ε and can be computed with a meet-in-the-middle search between the codomains of φ_1 and φ_2. The cost being in $\sqrt{\varepsilon}$, we can only apply this idea for very small values of ε. We did not implement this idea, as it appears to be less efficient in practice.

5 Parameters and Implementation for SQISign

We now present our methodology to set parameters for SQISign using our new ideal-to-isogeny algorithm, and report on our implementation. We start with a method to give a rough estimate of the relative efficiency of two parameter choices. Based on these estimates, we report on our search for new primes better suited to our new algorithm. Finally, we give benchmarks for our implementation and compare with the original SQISign implementation.

For the rest of this section we let p be a prime such that $\ell^f T \mid (p^2-1)$, where T is smooth and coprime with ℓ. Following [7], we will take $\ell = 2$, as this leads to the fastest verification and simplest implementation overall. It is an interesting question whether other choices for ℓ could lead to useful compromises. With the choice of $\ell = 2$, the authors from [7] advised to take a σ of degree 2^{1000}.

5.1 Cost estimate

It was already observed in [7] that algebraic operations over $\mathbb{F}_{p^2}$ make up for most of the cost of SQISign: up to $\approx 90\%$ in our experiments. It is thus reasonable to ignore computations over the quaternions and linear algebra, and focus on these. Ideally we would count the number of $\mathbb{F}_{p^2}$-operations performed for each choice of parameters, however this is already difficult given the complexity of the algorithms. Instead, we will use a much coarser metric based on four indicators.

We are only going to compare [8, Algorithm 9] and Algorithm 5. Both algorithms decompose an ideal of norm ℓ^e into ideals of smaller norm. The former decomposes into ideals of norm $\ell^{2f+\Delta}$ for some constant Δ, which are then translated to isogenies by [8, Algorithm 8]. The latter decomposes into ideals of norm ℓ^f, which are translated by Algorithm 4. Both sub-algorithms consist mostly of isogeny computations of degree T and ℓ^f. For each of them, we will count:

- (T_c) How many isogenies of degree T are *computed*;
- (T_e) On how many points the isogenies of degree T are *evaluated*;
- (ℓ_c) How many isogenies of degree ℓ^f are computed/evaluated;
- (Δ_c) How many meet-in-the-middle searches for isogenies of degree ℓ^Δ are performed (this is exclusive to [8, Algorithm 8]).

The costs of T_c and T_e depend on the factorization of T. Instead of using the full factorization, we will only base our estimate on a bound B such that all prime factors of T are $< B$. Using [2], the costs of computing and evaluating an isogeny of prime degree n grow as $\sqrt{n}$ (ignoring logarithmic factors), we will thus multiply T_c and T_e by $\sqrt{B}$. Since ℓ is small, the cost of computing and evaluating an isogeny of degree ℓ^f grows as $f\log(f)$ (ignoring the dependency in ℓ), we shall thus multiply ℓ_c by this factor. Finally, the meet-in-the-middle requires to compute all $\sqrt{\ell^\Delta}$ isogenies, so we multiply Δ_c by $\sqrt{\ell^\Delta}$.

Given an ideal of norm ℓ^e, SQISign will call [8, Algorithm 8] $\approx e/(2f+\Delta)$ times, whereas our new method will call Algorithm 5 $\approx e/f$ times. For this reason, we shall divide all counts by $2f+\Delta$ and f, respectively.

Summarizing, for [8, Algorithm 8] we will use the following 4-valued estimator:

$$(T_c\sqrt{B}\,,\, T_e\sqrt{B}\,,\, \ell_c f\log(f)\,,\, \sqrt{\ell^{\Delta}}\Delta_c)/(2f+\Delta), \tag{1}$$

where the division is applied component-wise. For Algorithm 5, given that it does not use a meet-in-the-middle search, we will instead use

$$(T_c\sqrt{B}/f\,,\, T_e\sqrt{B}/f\,,\, \ell_c\log(f)). \tag{2}$$

Original method. For convenience, Algorithm 7 reproduces [8, Algorithm 8] without modifications. Some of the steps therein are quite vague, so we also refer to the code at `https://github.com/SQISign/sqisign`.

Algorithm 7 $\mathsf{IdealToIsogeny}_{\ell^{2f+\Delta}}(I, J, K, \varphi_J, \varphi_K)$ [8, Algorithm 8]

Input: I a left $\mathcal{O}_0$-ideal of norm dividing $T^2\ell^{2f+\Delta}$, an $\mathcal{O}_0$-ideal in J containing I of norm dividing T^2, and an ideal $K \sim J$ of norm a power of ℓ, as well as φ_J and φ_K.

Output: $\varphi = \varphi_2 \circ \theta \circ \varphi_1 : E_1 \to E_2$ of degree $\ell^{2f+\Delta}$ such that $\varphi_I = \varphi \circ \varphi_J$, $L \sim I$ of norm dividing T^2 and φ_L.

0: Write $\varphi_J, \varphi_K : E_0 \to E_1$.
1: Let $I_1 = I + \ell^f\mathcal{O}_0$.
2: Let $\varphi_1' = \mathsf{IdealToIsogeny}_{\ell^f}(I_1)$.
3: Let $\varphi_1 = [\varphi_J]_*\varphi_1' : E_1 \to E_3$.
4: Let $L = \mathsf{KLPT}_T(I)$.
5: Let $\alpha \in K$ such that $J = \chi_K(\alpha)$.
6: Let $\beta \in I$ such that $L = \chi_I(\beta)$.
7: Let $\gamma = \beta\alpha/n(J)$. We have $\gamma \in K$, $\bar{\gamma} \in L$, and $n(\gamma) = T^2\ell^{2f+\Delta}n(K)$.
8: Let $H_1 = \langle \gamma, n(K)\ell^f T\rangle$. We have $\varphi_{H_1} = \psi_1 \circ \varphi_1 \circ \varphi_K : E_0 \to E_5$, where ψ_1 has degree T.
9: Let $H_2 = \langle \overline{\gamma}, \ell^f T\rangle$. We have $\varphi_{H_2} = \rho_2 \circ \psi_2 : E_0 \to E_6$, where ψ_2 has degree T and ρ_2 has degree ℓ^f.
10: Find $\eta : E_5 \to E_6$ of degree ℓ^{Δ} with meet-in-the-middle.
11: Let $\varphi_2 \circ \theta = [\hat{\psi}_1]_*\hat{\rho}_2 \circ \eta : E_3 \to E_2$ and $\psi_1' = [\hat{\varphi}_2 \circ \eta]_*\hat{\psi}_1$
12: **return** $\varphi = \varphi_2 \circ \theta \circ \varphi_1$, L and $\psi_1' \circ \psi_2$.

The operation count for Algorithm 7 goes as follows: Step 3 is 2 T_e (push $\ker\varphi_1$ through φ_J) and 1 ℓ_c (compute φ_1), 1 T_c for Step 8 (compute ψ_1), Step 9 is 1 T_e, 1 T_c (compute ψ_2 and $\ker\rho_2$) and 1 ℓ_c (compute φ_2), Step 10 is 1 Δ_c, Step 11 is 2 T_e (compute $\ker\hat{\psi}_1$), 2 ℓ_c (push $\ker\hat{\psi}_1$ through $\rho_2\circ\eta$), 1 T_c and 1 T_e (compute ψ_1' and $\ker\hat{\varphi}_2$), 1 ℓ_c (compute φ_2) and 1 Δ_c (compute θ). Thus a total of 3 T_c, 6 T_e, 2 Δ_c and 5 ℓ_c.

New Method. Step 7 requires to solve a DLP instance over the ℓ^f-torsion and we overestimate the complexity by saying that this is equivalent to 1 ℓ_c operation (asymptotically it is the same cost but the DLP is faster in practice). We obtain the following count: 2 T_c for Step 6, 5 T_e and 1 ℓ_c for Step 7 (see Algorithm 6), Step 8 is 1 ℓ_c. Overall, we get 2 T_c, 5 T_e and 2 ℓ_c.

5.2 New prime search

Recall that the main advantage of our new ideal-to-isogeny algorithm is to decrease T from $\sim p^{3/2}$ to $\sim p^{5/4}$. Primes p such that $\ell^f T \mid (p^2-1)$ for such large T are rare, and thus a search must be performed in order to instantiate SQISign. Following [7], we focus on primes of ≈ 256 bits, which offer ≈ 128 bits of classical security. In [7], a prime

$$\begin{aligned} p+1 &= 2^{33}\cdot 5^{21}\cdot 7^2\cdot 11\cdot 31\cdot 83\cdot 107\cdot 137\cdot 751\cdot 827\cdot 3691\cdot 4019\cdot 6983 \\ &\quad \cdot 517434778561\cdot 26602537156291\,, \\ p-1 &= 2\cdot 3^{53}\cdot 43\cdot 103^2\cdot 109\cdot 199\cdot 227\cdot 419\cdot 491\cdot 569\cdot 631\cdot 677\cdot 857\cdot 859 \\ &\quad \cdot 883\cdot 1019\cdot 1171\cdot 1879\cdot 2713\cdot 4283 \end{aligned}$$

is recommended, giving $f = 33$ and a $T > 2^{393}$ that is 2^{13}-smooth. We shall call it p_{6983}, after the largest factor in T. This prime can be used both for the old and the new method, however in our new method we can discard some of the largest factors of T, getting down to a $T' > 2^{333}$ that is 2^{11}-smooth. Knowing that $\Delta = 14$ in [7], we can already use our estimator to compare the two methods. The values are reported in Table 4. Based on this metric, it appears that the new method could be slightly faster than the old one.

However, a less stringent requirement on T makes the search for p considerably easier, it is thus natural to look for a new prime that is better adapted to our method. p_{6983} was found using an XGCD-based method described in [8, Appendix C], which we used to find more primes. In the meantime, more algorithms to find primes such that p^2-1 is smooth were introduced in [4,5], unfortunately only the sieve of [4], looking for primes of the form $p = 2x^n - 1$, adapts well to the requirement of having $2^f \mid (p^2-1)$ for some moderately large f. Indeed we can modify this method by forcing $2^{\lceil f/n \rceil} \mid x$. Trying to do the same in the sieve of [5] leads to a search space too small to yield any primes.

Regardless of the method we use, given that we look for a smaller T, we can choose to either increase f or decrease the smoothness bound B on T. Looking at estimator (2), it appears that we can divide the first two entries by 2 in one of two ways: multiplying f by 2, or dividing B by 4. We experimented with both. We used the method of [4] to look for primes $p = 2^{61}x^4 - 1$, sieving the whole interval $x \in [2^{47}, 2^{49}[$ in approximately 360 cpu-days. We found 398 integers such that p^2-1 has a 2^{11}-smooth odd factor of more than 330 bits, of which 15 were prime (see Table 2); none of them has a large enough 2^{10}-smooth factor.

143189100303149	369428710635531	391443251922757	411099446409699
424067696488337	431716591494287	491224940548057	491531434028942
512391149388477	512583833108361	514414280000642	515727186701509
548396183941255	550470785518701	562456538440551	

Table 2: List of integers $x \in [2^{47}, 2^{49}[$ such that $2^{61}x^4 - 1$ is prime and $x^4(2^{15}x-1)(2^{15}x+1)(2^{30}x^2+1)$ contains a 2^{11}-factor $> 2^{330}$.

Using the XGCD method of [8], we found that we could obtain primes with $f \approx 64$ and $B = 2^{12}$ at a reasonable cost. The best candidate we found, which we name p_{3923}, has 254 bits and

$$\begin{aligned} p+1 &= 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \\ &\qquad \cdot 3923 \cdot 62731 \cdot 96362257 \cdot 3924006112952623\,, \\ p-1 &= 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157 \cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563 \cdot 599 \\ &\qquad \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 10069\,. \end{aligned}$$

Despite the slightly larger smoothness bound, we found that p_{3923} performs better in practice than primes of the form $2^{61}x^4 - 1$, probably owing to the large power of 3, which contributes favorably to T-isogeny computations.

Reporting the estimator values for p_{3923} in Table 3, we see that applying our new algorithm to the new prime yields a significant gain during T-isogeny computations and meet-in-the-middle at the cost of a modest loss during ℓ^f-isogeny computations. Since the former tend to affect performance much more than the latter, in practice, we expect our new method to compare favorably to the old one. We will see in the next section that, in practice, the gain is even larger than predicted by our rough estimator. Finding more accurate estimators to guide the prime search in SQISign is an interesting problem for future research.

algorithm	p	$\log(p)$	f	B	T_c	T_e	ℓ_c	Δ_c	estimator
Old	p_{6983}	256	33	2^{13}	3	6	5	2	$(3.4, 6.8, 10.4, 3.2)$
New	p_{6983}	256	33	2^{11}	2	5	2	–	$(2.7, 6.9, 10.1)$
New	p_{3923}	254	65	2^{12}	2	5	2	–	$(2.0, 4.9, 12.0)$

Table 3: Operation estimates for several variants of *ideal-to-isogeny* translation. B is the smoothness bound of T.

Other changes. Having a smaller T forces some other changes to SQISign's challenge and commitment steps. To get λ bits of security, the commitment must have degree $T' \geq 2^{2\lambda}$, and the challenge degree $D_c \geq 2^{\lambda}$ coprime to T'. The authors of [7] could take $T'D_c = T \approx p^{3/2} \approx 2^{3\lambda}$. To optimize verification, they chose to D_c as smooth as possible, i.e., $D_c = 3^{53}5^{21}$.

However, with a smaller T, we can no longer have $T = T'D_c$. Instead, we incorporate some powers of ℓ in D_c; incidentally, this happens to increase verification speed. For p_{3923}, we take $D_c = 2^{65}3^{40}$, which is a marked improvement over $D_c = 3^{53}5^{21}$. Of course, one could incorporate powers of 2 to D_c also with p_{6983}, but $p_{6983} + 1$ only contains a factor 2^{33}, so verification with p_{3923} still beats p_{6983}.

In fact, at the cost of increasing the signer's work, it is possible to take D_c as a power of ℓ, which could further decreases verification time. The concrete gain for the instantiation with p_{3923} will be the difference between a 2^{64}-isogeny computation and a 3^{40} computation. This is a marginal gain compared to the cost for the signer (at least several additional executions of $\mathsf{IdealToIsogenyEichler}_{\ell f}$) so we chose not to pursue this idea further.

5.3 C implementation

We took the official SQISign implementation[5], and we substituted our new ideal-to-isogeny algorithm, plus some other minor improvements. In particular, we implemented the compression method described in [8, Section 8.5] for verification, which, along with using powers of 2 in the challenge degree D_c, explains the improved verification. Our code is available at `https://github.com/SQISign/sqisign2`. We ran benchmarks on the same platform as [7]: an Intel Core i7-6700 clocked at 3.40 GHz with turbo-boost deactivated. The results are in Table 4. With our improvements, and moving from p_{6983} to p_{3923}, we observe a more than two-fold speed-up in all operations.

		SQISign/p_{6983}			New/p_{6983}			New/p_{3923}		
		Keygen	Sign	Verify	Keygen	Sign	Verify	Keygen	Sign	Verify
Mcycles	1st quartile	1,918	7,722	135	2,569	6,673	82	717	3,645	64
	median	1,950	7,828	142	3,081	6,718	86	741	3,685	65
	3rd quartile	2,008	7,959	148	3,973	6,797	88	780	3,743	68
ms	1st quartile	562	2,266	39	754	1,958	24	210	1,070	19
	median	572	2,297	42	904	1,971	25	218	1,081	19
	3rd quartile	589	2,335	44	1,166	1,994	26	229	1,098	20

Table 4: Performance of SQISign in millions of cycles and in milliseconds. Statistics over 100 runs for key generation and signature, and over 250 runs for verification.

6 Cryptanalysis

In this section, we present a distinguisher against one of the computational assumption underlying the security of SQISign. This distinguisher does not lead to an attack on the signature scheme but it invalidates the claimed hardness of the problem. We present a fix to protect the scheme against the distinguisher and propose further theoretical analysis and experimental results to argue that a modified assumption holds. More concretely, the assumption introduced in [7] to prove that SQISign's underlying identification scheme is honest-verifier zero-knowledge relies on the hardness of distinguishing between a random isogeny of degree D and a random element from a particular subset of the D-isogenies. The exact formulation is given as Problem 1 below. In Section 6.1, we show that

[5] `https://github.com/SQISign/sqisign`.

this subset (as defined in [7]) satisfies one of the problematic property pointed out in [8, Appendix B.2], where it was argued that such a property would lead to a distinguisher. Fortunately, a slight change of $\mathsf{SigningKLPT}$, explained in Section 6.1, seems to be enough to remove the problem. In Section 6.2, we analyse the new assumption more precisely, to argue that it does not suffer from a similar weakness.

Before getting to our contributions, we give a quick summary of some of the relevant content from [7] regarding the zero-knowledge property. We start in Algorithm 8 with the description of the $\mathsf{SigningKLPT}$ algorithm from [7].

Algorithm 8 $\mathsf{SigningKLPT}(I, I_\tau)$

Input: I_τ a left $\mathcal{O}_0$-ideal and right $\mathcal{O}$-ideal of norm N_τ, and I, a left $\mathcal{O}$-ideal.
Output: $J \sim I$ of norm ℓ^e, where e is fixed.

1: Compute $K = \mathsf{EquivalentRandomEichlerIdeal}(I, N_\tau)$
2: Compute $K' = [I_\tau]^* K$ and set $L = \mathsf{EquivalentPrimeIdeal}(K')$, $L = \chi_{K'}(\delta)$ for $\delta \in K'$ with $N = n(L)$. Set $e_0 = e_0(N)$ and $e_1 = e - e_0$.
3: Compute $\gamma = \mathsf{RepresentInteger}_{\mathcal{O}_0}(N\ell^{e_0})$.
4: Compute $(C_0 : D_0) = \mathsf{IdealModConstraint}(L, \gamma)$.
5: Compute $(C_1 : D_1) = \mathsf{EichlerModConstraint}(\mathbb{Z} + I_\tau, \gamma, \delta)$.
6: Compute $C = \mathsf{CRT}_{N_\tau,N}(C_0, C_1)$ and $D = \mathsf{CRT}_{N_\tau,N}(D_0, D_1)$. If $\ell^e p(C^2 + D^2)$ is not a quadratic residue, go back to Step 3.
7: Compute $\mu = \mathsf{StrongApproximation}_{\ell^\bullet}(NN_\tau, C, D)$ of norm ℓ^{e_1}
8: Set $\beta = \gamma\mu$.
9: **return** $J = [I_\tau]_* \chi_L(\beta)$.

In SQISign, the output J of $\mathsf{SigningKLPT}$ is converted into the corresponding isogeny σ, and the signature is a representation of this isogeny. The zero-knowledge property is proved assuming the hardness of Problem 1, described below. This assumption formalises that σ is indistinguishable from a random isogeny of the same degree.

The structure of this isogeny is analysed in [7], with more details in [8, Lemma 13] and [8, Fig. 3], reproduced here as Lemma 3 and Fig. 2, for the reader's convenience.

Lemma 3. *Let $L \subset \mathcal{O}$ and $\beta \in L$ be as in steps 2, 8 respectively of Algorithm 8. The isogeny σ corresponding to the output J of Algorithm 8 is equal to $\sigma = [\tau]_*\iota$, where ι is an isogeny of degree ℓ^e verifying $\beta = \hat{\iota} \circ \varphi_L$.*

Before giving a precise statement of the distinguishing problem, we need to recall some notation from [7]. For what follows, we keep the notation introduced in Lemma 3, Fig. 2, and Algorithm 8. For a given ideal L of norm N, we consider $\mathcal{U}_{L,N_\tau}$ as the set of all isogenies ι computed as in Lemma 3 from elements $\beta = \gamma\mu \in L$ where γ is any possible output of the non-deterministic function $\mathsf{RepresentInteger}_{\mathcal{O}_0}(N\ell^{e_0(N)})$, and μ is computed as in Algorithm 8.

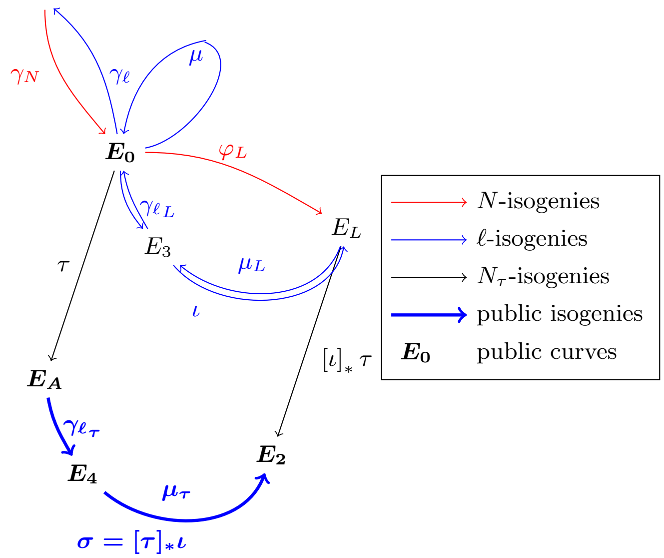


Fig. 2: Analysis of Algorithm 8 under the Deuring correspondence

For an equivalence class $\mathcal{C}$ in $\mathrm{Cl}(\mathcal{O}_0)$ we write $\mathcal{U}_{\mathcal{C},N_\tau}$ for $\mathcal{U}_{L,N_\tau}$ where $L = \mathsf{EquivalentPrimeIdeal}(\mathcal{C})$ (recall that $\mathsf{EquivalentPrimeIdeal}$ is deterministic).

Definition 2. $\mathcal{P}_{N_\tau} = \bigcup_{\mathcal{C} \in \mathrm{Cl}(\mathcal{O}_0)} \mathcal{U}_{\mathcal{C},N_\tau}$

For $D \in \mathbb{N}$ and a supersingular curve E, we define $\mathtt{Iso}_{D,j(E)}$ as the set of cyclic isogenies of degree D, whose domain is a curve inside the isomorphism class of E. When $\mathcal{P}$ is a subset of $\mathtt{Iso}_{D,j(E)}$ and $\tau : E \to E'$ is an isogeny with $\gcd(\deg \tau, D) = 1$, we write $[\tau]_* \mathcal{P}$ for the subset $\{[\tau]_* \varphi \mid \varphi \in \mathcal{P}\}$ of $\mathtt{Iso}_{D,j(E')}$. Finally, we denote by $\mathcal{K}$ a probability distribution on the set of cyclic isogenies whose domain is E_0, representing the distribution of SQISign private keys. With these notations, we define the following computational problem:

Problem 1. Let p be a prime, and D a smooth integer. Let $\tau : E_0 \to E_A$ be a random isogeny drawn from $\mathcal{K}$, and let N_τ be its degree. Let $\mathcal{P}_{N_\tau} \subset \mathtt{Iso}_{D,j_0}$ as in Definition 2, and let O_τ be an oracle sampling random elements in $[\tau]_* \mathcal{P}_{N_\tau}$. Let $\sigma : E_A \to \star$ of degree D where either

1. σ is uniformly random in $\mathtt{Iso}_{D,j(E_A)}$;
2. σ is uniformly random in $[\tau]_* \mathcal{P}_{N_\tau}$.

The problem is, given $p, D, \mathcal{K}, E_A, \sigma$, to distinguish between the two cases with a polynomial number of queries to O_τ.

6.1 An attack on SQISign's zero-knowledge assumption

Our distinguisher for Problem 1 is a consequence of the limitations pointed out in Section 3.1 and it occurs specifically when $\ell = 2$ (which is the value used in [7] and in our implementation), so for the rest of this section and the next we take $D = 2^e$. Lemma 4 and the resulting Proposition 4 links the observations of Section 3.1 to a property on the set $[\tau]_* \mathcal{P}_{N_\tau}$.

Lemma 4. *Let L be an $\mathcal{O}_0$-ideal of norm N and let γ, be an element in $\mathcal{O}_0$ of norm $N\ell^e$ for some prime N. Let us take $\mu \in \mathcal{O}_0$ such that $\beta = \gamma\mu \in L$. If $\gamma \in \langle 1, i, j, k\rangle$, then $\chi_L(\beta) \subset \mathcal{O}_0\langle 1+i, 2\rangle$.*

Proof. We have $\gamma \in \langle 1, i, j, k\rangle \subset \mathcal{O}_0\langle 1+i, 2\rangle$. Now, $\chi_L(\beta) = \mathcal{O}_0\langle \overline{\mu\gamma}, 2^e\rangle$, hence $\overline{\mu\gamma} \in \mathcal{O}_0\overline{\gamma} \subseteq \mathcal{O}_0\langle 1+\bar{i}, 2\rangle = \mathcal{O}_0\langle 1+i, 2\rangle$, which proves the proposition.

Proposition 4. *Let $D = 2^e$ and τ, N_τ be as in 1 and let the set $\mathcal{P}_{N_\tau}$ be defined from Algorithm 8. There exists an isogeny $\iota_0 \in \mathtt{Iso}_{2,j(E_0)}$ such that every $\iota \in \mathcal{P}_{N_\tau}$ can be decomposed as $\iota = \iota_1 \circ \iota_0$ where ι_1 is an isogeny of degree 2^{e-1}.*

Proof. Let J be the ideal corresponding to $\sigma \in [\tau]_*\mathcal{P}_{N_\tau}$. By definition of $\mathcal{P}_{N_\tau}$, ι corresponds to the ideal $\chi_L(\gamma\mu)$. It is easily verified that L, γ, μ verify the requirements of Lemma 4 and that $\gamma \in \langle 1, i, j, k\rangle$ since it is a possible output of $\mathsf{RepresentInteger}_{\mathcal{O}_0}$. Thus, we can apply Lemma 4 and we get that $\chi_L(\beta) \subset \mathcal{O}_0\langle 1+i, 2\rangle$. This proves the result by taking ι_0 to be the isogeny corresponding to the ideal $\mathcal{O}_0\langle 1+i, 2\rangle$.

Thus, Proposition 4 implies that, when defined as in Definition 2, the family $\mathcal{P}_{N_\tau}$ satisfies one of the special properties introduced in [8, Appendix B.2]. Indeed, we obtain that $I^1_\tau = \{\iota_1 \text{ of degree } 2, \text{ s.t } \exists \iota_2,\ \iota_2 \circ \iota_1 \in \mathcal{P}_{N_\tau}\}$ has size 1 (instead of 3), and so a trivial distinguisher can be built against Problem 1 simply by looking at the distribution of the first step of σ.

A fix against the attack. To block the distinguisher, it suffices to use the $\mathsf{FullRepresentInteger}$ variant that we described in Algorithm 1 during Step 3 of Algorithm 8, instead of $\mathsf{RepresentInteger}$. This alternate version of the algorithm was designed specifically to produce solutions γ that were not necessarily contained in $\langle 1, i, j, k\rangle$. If $\gamma = (x'+y'i+z'j+t'k)/2$ it is easy to see that $\gamma \notin \langle 1, i, j, k\rangle$ as soon as $(x', y', z', t') \neq (0,0,0,0) \mod 2$. Our analysis at the end of Section 3.1 showed that there were 4 possible configurations for $(x', y', z', t') \mod 2$ and each can be obtained when the value of m' is bigger than 1 (which we may assume). The reasoning above justifies that $\#I^1_\tau > 1$ but not that it reaches the desired value of 3. Let us write I_1, I_2 the two other $\mathcal{O}_0$ ideals of norm 2. It can be verified that $I_1 = I_2 i$. Since $(x'+y'i+z'j+t'k)i = -y'+x'i+t'j-z'k$, it is easy to see that if some outputs of $\mathsf{FullRepresentIntegers}$ are contained in I_1, then the same must be true for I_2 (and conversely). This proves that $\#I^1_\tau = 3$, i.e., all three first steps are possible. Yet, there could still be a bias in the distribution of that step, which would still give rise to an attack on Problem 1. We argue in the next section that there is no such exploitable bias.

6.2 Further analysis on the first steps of σ

We continue the analysis by looking at what happens beyond the first 2-isogeny of the elements $\iota \in \mathcal{P}_{N_\tau}$. Henceforth, we will consider the set $\mathcal{P}_{N_\tau}$ associated to a modified version of $\mathsf{SigningKLPT}$. First, we replace $\mathsf{RepresentInteger}$ by $\mathsf{FullRepresentInteger}$ as suggested in Section 6.1. Second, we modify the computation of the exponent e_0. Instead of setting a unique value $e_0(N)$ and then taking $e_0 = e_0(N)$, we propose to take $e_0(N)$ as a range of values from which e_0 will be sampled. The rationale behind this last modification is to cover more γ's (and expand the size of I_k^τ as a result) and it will play a role in the proof of Proposition 6. The proposed range for $e_0(N)$ will be given precisely below.

For any $k \in \mathbb{N}$ smaller than e, let us define $\pi_k : \iota \mapsto \iota_k$ where ι_k is the unique isogeny of degree 2^k such that $\iota = \iota' \circ \iota_k$. We will study the sets $I_\tau^k = \pi_k(\mathcal{P}_{N_\tau})$. We will start by trying to estimate $\# I_\tau^k$ for values of $k \approx 1/2 \log(p)$. Our analysis culminates in Proposition 6, which we prove under several plausible assumptions. Even though it does not prove that Problem 1 is hard, showing that $\# I_\tau^k$ is exponential in the security parameter rules out attacks similar to the one outlined in Section 6.1.

A truly meaningful result would be to show that the distribution $\mathcal{D}_\tau^k$ of the $\pi_k(\iota)$ when ι is uniformly random in $\mathcal{P}_{N_\tau}$ is indistinguishable from the uniform distribution on the isogenies of degree 2^k. In the end of this section, we will try to argue that the D_k^τ are not biased for small values of k. The result we obtain are not very formal but we back them up with experimental results.

The size of $\boldsymbol{I_\tau^k}$. Our goal is to show that I_τ^k contains a good portion of the isogenies of degree 2^k for values of $k \approx p^{1/2}$. Our final result is stated in Proposition 6 and basically follows from the fact that the isogenies of I_τ^k only depend on the quaternion element γ of norm $N\ell^{e_0}$ when $k \leq e_0$ (this fact follows from the analysis underlying Lemma 3). We recall that in the definition of $\mathcal{P}_{N_\tau}$, γ is a possible output of $\mathsf{FullRepresentInteger}$ such that the end of the computation in Algorithm 8 terminates. Thus, one of the main ingredients of our proof is a result (stated as Proposition 5) on the number of γ of norm M that can be obtained as output of $\mathsf{FullRepresentInteger}$. We use the notation Γ_M for the set of primitive $\gamma \in \mathcal{O}_0$ of norm M.

For Proposition 5, we assume that the algorithm $\mathsf{Cornacchia}$ outputs $\perp$ on input M' when M' is not a near-prime (the multiple of a prime by a smooth factor) or if M' is a near-prime but cannot be represented by the quadratic form $x^2 + y^2$. Otherwise, the algorithm outputs any of the possible solutions to the quadratic equation.

Proposition 5. *Let $M > p$. Under plausible heuristics, there exists a constant $c_1 > 0$ such that the number of $\gamma \in \Gamma_M$ that are possible outputs of* $\mathsf{FullRepresentInteger}$ *on input M is larger than $\#\Gamma_M c_1 / \log(M)$.*

Proof. Let $2\gamma = x' + iy' + jz' + kt'$ and $M' = 4M - p(f(z', t'))$. Given our assumption on $\mathsf{Cornacchia}$, γ is going to be an admissible output if and only if

M' is a near-prime and the pair z', t' can be sampled during the first two steps of Algorithm 3. For z', t' it is easy to verify that this is the case. Indeed, the value of $|z'|$ must be smaller than $2m$. Thus, there is a possibility that this value is picked. After that, we know that the correct value of $|t'|$ must be smaller than m' and so there is also a possibility that the correct value is picked. Then, under the assumption that the M' behave as normal integers of the same size, we get that there exists a constant c_1 such that a fraction $c_1/\log(M)$ of all the M' are near-primes. Thus, the same fraction of γ are going to be possible outputs of $\mathsf{FullRepresentInteger}$ and this concludes the proof.

Before proceeding to the last part of the proof, we will need some of the estimates used in [8, Section 6.4]. We give without proof a reformulation of [8, Lemmas 9 and 10] as Lemmas 5 and 6.

Lemma 5. *There exists* $\varepsilon = O(\log\log(p))$ *such that for a random class* $\mathcal{C} \in \mathrm{Cl}(\mathcal{O}_0)$*, the norm* N *of* $\mathsf{EquivalentPrimeIdeal}(\mathcal{C})$ *verifies* $\log(p)/2 - \varepsilon < \log(N) < \log(p)/2 + \varepsilon$ *with overwhelming probability.*

Lemma 6. *For any* $\kappa \in \mathbb{N}$*, there exists* $\eta_0 = O(\log\log(p) + \log(\kappa))$ *such that for any* $e_0 \geq \log(p) - \log(N) + \varepsilon + \eta_0$*, the probability that there exists a solution* $\gamma = \mathsf{FullRepresentInteger}_{\mathcal{O}_0}(N\ell^{e_0})$ *that will lead to a correct execution of* $\mathsf{SigningKLPT}$ *is higher than* $1 - 2^{-\kappa}$.

We recall that we study $\mathcal{P}_{N_\tau}$ for a modified version of $\mathsf{SigningKLPT}$ (the full list of changes is given in the beginning of this section) that samples the exponent e_0 inside a range that we denote by $e_0(N)$. We define $e_0(N) = [\log_2(p) - \log_2(N) + \varepsilon + \eta_0, \log_2(p) - \log_2(N) + \varepsilon + \eta_0 + \delta]$, where ε, η_0 are defined as in Lemmas 5 and 6 (these results tells us that the execution of $\mathsf{SigningKLPT}$ succeeds with overwhelming probability when $e_0 \in e_0(N)$). We also introduce the variable parameter δ upon which the statement of Proposition 6 will depend. If we want that $\mathsf{SigningKLPT}$ terminates with overwhelming probability we need to have $\delta = O(\log\log(p))$ so that $e_1 = e - e_0$ remains in the range prescribed by [8, Lemma 11].

Proposition 6. *Let* δ *be a positive value and* ε, η_0 *be as defined for Lemmas 5 and 6. If* $k \in [\frac{\log_\ell(p)}{2} + \eta_0, \frac{\log_\ell(p)}{2} + 2\varepsilon + \eta_0 + \delta]$*, then under plausible heuristics there exists a constant* $c > 0$ *such that*

$$\# I_\tau^k \geq c \cdot 2 \cdot 3^{k-1}/(\log(p) + \delta).$$

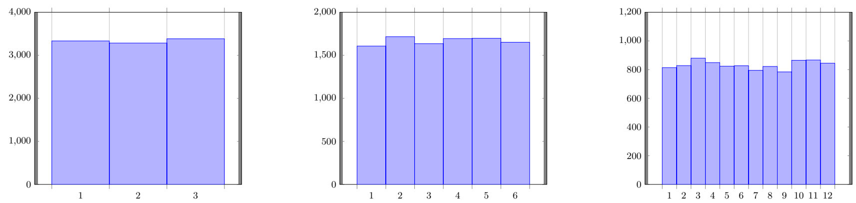
Proof. Let φ be an isogeny of degree 2^k. We write I_φ for the corresponding ideal and $L_\varphi = \mathsf{EquivalentPrimeIdeal}(I_\varphi)$, $N_\varphi = n(L_\varphi)$. There exists a quaternion element γ_φ of norm $N_\varphi 2^k$ such that $\mathcal{O}_0\gamma_\varphi = I_\varphi \cdot \overline{L}_\varphi$. It can be easily verified that $\varphi \in I_\tau^k$ if and only γ_φ is in the set of possible γ involved in the definition of $\mathcal{P}_{N_\tau}$. We write this set Γ_τ. For γ_φ to be in Γ_τ, we need to verify the following things: $k \in e_0(N_\varphi)$, γ_φ is a possible output of $\mathsf{FullRepresentInteger}$ on input $N_\varphi 2^k$ and the rest of the computation of $\mathsf{SigningKLPT}$ (Step 4 to Step 7) must succeed from γ_φ.

Lemmas 5 and 6 and the definition of $e_0(N)$ and k ensures that only a negligible number of isogenies φ would have $k \notin e_0(N_\varphi)$. After that, if we assume that γ_φ is distributed correctly in the $\Gamma_{N_\varphi 2^k}$, Proposition 5 tells us there exists a constant $c_2 > 0$ such that more than a fraction $c_2/(\log(p)+\delta)$ of the γ_φ will be possible outputs of FullRepresentInteger. Finally, we can make the assumption that a constant fraction of those γ_φ will satisfy the last requirement (see the analysis led in [7] to justify this assumption). Thus, we obtain that there exists some constant $c > 0$ such that a fraction bigger than $c/(\log(p)+\delta)$ of all the γ_φ are contained in Γ_τ, and we can conclude the proof.

The exponential size of I_τ^k proven in Proposition 6 is enough to guarantee that a distinguisher cannot be constructed by counting the elements of I_τ^k after a polynomial number of calls to the oracle of Problem 1.

The distribution $\mathcal{D}_\tau^k$ is another matter of importance. Biased distributions, especially for small values of k, can be easily detected which would break Problem 1. Once again, our analysis focuses of the quaternion element γ. Any bias on the distribution $\mathcal{D}_\tau^k$ is a consequence of a bias on the distribution of $\mathcal{O}_0\langle\gamma, \ell^k\rangle$ among the ideals of norm 2^k. If $2\gamma = x + yi + zj + tk$ for $x, y, z, t \in \mathbb{Z}$, it can be shown that $\mathcal{O}_0\langle\gamma, \ell^k\rangle$ will depend on the values of $(x, y, z, t) \mod (2\ell^k)$. It is easy to argue that the values of z, t are sampled without any bias $\mod (2\ell^k)$ when m', m'' are big enough compared to ℓ^k (which we may assume since we look at small values of k). After that, we can only argue informally that the near-primality condition on $M - pf(z,t)$ should not introduce any bias on the value of $z, t \mod 2 \cdot \ell^k$. It also seems plausible that the output of Cornacchia on random near-prime inputs of a given size should not skew the distribution of x, y but we cannot really prove it. Short of proving a positive result, we can at least point out that our formulation of FullRepresentInteger avoids several pitfalls that would have lead to a noticeable bias. The two examples that we give below are focused on the case $\ell = 2$ and $k = 1$. Both can be verified easily with small experiments. When $\ell = 2$, one might be tempted to allow the computation to succeed even when the condition $x' = t' \mod 2$ and $y' = z' \mod 2$ is not met. In the end, if $M = N \cdot 2^{e_0}$, then $x' + y'i + z'j + t'k$ has norm $N \cdot 2^{2+e_0}$ and might be a suitable solution for the rest of the computation. However, this would result in a strong bias in the distribution of isogenies in I_τ^1 as it would favor elements in $\langle 1, i, j, k\rangle$ that are contained inside $\mathcal{O}_0\langle 1+i, 2\rangle$. A skewed distribution is also the explanation behind Remark 3. When swapping x', y' to satisfy $x' = t' \mod 2$ and $y' = z' \mod 2$, one increase the chances that $(x', y', z', t') \mod 2 \in \langle(1,0,0,1), (0,1,1,0)\rangle$ and thus modifies the distribution $\mathcal{D}_\tau^1$.

Experimental evidence. We present below in Fig. 3 the result of an experiment to study the distributions $\mathcal{D}_\tau^k$ for small values of k. The results are consistent with our informal analysis.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Fig. 3: Distribution of the k-first steps of σ for 10 SQISign keys and random ideal in input over 1000 attempts.

7 Conclusion and open problems

We introduced a new algorithm to perform the ideal-to-isogeny translation in the prime power case. We showed that our approach is more efficient than the previous method from [7] by substituting it into the original SQISign implementation. Other schemes might benefit from our new method. For instance, the key generation of the Séta encryption protocol from [6] is based on the same mechanisms. The key generation of their implementation currently takes hours due to the big prime size and it should benefit greatly from our new method. We also made some theoretical and experimental advances on the cryptanalysis of SQISign by exhibiting a distinguisher on one of the security assumptions, and proposing a fix.

A number of questions remain open. The first is about efficiency. In particular, we need finer cost metrics to improve our understanding of our algorithm's behaviour. This is important for both optimization and parameter selection. The second is about further improvements of the ideal-to-isogeny procedure. Our new algorithm simplifies and improves upon the method from [7], yet it is still very slow and the algorithm remains very convoluted. Short of any radically new ideas, one might try to improve what we already have. The impact of improving the quality of the outputs of KLPT has been argued in [7], and the same is true for SpecialEichlerNorm. In general, any improvement in solving norm equations inside the lattices of $B_{p,\infty}$ could have a positive impact on our scheme. Finally, cryptanalysis of SQISign is still in its infancy. We provided some heuristic evidence that our proposed fixes prevent distinguishing attacks, however we are far from a formal proof, even based on heuristics, that distributions of simulated transcripts are statistically close to real ones.

References

1. Sarah Arpin, Mingjie Chen, Kristin E. Lauter, Renate Scheidler, Katherine E. Stange, and Ha T. N. Tran, *Orienteering with one endomorphism*, Cryptology ePrint Archive, Report 2022/098, 2022, https://ia.cr/2022/098.

2. Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith, *Faster computation of isogenies of large prime degree*, ANTS (2020).
3. Giuseppe Cornacchia, *Su di un metodo per la risoluzione in numeri interi dell'equazione* $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$, Giornale di Matematiche di Battaglini **46** (1908), 33–90.
4. Craig Costello, *B-sidh: supersingular isogeny diffie-hellman using twisted torsion*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2020, pp. 440–463.
5. Craig Costello, Michael Meyer, and Michael Naehrig, *Sieving for twin smooth integers with solutions to the prouhet-tarry-escott problem*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2021, pp. 272–301.
6. Luca De Feo, Cyprien Delpech de Saint Guilhem, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, and Benjamin Wesolowski, *Séta: Supersingular encryption from torsion attacks*, Tech. report, Cryptology ePrint Archive, Report 2019/1291, 2019. https://eprint. iacr. org . . . , 2019.
7. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, *Sqisign: compact post-quantum signatures from quaternions and isogenies*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2020, pp. 64–93.
8. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, *SQISign: compact post-quantum signatures from quaternions and isogenies (extended version)*, Cryptology ePrint Archive, Report 2020/1240, January 2021, retrieved 16 Jan 2022.
9. Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso, *Verifiable delay functions from supersingular isogenies and pairings*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2019, pp. 248–277.
10. Max Deuring, *Die typen der multiplikatorenringe elliptischer funktionenkörper*, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **14** (1941), no. 1, 197–272.
11. Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit, *Supersingular isogeny graphs and endomorphism rings: Reductions and solutions*, Advances in Cryptology – EUROCRYPT 2018 (Jesper Buus Nielsen and Vincent Rijmen, eds.), Springer International Publishing, 2018, pp. 329–368.
12. Amos Fiat and Adi Shamir, *How to prove yourself: Practical solutions to identification and signature problems*, Conference on the theory and application of cryptographic techniques, Springer, 1986, pp. 186–194.
13. Steven D. Galbraith, Christophe Petit, and Javier Silva, *Identification protocols and signature schemes based on supersingular isogeny problems*, ASIACRYPT, 2017.
14. D. Kohel, *Endomorphism rings of elliptic curves over finite fields*, Ph.D. thesis, University of California at Berkeley, 1996.
15. David Kohel, Kristin E. Lauter, Christophe Petit, and Jean-Pierre Tignol, *On the quaternion ℓ-isogeny path problem*, IACR Cryptology ePrint Archive **2014** (2014), 505.
16. Antonin Leroux, *A new isogeny representation and applications to cryptography*, Cryptology ePrint Archive (2021).
17. Jonathan Love and Dan Boneh, *Supersingular curves with small non-integer endomorphisms*, ArXiv **abs/1910.03180** (2019).

18. Joseph H. Silverman, *The arithmetic of elliptic curves*, Gradute Texts in Mathematics, vol. 106, Springer-Verlag, 1986.
19. J. Vélu, *Isogénies entre courbes elliptiques*, Comptes rendus de l'Académie des Sciences, Séries A-B **273** (1971), A238–A241.
20. William C. Waterhouse, *Abelian varieties over finite fields*, Annales scientifiques de l'École Normale Supérieure **2** (1969), no. 4, 521–560 (eng).
21. Benjamin Wesolowski, *Orientations and the supersingular endomorphism ring problem*, Cryptology ePrint Archive, Report 2021/1583, 2021, https://ia.cr/2021/1583.
22. Benjamin Wesolowski, *The supersingular isogeny path and endomorphism ring problems are equivalent*, FOCS 2021-62nd Annual IEEE Symposium on Foundations of Computer Science, 2022.