Signature Schemes Based On Supersingular Isogeny Problems

Steven D. Galbraith[1], Christophe Petit[2], and Javier Silva[2]

[1] Mathematics Department, University of Auckland, NZ.
s.galbraith@auckland.ac.nz
[2] Mathematical Institute, Oxford University, Oxford OX2 6GG, UK.
christophe.petit@maths.ox.ac.uk,javier.silva@upf.edu

Abstract. We present the first signature schemes whose security relies on computational assumptions relating to isogeny graphs of supersingular elliptic curves. We give two schemes. The first one is obtained from an interactive identification protocol due to De Feo, Jao and Plût. The second signature scheme uses novel ideas that have not been used in cryptography previously, and is based on a more standard and potentially stronger computational problem.

1 Introduction

A recent research area is cryptosystems whose security is based on the difficulty of finding a path in the isogeny graph of supersingular elliptic curves [7, 9, 12, 18, 19]. Unlike other elliptic curve cryptosystems, the only known quantum algorithm for these problems, due to Biasse-Jao-Sankar [5], has exponential complexity. Hence, additional motivation for the study of these cryptosystems is that they are possibly suitable for post-quantum cryptography.

Currently there is not a full suite of cryptographic functions available based on isogeny assumptions. The work of Charles-Goren-Lauter [7] gave a collision-resistant hash function. Jao-De Feo [18] gave a key exchange protocol, De Feo-Jao-Plût [12] gave a public key encryption scheme and an interactive identification protocol, Jao-Soukharev [19] gave an undeniable signature, and Xi-Tian-Wang [32] gave a designated verifier signature. Among the cryptographic functions not yet available, the most obvious and important omission is digital signatures.

In this paper we present two public key signature schemes whose security relies on computational problems related to finding a path in the isogeny graph of supersingular elliptic curves.

The first scheme is obtained relatively simply from the De Feo-Jao-Plût [12] interactive identification protocol by using the Fiat-Shamir transform to turn it into a non-interactive signature scheme. We also use a variant of the Fiat-Shamir transform due to Unruh to obtain a post-quantum signature scheme. This scheme has the advantage of being simple to describe, at least to a reader who is familiar with the previous work in the subject, and easy to implement. On the other hand, it inherits the disadvantages of [12], in particular it relies on a non-standard isogeny problem using small isogeny degrees, reveals auxiliary points, and uses special primes.

The fastest classical attack on this scheme has heuristic running time of $\tilde{O}(p^{1/4})$ bit operations, and the fastest quantum attack (see Section 5.1 of [12]) has running time of $\tilde{O}(p^{1/6})$. The recent paper by Galbraith-Petit-Shani-Ti [15] shows that revealing auxiliary points may be dangerous in certain contexts. It is therefore highly advisable to build cryptographic schemes on the most general, standard and potentially hardest isogeny problems.

Our second scheme uses completely different ideas and relies on the difficulty of a more standard computational problem, namely the problem of computing the endomorphism ring of a supersingular elliptic curve (equivalently, computing an isogeny between two given elliptic curves). This computational problem has heuristic classical complexity of $\tilde{O}(p^{1/2})$ bit operations, and quantum complexity $\tilde{O}(p^{1/4})$. The scheme is based on a sigma protocol that is very similar to the proof of graph isomorphism. One obtains a signature scheme by applying the Fiat-Shamir transform or Unruh's transform. We now briefly sketch the main ideas behind our second scheme. The public key is a pair of elliptic curves (E_0, E_1) and the private key is an isogeny $\phi : E_0 \to E_1$. To interactively prove knowledge of ϕ one chooses a random isogeny $\psi : E_1 \to E_2$ and sends E_2 to the verifier. The verifier sends a bit b. If $b = 0$ the prover reveals ψ. If $b = 1$ the prover reveals an isogeny $\mu : E_0 \to E_2$. In either case, the verifier checks that the response is correct. The interaction is repeated a number of times until the verifier is convinced that the prover knows an isogeny from E_0 to E_1. However, the subtlety is that we cannot just set $\mu = \psi \circ \phi$, as then E_1 would appear on the path in the graph from E_0 to E_2 and so we would have leaked the private key. The crucial idea is to use the algorithm of Kohel-Lauter-Petit-Tignol [23] to produce an isogeny $\mu : E_0 \to E_2$ that is completely independent of ϕ. The mathematics behind the algorithm of Kohel-Lauter-Petit-Tignol goes beyond what usually arises in elliptic curve cryptography.

The paper is organized as follows. In Section 2 we give preliminaries on isogeny problems, random walks in isogeny graphs, security definitions and the Fiat-Shamir transform. Sections 3 and 4 describe our two signature schemes and Section 5 concludes the paper. In a first reading to get the intuition of our schemes without all implementation details, one can safely skip parts of the paper, namely Sections 2.2, 2.3, 2.4, 2.6, 4.3 and 4.4.

2 Preliminaries

2.1 Hard Problem Candidates Related to Isogenies

Let E, E' be two elliptic curves over a finite field $\mathbb{F}_q$. An *isogeny* $\varphi : E \to E'$ is a non-constant morphism from E to E' that maps the neutral element into the neutral element. The degree of an isogeny φ is the degree of φ as a morphism. An isogeny of degree ℓ is called an ℓ-isogeny. If φ is separable, then $\deg \varphi = \# \ker \varphi$. If there is a separable isogeny between two curves, we say that they are *isogenous*. Tate's theorem is that two curves E, E' over $\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.

An isogeny can be identified with its kernel [31]. Given a subgroup G of E, we can use Vélu's formulae [30] to explicitly obtain an isogeny $\varphi : E \to E'$ with kernel G and such that $E' \cong E/G$. These formulas involve sums over points in G, so using them is efficient as long as $\#G$ is small. Kohel [22] and Dewaghe [11] have (independently) given formulae for the Vélu isogeny in terms of the coefficients of the polynomial defining the kernel, rather than in terms of the points in the kernel. Given a prime ℓ, the torsion group $E[\ell]$ contains exactly $\ell + 1$ cyclic subgroups of order ℓ, each one corresponding to a different isogeny.

A composition of n separable isogenies of degrees ℓ_i for $1 \leq i \leq n$ gives an isogeny of degree $N = \prod_i \ell_i$ with kernel a group G of order N. Conversely any isogeny whose kernel is a group of smooth order can be composed as a sequence of isogenies of small degree, hence can be computed efficiently. For any permutation σ on $\{1, \ldots, n\}$, by considering appropriate subgroups of G, one can write the isogeny as a composition of isogenies of degree $\ell_{\sigma(i)}$. Hence, there is no loss of generality in the protocols in our paper of considering chains of isogenies of increasing degree.

For each isogeny $\varphi : E \to E'$, there is a unique isogeny $\hat{\varphi} : E' \to E$, which is called the *dual isogeny* of φ, and which satisfies $\varphi\hat{\varphi} = \hat{\varphi}\varphi = [\deg \varphi]$. If we have two isogenies $\varphi : E \to E'$ and $\varphi' : E' \to E$ such that $\varphi\varphi'$ and $\varphi'\varphi$ are the identity in their respective curves, we say that φ, φ' are *isomorphisms*, and that E, E' are *isomorphic*. Isomorphism classes of elliptic curves over $\mathbb{F}_q$ can be labeled with their j-invariant [27, III.1.4(b)]. An isogeny $\varphi : E \to E'$ such that $E = E'$ is called an *endomorphism*. The set of endomorphisms of an elliptic curve, denoted by $\mathrm{End}(E)$, has a ring structure with the operations point-wise addition and function composition.

Elliptic curves can be classified according to their endomorphism ring. Over the algebraic closure of the field, $\mathrm{End}(E)$ is either an order in a quadratic imaginary field or a maximal order in a quaternion algebra. In the first case, we say that the curve is *ordinary*, whereas in the second case we say that the curve is *supersingular*. Indeed, the endomorphism ring of a supersingular curve over a field of characteristic p is a maximal order $\mathcal{O}$ in the quaternion algebra $B_{p,\infty}$ ramified at p and ∞.

In the case of supersingular elliptic curves, there is always a curve in the isomorphism class defined over $\mathbb{F}_{p^2}$, and the j-invariant of the class is also an element of $\mathbb{F}_{p^2}$. A theorem by Deuring [10] gives an equivalence of categories between the j-invariants of supersingular elliptic curves over $\mathbb{F}_{p^2}$ up to Galois conjugacy in $\mathbb{F}_{p^2}$, and the maximal orders in the quaternion algebra $B_{p,\infty}$ up to the equivalence relation given by $\mathcal{O} \sim \mathcal{O}'$ if and only if $\mathcal{O} = \alpha^{-1}\mathcal{O}'\alpha$ for some $\alpha \in B_{p,\infty}^*$. Specifically, the equivalence of categories associates to every j-invariant a maximal order that is isomorphic to the endomorphism ring of any curve with that j-invariant. Furthermore, if E_0 is an elliptic curve with $\mathrm{End}(E_0) = \mathcal{O}_0$, there is a one-to-one correspondence (which we call the *Deuring correspondence*) between isogenies $\psi : E_0 \to E$ and left $\mathcal{O}_0$-modules I.

We now present some hard problem candidates related to supersingular elliptic curves, and discuss the related algebraic problems in the light of the Deuring correspondence.

Problem 1 *Let p, ℓ be distinct prime numbers. Let E, E' be two supersingular elliptic curves over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = \#E'(\mathbb{F}_{p^2}) = (p+1)^2$, chosen uniformly at random. Find $k \in \mathbb{N}$ and an isogeny of degree ℓ^k from E to E'.*

The fastest classical algorithm known for this problem has heuristic running time of $\tilde{O}(p^{1/2})$ bit operations.

Problem 2 *Let p, ℓ be distinct prime numbers. Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, chosen uniformly at random. Find $k_1, k_2 \in \mathbb{N}$, a supersingular elliptic curve E' over $\mathbb{F}_{p^2}$, and two distinct isogenies of degrees ℓ^{k_1} and ℓ^{k_2}, respectively, from E to E'.*

The hardness assumption of the second problem has been used in [7] to prove collision-resistance of a proposed hash function. Slightly different versions of the first problem, in which some extra information is provided, were used in [12] to build an identification scheme, a key exchange protocol and a public-key encryption scheme.

More precisely, the identification protocol of De Feo-Jao-Plût [12] relies on problems 3 and 4 below (which De Feo, Jao and Plût call the *Computational Supersingular Isogeny (CSSI)* and *Decisional Supersingular Product (DSSP)* problems). In order to state them we need to introduce some notation. Let p be a prime of the form $\ell_1^{e_1}\ell_2^{e_2} \cdot f \pm 1$, and let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Let $\{R_1, S_1\}$ and $\{R_2, S_2\}$ be bases for $E_0[\ell_1^{e_1}]$ and $E_0[\ell_2^{e_2}]$, respectively.

Problem 3 (Computational Supersingular Isogeny - CSSI) *Let $\phi_1 : E_0 \to E_1$ be an isogeny with kernel $\langle [m_1]R_1 + [n_1]S_1 \rangle$, where m_1, n_1 are chosen uniformly at random from $\mathbb{Z}/\ell_1^{e_1}\mathbb{Z}$, and not both divisible by ℓ_1. Given E_1 and the values $\phi_1(R_2), \phi_1(S_2)$, find a generator of $\langle [m_1]R_1 + [n_1]S_1 \rangle$.*

The fastest known classical algorithm for this problem has heuristic running time of $\tilde{O}(\ell_1^{e_1/2})$ bit operations, which is $\tilde{O}(p^{1/4})$ in the context of De Feo-Jao-Plût [12].

Problem 4 (Decisional Supersingular Product - DSSP) *Let E_0, E_1 be supersingular elliptic curves over $\mathbb{F}_{p^2}$ such that there exists an isogeny $\phi : E_0 \to E_1$ of degree $\ell_1^{e_1}$. Fix generators $R_2, S_2 \in E_0[\ell_2^{e_2}]$ and suppose $\phi(R_2)$ and $\phi(S_2)$ are given. Consider the two distributions of pairs (E_2, E_3) as follows:*

- *(E_2, E_3) such that there is a cyclic group $G \subseteq E_0[\ell_2^{e_2}]$ of order $\ell_2^{e_2}$ and $E_2 \cong E_0/G$ and $E_3 \cong E_1/\phi(G)$.*
- *(E_2, E_3) where E_2 is chosen at random among the curves having the same cardinality as E_0, and $\phi' : E_2 \to E_3$ is a random $\ell_1^{e_1}$-isogeny.*

The problem is, given (E_0, E_1) and the auxiliary points $(R_2, S_2, \phi(R_2), \phi(S_2))$, plus a pair (E_2, E_3), to determine from which distribution the pair is sampled.

We stress that Problems 3 and 4 are potentially weaker than Problems 1 and 2 because special primes are used, extra points are revealed, and particularly small degree isogenies exist between the curves. The following problem, on the other hand, offers better foundations for cryptography based on supersingular isogeny problems.

Problem 5 *Let p be a prime number. Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, chosen uniformly at random. Determine the endomorphism ring of E.*

Note that it is essential that the curve is chosen randomly in this problem, as for special curves the endomorphism ring is easy to compute. Essentially, Problem 5 is the same as explicitly computing the forward direction of Deuring's correspondence. This problem was studied in [22], in which an algorithm to solve it was obtained, but with expected running time $\tilde{O}(p)$. It was later improved by Galbraith to $\tilde{O}(p^{\frac{1}{2}})$, under heuristic assumptions [14]. Interestingly, the best *quantum* algorithm for this problem runs in time $\tilde{O}(p^{\frac{1}{4}})$, only providing a quadratic speedup over classical algorithms. This has largely motivated the use of supersingular isogeny problems in cryptography.

Problem 6 *Let p be a prime number. Let E, E' be supersingular elliptic curves over $\mathbb{F}_{p^2}$, chosen uniformly at random. Find an isogeny $E \to E'$.*

Heuristically, if we can solve Problem 1 or Problem 6, then we can solve Problem 5. If we can compute isogenies, we can fix $E_0 = E_1$ to obtain endomorphisms, and in this case it is easy to find four endomorphisms that are linearly independent, thus generating a subring of $\mathrm{End}(E_0)$, and this subring is likely to be of small index so that the full ring can be recovered.

For the converse, suppose that we can compute the endomorphism rings of both E_0 and E_1. The strategy is to compute a module I that is a left ideal of $\mathrm{End}(E_0)$ and a right ideal of $\mathrm{End}(E_1)$ of appropriate norm, and to translate it back to the geometric setting to obtain an isogeny. This approach motivated the quaternion ℓ-isogeny algorithm of Kohel-Lauter-Petit-Tignol [23, 26], which solves the following problem:

Problem 7 *Let p, ℓ be distinct prime numbers. Let $\mathcal{O}_0, \mathcal{O}_1$ be two maximal orders in $B_{p,\infty}$, chosen uniformly at random. Find $k \in \mathbb{N}$ and an ideal I of norm ℓ^k such that I is a left $\mathcal{O}_0$-ideal and its right order is isomorphic to $\mathcal{O}_1$.*

The algorithm can be adapted to produce ideals of B-powersmooth norm (meaning the norm is $\prod_i \ell_i^{e_i}$ where the ℓ_i are distinct primes and $\ell_i^{e_i} \leq B$) for $B \approx \frac{7}{2}\log p$ and using $O(\log p)$ different primes, instead of ideals of norm a power of ℓ. We will use that version in our second signature scheme.

For completeness we mention that ordinary curve versions of Problems 1 and 5 are not known to be equivalent, and in fact there is a subexponential algorithm for computing the endomorphism ring of ordinary curves [6], whereas the best clasical algorithm known for computing isogenies is still exponential. There is, however, a subexponential quantum algorithm for computing an isogeny between ordinary curves [8], which is why the main interest in cryptography is the supersingular case.

2.2 Random Walks in Isogeny Graphs

Let $p \geq 5$ be a prime number. There are $N_p := \frac{p}{12} + \epsilon_p$ supersingular j-invariants in characteristic p, with $\epsilon_p = 0, 1, 1, 2$ when $p = 1, 5, 7, 11 \bmod 12$ respectively. For any prime $\ell \neq p$, one can construct a so-called isogeny graph, where each vertex is associated to a supersingular j-invariant, and an edge between two vertices is associated to a degree ℓ isogeny between the corresponding vertices.

Isogeny graphs are regular[3] with regularity degree $\ell + 1$; they are undirected since to any isogeny from j_1 to j_2 corresponds a dual isogeny from j_2 to j_1. Isogeny graphs

[3] One needs to pay close attention to the cases $j = 0$ and $j = 1728$ when counting isogenies, but this has no effect on our general schemes.

are also very good *expander graphs* [17]; in fact they are optimal expander graphs in the following sense:

Definition 1 (Ramanujan graph) *Let G be a k-regular graph, and let $1, \lambda_2, \cdots, \lambda_r$ be the eigenvalues of the normalized adjacency matrix sorted by decreasing order of the absolute value. Then G is a* Ramanujan graph *if*

$$\lambda_2 \leq \frac{2\sqrt{k-1}}{k}.$$

This is optimal by the Alon-Boppana bound: given a family $\{G_N\}$ of k-regular graphs as above, and denoting by $\lambda_{2,N}$ the corresponding second eigenvalue of each graph G_N, we have $\liminf_{N\to\infty} \lambda_{2,N} \geq \frac{2\sqrt{k-1}}{k}$. The Ramanujan property of isogeny graphs follows from the Weil conjectures (proved by Deligne).

Let p and ℓ be as above, and let j be a supersingular invariant in characteristic p. We define a random step of degree ℓ from j as the process of randomly and uniformly choosing a neighbour of j in the ℓ-isogeny graph, and returning that vertex. For a composite degree $n = \prod_i \ell_i$, we define a random walk of degree n from j_0 as a sequence of j-invariants j_i such that j_i is a random step of degree ℓ_i from j_{i-1}. We do not require the primes ℓ_i to be distinct.

The output of random walks in expander graphs converge quickly to a uniform distribution. In our second signature scheme we will be using random walks of B-*powersmooth* degree n, namely $n = \prod_i \ell_i^{e_i}$, with all prime powers $\ell_i^{e_i}$ smaller than some bound B, with B as small as possible. To analyze the ouptut distribution of these walks we will use the following generalization[4] of classical random walk theorems [17]:

Theorem 1 (Random walk theorem). *Let p be a prime number, and let j_0 be a supersingular invariant in characteristic p. Let j be the final j-invariant reached by a random walk of degree $n = \prod_i \ell_i^{e_i}$ from j_0. Then for every j-invariant $\tilde{j}$ we have*

$$\left|\Pr[j = \tilde{j}] - \frac{1}{N_p}\right| \leq \prod_i \left(\frac{2\sqrt{\ell_i}}{\ell_i + 1}\right)^{e_i}.$$

PROOF: Let v_{kj} be the probability that the outcome of the first k random steps is a given vertex j, and let $v_k = (v_{kj})_j$ be vectors encoding these probabilities. Let A_{ℓ_i} be the normalized adjacency matrix of the ℓ_i-isogeny graph. Clearly A_{ℓ_i} is a stochastic matrix, so its largest eigenvalue is 1. By the Ramanujan property the second largest eigenvalue is smaller than 1 in absolute value, so the eigenspace associated to $\lambda_1 = 1$ is of dimension 1 and generated by the vector $u := (N_p^{-1})_j$ corresponding to the uniform distribution. Let λ_{2i} be the second largest eigenvalue of A_{ℓ_i} in absolute value.

If step k is of degree ℓ_i we have $v_k = A_{\ell_i} v_{k-1}$. Moreover we have $||v_k - u||_2 \leq \lambda_{2i}||v_{k-1} - u||_2$ since the eigenspace associated to 1 is of dimension 1. Iterating on all steps we deduce

$$||v_k - u||_2 \leq \prod_i \lambda_{2i}^{e_i}||v_0 - u||_2 \leq \prod_i |\lambda_{2i}|^{e_i}$$

[4] Random walks theorems are usually stated for a single graph whereas our walks will switch from one graph to another, all with the same vertex set but different edges.

since $||v_0 - u||_2^2 = (1 - \frac{1}{N_p})^2 + \frac{N_p - 1}{N_p}(\frac{1}{N_p})^2 \leq 1 - \frac{2}{N} + \frac{2}{N^2} < 1$. Finally we have

$$\left|\Pr[j = \tilde{j}] - \frac{1}{N_p}\right| = ||v_k - u||_\infty \leq ||v_k - u||_2 \leq \prod_i |\lambda_{2i}|^{e_i} \leq \prod_i \left(\frac{2\sqrt{\ell_i}}{\ell_i + 1}\right)^{e_i},$$

where we have used the Ramanujan property to bound the eigenvalues. □

In our security proof we will want the right-hand term to be smaller than $\frac{1}{2N_p^2} \approx \frac{72}{p^2}$, and at the same time we will want the powersmooth bound B to be as small as possible. The following lemma shows that taking $B \approx 4 \log p$ suffices asymptotically.

Lemma 1 *There is a function $c_p = c(p)$ such that $\lim_{p \to \infty} c_p = 4$, and, for each p,*

$$\prod_{\substack{\ell_i \text{ prime} \\ e_i := \max\{e | \ell_i^e < c_p \log p\}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right)^{e_i} > \frac{p^2}{72}.$$

We refer to Appendix A for the proof of this lemma. In our later schemes, for security parameter λ, we will choose p such that $\log p = 2\lambda$. For concrete values, in particular for $\lambda = 128$ and $\lambda = 256$, it is easy to verify that the inequality holds if one takes the product of all prime powers up to $B = c \log p$, for the values $c = 3.71$ and $c = 3.56$, respectively.

2.3 Efficient Representations of Isogeny Paths and Other Data

Our schemes require representing/transmitting elliptic curves and isogenies. In this section we first explain how to represent certain mathematical objects appearing in our protocol as bitstrings in a canonical way so that minimal data needs to be sent and stored. Next, we discuss different representations of isogeny paths and their impact on the efficiency of our signature schemes. As these paths will be sent from one party to another, the second party needs an efficient way to verify that the bitstring received corresponds to an isogeny path between the right curves.

Let p be a prime number. Every supersingular j-invariant is defined over $\mathbb{F}_{p^2}$. A canonical representation of $\mathbb{F}_{p^2}$-elements is obtained via a canonical choice of degree 2 irreducible polynomial over $\mathbb{F}_p$. Canonical representations in any other extension fields are defined in a similar way. Although there are only about $p/12$ supersingular j-invariants in characteristic p, we are not aware of an efficient method to encode these invariants into $\log p$ bits, so we represent supersingular j-invariants with the $2 \log p$ bits it takes to represent an arbitrary $\mathbb{F}_{p^2}$-element.

Elliptic curves are defined by their j-invariant up to isomorphism. Hence, rather than sending the coefficients of the elliptic curve equation, it suffices to send the j-invariant. For any invariant j there is a canonical elliptic curve equation $E_j : y^2 = x^3 + \frac{3j}{1728-j}x + \frac{2j}{1728-j}$ when $j \neq 0, 1728$, $y^2 = x^3 + 1$ when $j = 0$, and $y^2 = x^3 + x$ when $j = 1728$. The last one will be of particular interest in our second signature scheme.

We now turn to representing chains $E_0, E_1, \ldots, E_n$ of isogenies $\phi_i : E_{i-1} \to E_i$ each of prime degree ℓ_i where $1 \leq i \leq n$. Here ℓ_i are always very small primes. A useful feature of our protocols is that isogeny chains can always be chosen such that the isogeny degrees are increasing $\ell_i \geq \ell_{i-1}$. First we need to discuss how to represent the sequence of isogeny degrees. If all degrees are equal to a constant ℓ (e.g., $\ell = 2$) then there is nothing to send. If the degrees are different then the most compact representation seems to be to compute and send

$$N = \prod_{i=1}^{n} \ell_i.$$

The receiver can recover the sequence of isogeny degrees from N by factoring using trial division and ordering the primes by size. This representation is possible due to our convention the isogeny degrees are increasing and since the degrees are all small.

Now we discuss how to represent the curves themselves in the chain of isogenies. We give several methods.

1. There are two naive representations. One is to send all the j-invariants $j_i = j(E_i)$ for $0 \leq i \leq n$. This requires $2(n+1)\log_2(p)$ bits.
   Note that the verifier is able to check the correctness of the isogeny chain by checking that $\Phi_{\ell_i}(j_{i-1}, j_i) = 0$ for all $1 \leq i \leq n$, where Φ_{ℓ_i} is the ℓ_i-th modular polynomial. The advantage of this method is that verification is relatively quick (just evaluating a polynomial that can be precomputed and stored).
   The other naive method is to send the x-coordinate of a kernel point $P_i \in E_{j_i}$ on the canonical curve. Given j_{i-1} and the kernel point P_{i-1} one computes the isogeny $\phi_i : E_{j_{i-1}} \to E_{j_i}$ using the Vélu formula and hence deduces j_i. Note that the kernel point is not unique (indeed, in some rare cases there can be more than one subgroup that corresponds to an isogeny $E_{j_{i-1}} \to E_{j_i}$).
   Both these methods require huge bandwidth.
   A refinement of the second method is used in our first signature scheme, where ℓ is fixed and one can publish a point that defines the kernel of the entire isogeny chain. Precisely a curve E and points $R, S \in E[\ell^n]$ are fixed. Each integer $0 \leq \alpha < \ell^n$ defines a subgroup $\langle R + [\alpha]S \rangle$ and hence an ℓ^n isogeny. It suffices to send α, which requires $\log_2(\ell^n)$ bits. In the case $\ell = 2$ this is just n bits, which is smaller than all the other suggestions in this section.
2. One can improve upon the naive method in several simple ways. One method is to send every second j-invariant. The Verifier accepts this as a valid path if, for all odd integers i, the greatest common divisor over $\mathbb{F}_{p^2}[y]$

   $$\gcd(\Phi_{\ell_i}(j_{i-1}, y), \Phi_{\ell_{i+1}}(y, j_{i+1}))$$

   is a linear polynomial $(y - \alpha)$ for some α (which is therefore j_i).
   Another method is to send only some least significant bits (more than $\log_2(\ell_i + 1)$ of them) of the j_i instead of the entire value. The verifier reconstructs the isogeny path by factoring $\Phi_{\ell_i}(j_{i-1}, y)$ over $\mathbb{F}_{p^2}$ (it will always split completely in the supersingular case) and then selecting j_i to be the root that has the correct least significant bits.

3. An optimal compression method seems to be to define a well-ordering on $\mathbb{F}_{p^2}$ (e.g., lexicographic order on the binary representation of the element). Instead of j_i one sends the index k such that when the ℓ_i+1 roots of $\Phi_{\ell_i}(j_{i-1}, y)$ are written in order, j_i is the k-th root. It is clear that the verifier can reconstruct the value j_i and hence can reconstruct the whole chain from this information. The sequence of integers k can be encoded as a single integer in terms of a "base $\prod_{j=1}^{i}(\ell_i+1)$" representation. If the walk is non-backtracking and the primes ℓ_i are repeated then one can remove the factor $(y - j_{i-2})$ that corresponds to the dual isogeny of the previous step, this can save some bandwidth.
   We call this method "optimal" since it is hard to imagine doing better than $\log_2(\ell_i+1)$ bits for each step in general.[5] Though we have no proof that one cannot do better. However, note that the verifier now needs to perform polynomial factorisation, which may cause some overhead in a protocol. Note that in the case where all $\ell_i = 2$ and the walk is non-backtracking then this method also requires n bits, which matches the method we use in our first signature scheme (mentioned in item 1 above).
4. A variant of the optimal method is to use an ordering on points/subgroups rather than j-invariants. At each step one sends an index k such that the isogeny $\phi : E_{i-1} \to E_i$ is defined by the k-th cyclic subgroup of $E_{j_{i-1}}[\ell_i]$. Again the verifier can reconstruct the path, but this requires factoring ℓ_i-division polynomials.
   To be precise: Given a canonical ordering on the field of definition of $E[\ell]$, one can define a canonical ordering of the cyclic kernels, hence represent them by a single integer in $\{0, \dots, \ell\}$. One can extend this canonical ordering to kernels of composite degrees in various simple ways (see also [3, Section 3.2]). If two curves are connected by two distinct isogenies of the same degree then either one can be chosen (it makes no difference in our protocols), so the ambiguity in exceptional cases is never a problem for us.
   In practice, since these points may be defined over an extension of $\mathbb{F}_{p^2}$, we believe that ordering the roots of $\Phi_{\ell_i}(j_{i-1}, y)$ is significantly more efficient than ordering kernel subgroups.

When $p = 3 \bmod 4$, the quaternion algebra $B_{p,\infty}$ ramified at p and ∞ can be canonically represented as $\mathbb{Q}\langle \mathbf{i}, \mathbf{j} \rangle$, where $\mathbf{i}^2 = -1$, $\mathbf{j}^2 = -p$ and $\mathbf{k} := \mathbf{ij} = -\mathbf{ji}$. The maximal order O_0 with $\mathbb{Z}$-basis $\{1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2}\}$ corresponds to the curve E_0 with j-invariant $j_0 := 1728$ under Deuring's correspondence, and there is an isomorphism of quaternion algebras $\theta : B_{p,\infty} \to \mathrm{End}(E_0) \otimes \mathbb{Q}$ sending $(1, \mathbf{i}, \mathbf{j}, \mathbf{k})$ to $(1, \phi, \pi, \pi\phi)$ where $\pi : (x, y) \to (x^p, y^p)$ is the Frobenius endomorphism, and $\phi : (x, y) \to (-x, \iota y)$ with $\iota^2 = -1 \bmod p$.

We now give a brief analysis of the complexity of the operations, assuming fast (quasi-linear) modular and polynomial arithmetic.

As discussed above, an isogeny step of prime degree ℓ can be described by a single integer in $\{0, \dots, \ell\}$. Similarly, by combining integers in a product, an isogeny of degree $\prod_i \ell_i^{e_i}$ can be described by a single positive integer smaller than $\prod_i (\ell_i + 1)^{e_i}$.

[5] In the most general case, when all primes ℓ_i are distinct, then there are $\prod_i(\ell_i + 1)$ possible isogeny paths and thus one cannot expect to represent an arbitrary path using fewer than $\log_2(\prod_i \ell_i)$ bits.

This integer can define either a list of subgroups (specified in terms of some ordering), or a list of supersingular j-invariants (specified in terms of an ordering on the roots of the modular polynomial). In the first case, the verifier will need at each step given a j-invariant to compute the curve equation, then its full ℓ_i torsion (which may be over a large field extension), then to sort with respect to some canonical ordering the cyclic subgroups of order ℓ_i to identify the correct one, and finally to compute the next j-invariant with Vélu's formulae [30]. In the second case the verifier will need at each step given a j-invariant, to specialize one variable of the ℓ_i-th modular polynomial, then to compute all roots of the resulting univariate polynomial and finally to sort the roots to identify the correct one. The second method is more efficient as it does not require running Vélu's formulae over some large field extension, and the root-finding and sorting routines are applied on smaller inputs. We assume that the modular polynomials are precomputed.

In our second signature scheme we will have $\ell_i^{e_i} = O(\log p)$. The cost of computing an isogeny increases with the size of ℓ_i. Hence it suffices to analyse the larger case, for which $e_i = 1$ and $\ell_i = O(\log p)$. Assuming precomputation of the modular polynomials and using [29] for polynomial factorization, the most expensive part of an isogeny step is evaluating the modular polynomials $\Phi_{\ell_i}(x, y)$ at $x = j_{i-1}$: as these polynomials are bivariate with degree ℓ_i in each variable they have $O(\ell_i^2)$ monomials and so this requires $O(\log^2 p)$ field operations for a total cost of $\tilde{O}(\log^3 p)$ bit operations since j-invariants are defined over $\mathbb{F}_{p^2}$. In our first signature scheme based on the De Feo-Jao-Plût protocol we have $\ell_i = O(1)$ so each isogeny step costs $\tilde{O}(\log p)$ bit operations.

Alternatively, isogeny paths can be given as a sequence of j-invariants. To verify the path is correct one still must compute $\Phi_{\ell_i}(j_{i-1}, j_i)$, which still requires $\tilde{O}(\log p)$ bit operations. However, in practice it would be much quicker to not require root-finding algorithms. Also, all the steps can be checked in parallel, and all the steps of a same degree are checked using the same polynomial, so we expect many implementation optimizations to be possible.

2.4 Identification Schemes and Security Definitions

An identification scheme is an interactive protocol between two parties (a Prover and a Verifier). We use the terminology and notation of Abdalla-An-Bellare-Namprempre [1] (also see Bellare-Poettering-Stebila [4]). We also introduce a notion of "recoverability" which is implicit in the Schnorr signature scheme and seems to be folklore in the field.

Definition 1. *A* canonical identification scheme *is $\mathcal{ID} = (K, \mathcal{P}, \mathcal{V}, c)$ where K is a randomised algorithm (key generation) that on input a security parameter λ outputs a pair (PK, SK); $\mathcal{P}$ is an algorithm taking input* SK*, random coins r and state information* ST *and returns a message, c is the length of the challenge (a function of the parameter k) and $\mathcal{V}$ is a deterministic verification algorithm that takes as input* PK *and a transcript and outputs* 0 *or* 1*. A* transcript of an honest execution *of the scheme $\mathcal{ID}$ is the sequence:* $\text{CMT} \leftarrow \mathcal{P}(\text{SK}, r)$, $\text{CH} \leftarrow \{0,1\}^c$, $\text{RSP} \leftarrow \mathcal{P}(\text{SK}, r, \text{CMT}\|\text{CH})$. *On an honest execution we require that* $\mathcal{V}(\text{PK}, \text{CMT}\|\text{CH}\|\text{RSP}) = 1$.

An impersonator *for $\mathcal{ID}$ is an algorithm I that plays the following game: I takes as input a public key* $\textsc{pk}$ *and a set of transcripts of honest executions of the scheme $\mathcal{ID}$; I outputs* $\textsc{cmt}$*, receives* $\textsc{ch} \leftarrow \{0,1\}^c$ *and outputs* $\textsc{rsp}$*. We say that I wins if* $\mathcal{V}(\textsc{pk}, \textsc{cmt}\|\textsc{ch}\|\textsc{rsp}) = 1$*. The advantage of I is* $|\Pr(I \text{ wins}) - \frac{1}{2^c}|$*. We say that $\mathcal{ID}$ is* secure against impersonation under passive attacks *if the advantage is negligible for all proababilistic polynomial-time adversaries.*

An ID-scheme $\mathcal{ID}$ is non-trivial *if $c \geq \lambda$.*

An ID-scheme is recoverable *if there is a deterministic algorithm Rec such that for any transcript* $\textsc{cmt}\|\textsc{ch}\|\textsc{rsp}$ *of an honest execution we have* $Rec(\textsc{pk}, \textsc{ch}, \textsc{rsp}) = \textsc{cmt}$*.*

One can also formulate an ID-scheme as a special case of a sigma-protocol with respect to the relation given by $(\textsc{pk}, \textsc{sk}) \leftarrow K$, where we think of $\textsc{sk}$ as a witness. In the language of sigma protocols the following properties are standard:

- **Correctness (or completeness):** if the prover knows the secret, then the verifier will accept.
- **Soundness:** if an adversary does not know the secret $\textsc{sk}$, he should not be able to convince a verifier.
- **Honest Verifier Zero-knowledge:** A transcript of an honest execution does not reveal any information about the secret.

To formalise soundness, we use the notion of *n-special soundness*, which essentially says that given a fixed commitment and n valid answers to n different challenges, there exists an algorithm to recover the witness efficiently. This captures the idea that if the prover had not known the secret, he would not have been able to produce valid answers. Honest verifier zero-knowledge is proved by showing that one can simulate transcripts of honest executions without knowledge of $\textsc{sk}$.

For future reference we give precise definitions for two of these properties.

Definition 2. *Let $\mathcal{ID} = (K, \mathcal{P}, \mathcal{V}, c)$ be an identification scheme, viewed as a sigma protocol. Let R be the relation* $(\textsc{pk}, \textsc{sk})$ *given by the outputs of the key generation function K.*

- *$\mathcal{ID}$ is* n-special sound *if there exists a probabilistic polynomial time algorithm $\mathcal{X}$ such that for all probabilistic polynomial time adversaries $\mathcal{A}$, we have*

$$\left| \Pr \begin{bmatrix} \textsc{pk} \leftarrow K(1^\lambda); (\textsc{pk}, \textsc{cmt}, \{\textsc{ch}_i\}_{i=1}^n, \{\textsc{rsp}_i\}_{i=1}^n) \leftarrow \mathcal{A}(\textsc{pk}); \\ \mathcal{V}(\textsc{pk}, \textsc{cmt}\|\textsc{ch}_i\|\textsc{rsp}_i) = 1 \ \forall i \in \{1, \ldots, n\}; \\ \textsc{sk} \leftarrow \mathcal{X}(\textsc{pk}, \textsc{cmt}, \{\textsc{ch}_i\}_{i=1}^n, \{\textsc{rsp}_i\}_{i=1}^n) : (\textsc{pk}, \textsc{sk}) \in R \end{bmatrix} - 1 \right| \leq \operatorname{negl}(\lambda).$$

- *$\mathcal{ID}$ is* honest verifier zero-knowledge *if there exists a probabilistic polynomial time simulator $\mathcal{S}$ such that for all probabilistic polynomial time adversaries $\mathcal{A}$, we have*

$$\begin{aligned} &|\Pr[(\textsc{pk}, \textsc{sk}) \leftarrow K(1^\lambda), r \leftarrow \$; \textsc{ch} \leftarrow \mathcal{A}(\textsc{pk}); \textsc{cmt} \leftarrow \mathcal{P}(\textsc{sk}, r); \\ &\qquad \textsc{rsp} \leftarrow \mathcal{P}(\textsc{sk}, r, \textsc{cmt}\|\textsc{ch}) : \mathcal{A}(\textsc{cmt}, \textsc{rsp}) = 1] \\ &- \Pr[\textsc{pk} \leftarrow K(1^\lambda); \textsc{ch} \leftarrow \mathcal{A}(\textsc{pk}); (\textsc{cmt}, \textsc{rsp}) \leftarrow \mathcal{S}(\textsc{pk}, \textsc{ch}) : \mathcal{A}(\textsc{cmt}, \textsc{rsp}) = 1]| \leq \operatorname{negl}(\lambda). \end{aligned}$$

One can transform any ID scheme into a non-trivial scheme by running t sessions in parallel: One first generates $\text{CMT}_i \leftarrow \mathcal{P}(\text{SK}, r_i)$ for $1 \leq i \leq t$. One then samples $\text{CH} \leftarrow \{0,1\}^{ct}$ and parses it as $\text{CH}_i \in \{0,1\}^c$ for $1 \leq i \leq t$. Finally one computes $\text{RSP}_i \leftarrow P(\text{SK}, r_i, \text{CMT}_i \| \text{CH}_i)$. We define

$$\mathcal{V}(\text{PK}, \text{CMT}_1 \| \cdots \text{CMT}_t \| \text{CH} \| \text{RSP}_1 \| \cdots \| \text{RSP}_t) = 1$$

if and only if $\mathcal{V}(\text{PK}, \text{CMT}_i \| \text{CH}_i \| \text{RSP}_i) = 1$ for all $1 \leq i \leq t$. The successful cheating probability is then improved to $1/2^{ct}$, which is non-trivial when $t \geq \lambda/c$.

2.5 Signatures and the Fiat-Shamir Transform

For signature schemes we use the standard definition of *existential unforgeability under chosen message attacks* [20] (we sometimes abbreviate this to *secure*). An adversary can ask for polynomially many signatures of messages of his choice to a signing oracle $\mathsf{Sign}_{\text{SK}}(\cdot)$. Then, the attack is considered successful if the attacker is able to produce a valid pair of message and signature for a message different from those queried to the oracle.

We now discuss the Fiat-Shamir transform [13] to build a signature scheme from an identification scheme. The idea is to make the interactive protocol $\mathcal{ID} = (K, \mathcal{P}, \mathcal{V}, c)$ non-interactive by using a random oracle to produce the challenges. Suppose the protocol $\mathcal{ID}$ must be executed in parallel t times to be non-trivial (with soundness probability $1/2^{tc}$). Let H be a random oracle that outputs a bit string of length ct.

- $(\text{PK}, \text{SK}) \leftarrow K(\lambda)$: this is the same as in the identification protocol. The public key and secret key are the public key and the secret key from key generation algorithm K of the identification protocol.
- $\mathsf{Sign}(\text{SK}, m)$: Compute the commitments $\text{CMT}_i \leftarrow \mathcal{P}(\text{SK}, r_i)$ for $1 \leq i \leq t$. Compute $h = H(m, \text{CMT}_1 \| \cdots \| \text{CMT}_t)$. Parse h as the t values $\text{CH}_i \in \{0,1\}^c$. Compute $\text{RSP}_i \leftarrow \mathcal{P}(\text{SK}, r_i, \text{CMT}_i \| \text{CH}_i)$ for $1 \leq i \leq t$. Output the signature $\sigma = (\text{CMT}_1, \ldots, \text{CMT}_t, \text{RSP}_1, \ldots, \text{RSP}_t)$.
- $\mathsf{Verify}(m, \sigma, \text{PK})$: compute $h = H(m, \text{CMT}_1 \| \cdots \| \text{CMT}_t)$. Parse h as the t values $\text{CH}_i \in \{0,1\}^c$. Check that $\mathcal{V}(\text{PK}, \text{CMT}_i \| \text{CH}_i \| \text{RSP}_i) = 1$ for all $1 \leq i \leq t$. If $\mathcal{V}$ returns 1 for all i then output 1, else output 0.

Abdalla-An-Bellare-Namprempre [1] (also see Bellare-Poettering-Stebila [4]) have proved the security of the Fiat-Shamir transform to a high degree of generality.

Theorem 2. *Let $\mathcal{ID}$ be a non-trivial canonical identification protocol that is secure against impersonation under passive attacks. Let $\mathcal{S}$ be the signature scheme derived from $\mathcal{ID}$ using the Fiat-Shamir transform. Then $\mathcal{S}$ is secure against chosen-message attacks in the random oracle model.*

Remark 1. If the ID-scheme $\mathcal{ID}$ is recoverable then one can obtain a more compact signature scheme. Recall that "recoverable" means there is a deterministic algorithm Rec such that for any transcript of an honest execution we have $\text{Rec}(\text{PK}, \text{CH}, \text{RSP}) = \text{CMT}$. We now describe the signature scheme.

– $(\text{PK}, \text{SK}) \leftarrow K(\lambda)$.
– $\mathsf{Sign}(\text{SK}, m)$: Compute the commitments $\text{CMT}_i \leftarrow \mathcal{P}(\text{SK}, r_i)$ for $1 \leq i \leq t$. Compute $h = H(m, \text{CMT}_1 \| \cdots \| \text{CMT}_t)$. Parse h as the t values $\text{CH}_i \in \{0,1\}^c$. Compute $\text{RSP}_i \leftarrow \mathcal{P}(\text{SK}, r_i, \text{CMT}_i \| \text{CH}_i)$ for $1 \leq i \leq t$. Output the signature $\sigma = (h, \text{RSP}_1, \ldots, \text{RSP}_t)$.
– $\mathsf{Verify}(m, \sigma, \text{PK})$: Parse h as the t values $\text{CH}_i \in \{0,1\}^c$. Compute $\text{CMT}_i = \text{Rec}(\text{PK}, \text{CH}_i, \text{RSP}_i)$ for $1 \leq i \leq t$. Check that $h = H(m, \text{CMT}_1 \| \cdots \| \text{CMT}_t)$ and that $\mathcal{V}(\text{PK}, \text{CMT}_i \| \text{CH}_i \| \text{RSP}_i) = 1$ for all $1 \leq i \leq t$. If V returns 1 for all i then output 1, else output 0.

An attacker against this signature scheme can be turned into an attacker on the original signature scheme (and vice versa), which shows that both schemes have the same security. This is addressed in the following result.

Theorem 3. *Let $\mathcal{ID}$ be a non-trivial canonical recoverable identification protocol that is secure against impersonation under passive attacks. Let $\mathcal{S}$ be the signature scheme derived from $\mathcal{ID}$ using the Fiat-Shamir transform of Remark 1. Then $\mathcal{S}$ is secure against chosen-message attacks in the random oracle model.*

PROOF: Let A be an algorithm that forges signatures against the signature scheme of Remark 1. We will convert A into an algorithm B that forges signatures for the original Fiat-Shamir signature scheme that is proved secure in Theorem 2.

Let B be given as input a public key PK, and call A on that key. When A makes a sign query or a hash query, pass these on as queries made by B. Results of hash queries are forwarded to A. When B gets back a signature $(\text{CMT}_1, \ldots, \text{CMT}_t, \text{RSP}_1, \ldots, \text{RSP}_t)$ for message m we compute $h = H(m, \text{CMT}_1 \| \ldots \| \text{CMT}_t,)$ and return to A the signature $\sigma = (h, \text{RSP}_1, \ldots, \text{RSP}_t)$.

Finally A outputs a forgery $\sigma^* = (h^*, \text{RSP}_1^*, \ldots, \text{RSP}_t^*)$ on message m. This is different from previous outputs of the sign oracle, which means that $\sigma \neq (h, \text{RSP}_1, \ldots, \text{RSP}_t)$ for every output of the sign oracle. Note that this non-equality means either $\text{RSP}_i^* \neq \text{RSP}_i$ for some i or $h \neq h^*$. Compute $\text{CMT}_i^* = \text{Rec}(\text{PK}, \text{CH}_i^*, \text{RSP}_i^*)$ for $1 \leq i \leq t$ and return $(\text{CMT}_1^*, \ldots, \text{CMT}_t^*, \text{RSP}_1^*, \ldots, \text{RSP}_t^*)$ as a forgery on message m for the original scheme. We claim that this is also distinct from all other signatures that have been returned to B: if equal to some previous signature $(\text{CMT}_1, \ldots, \text{CMT}_t, \text{RSP}_1, \ldots, \text{RSP}_t)$ on message m then $\text{RSP}_i^* = \text{RSP}_i$ and $h^* = H(m, \text{CMT}_1^* \| \ldots \| \text{CMT}_t^*) = h$, which violates the fact that σ^* was a valid forgery on m. □

2.6 Post-Quantum Fiat Shamir Transformations

We now discuss variants of the Fiat-Shamir transform that provide full security against quantum adversaries. This is an active area of research and more efficient transforms may yet be discovered.

Unruh [28] has given a transform that converts a secure interactive identification scheme into a signature scheme that is secure against a quantum adversary. His transform is also discussed by Goldfeder, Chase and Zaverucha [16]. The basic transform (Figure 1 of [28]) takes a identification scheme $\mathcal{ID} = (K, P, V, c)$ with keys (PK, SK)

given as a sigma protocol (to be executed in parallel t times) with transcripts $\textsc{cmt} \leftarrow P(\textsc{sk}, r)$, $\textsc{ch} \leftarrow \{0,1\}^c$, $\textsc{rsp} \leftarrow P(\textsc{sk}, r, \textsc{cmt} \| \textsc{ch})$. The additional requirement is a random permutation G and a hash function H that outputs tc bits. The signature scheme has public key $\textsc{pk}$ and private key $\textsc{sk}$. The signer computes the commitments $\textsc{cmt}_i \leftarrow P(\textsc{sk}, r_i)$ for $1 \le i \le t$. Now, for every possible challenge $\textsc{ch}_j \in \{0,1\}^c$ compute all responses $\textsc{rsp}_{i,j} \leftarrow P(\textsc{sk}, r_i, \textsc{cmt}_i \| \textsc{ch}_j)$ and $g_{i,j} = G(\textsc{rsp}_{i,j})$ (note that this is $t2^c$ values). Let $h = H(m, \textsc{cmt}_1, \ldots, \textsc{cmt}_t, g_{1,1}, \ldots, g_{t,2^c})$ and parse as $\textsc{ch}_1, \ldots, \textsc{ch}_t \in \{0,1\}^c$. The signature is

$$\sigma = (\textsc{cmt}_1, \ldots, \textsc{cmt}_t, \textsc{rsp}_{1,\textsc{ch}_1}, \ldots, \textsc{rsp}_{t,\textsc{ch}_t}, g_{1,1}, \ldots, g_{t,2^c}).$$

The verification algorithm is obvious.

Theorems 13 and 18 of [28] prove that if $\mathcal{ID}$ is honest-verifier zero knowledge and has n-special soundness then the resulting signature scheme is existentially unforgeable under a chosen-message attack.

3 First Signature Scheme

This section presents a signature scheme obtained from the interactive identification protocol of De Feo-Jao-Plût [12]. First we describe their scheme.

3.1 De Feo-Jao-Plût Identification Scheme

Let p be a large prime of the form $\ell_1^{e_1}\ell_2^{e_2} \cdot f \pm 1$, where ℓ_1, ℓ_2 are small primes (typically $\ell_1 = 2$ and $\ell_2 = 3$). We start with a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ with $\#E_0(\mathbb{F}_{p^2}) = \ell_1^{e_1}\ell_2^{e_2} \cdot f$ and a primitive $\ell_1^{e_1}$-torsion point P_1. Define $E_1 = E_0/\langle P_1 \rangle$ and denote the corresponding $\ell_1^{e_1}$-isogeny by $\phi : E_0 \to E_1$.

Let R_2, S_2 be a pair of generators of $E_0[\ell_2^{e_2}]$. The public key is $(E_0, E_1, R_2, S_2, \phi(R_2), \phi(S_2))$. The private key is the point P_1. The interaction goes as follows:

1. The prover chooses a random primitive $\ell_2^{e_2}$-torsion point P_2 as $P_2 = aR_2 + bS_2$ for some integers $0 \le a, b < \ell_2^{e_2}$. Note that $\phi(P_2) = a\phi(R_2) + b\phi(S_2)$. The prover defines the curves $E_2 = E_0/\langle P_2 \rangle$ and $E_3 = E_1/\langle \phi(P_2) \rangle = E_0/\langle P_1, P_2 \rangle$, and uses Vélu's formulae to compute the following diagram.

$$\begin{tikzcd} E_0 \arrow[r, "\phi"] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\ E_2 \arrow[r, "\phi'"] & E_3 \end{tikzcd}$$

The prover sends E_2 and E_3 to the verifier.

2. The verifier challenges the prover with a random bit $\text{CH} \leftarrow \{0, 1\}$.
3. If $\text{CH} = 0$, the prover reveals P_2 and $\phi(P_2)$.
   If $\text{CH} = 1$, the prover reveals $\psi(P_1)$.

In both cases, the verifier accepts the proof if the points revealed have the right order and are the kernels of isogenies between the right curves. We iterate this process to reduce the cheating probability.

The following theorem is the main security result for this section. The basic ideas of the proof are by De Feo-Jao-Plût [12], but we give a slightly different formalisation that is required for our signature proof.

Theorem 4. *If Problems 3 and 4 are computationally hard, then the interactive protocol defined above, repeated t times in parallel for a suitable parameter t, is a non-trivial canonical identification protocol that is secure against impersonation under passive attacks.*

PROOF: It is straightforward to check that the scheme is correct (in other words, the sigma protocol is complete). We now show that parallel executions of the signma protocol are sound and honest verifier zero knowledge.

For soundness: Suppose $\mathcal{A}$ is an adversary that takes as input the public key and succeeds in the identification protocol with noticeable probability ϵ. Given a challenge instance $(E_0, E_1, R_1, S_1, R_2, S_2, \phi(R_2), \phi(S_2))$ for Problem 3 we run $\mathcal{A}$ on this tuple as the public key. In the first round, $\mathcal{A}$ outputs commitments $(E_{i,2}, E_{i,3})$ for $1 \le i \le t$. We then send a challenge $\text{CH} \in \{0, 1\}^t$ to $\mathcal{A}$ and, with probability ϵ outputs a response RSP that satisfies the verification algorithm. Now, we use the standard replay technique: Rewind $\mathcal{A}$ to the point where it had output its commitments and then respond with a different challenge $\text{CH}' \in \{0, 1\}^t$. With probability ϵ, $\mathcal{A}$ outputs a valid response RSP'.

Now, choose some index i such that $\text{CH}_i \neq \text{CH}'_i$. We now restrict our focus to the components CMT_i, RSP_i and RSP'_i. It means $\mathcal{A}$ sent E_2, E_3 and can answer both challenges $\text{CH} = 0$ and $\text{CH} = 1$ successfully. Hence we have the following diagram.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\phi"] \arrow[r, bend right, "\tilde{\phi}"'] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\
E_2 \arrow[r, "\phi'"] & E_3
\end{tikzcd}$$

From this, one has an explicit description of an isogeny $\tilde{\phi} = \hat{\psi}' \circ \phi' \circ \psi$ from E_0 to E_1. The degree of $\tilde{\phi}$ is $\ell_1^{e_1} \ell_2^{2e_2}$. One can determine $\ker(\tilde{\phi}) \cap E_0[\ell_1^{e_1}]$ by iteratively testing points in $E_0[\ell_1^j]$ for $j = 1, 2, \ldots$. Hence, one determines the kernel of ϕ, as desired. This proves soundness.

Now we show honest verifier zero-knowledge. For this it suffices to show that one can simulate transcripts of the protocol without knowing the private key. When $b = 0$

we simulate correctly by choosing $u, v \in \mathbb{Z}_{\ell_2^{e_2}}$ and setting $E_2 = E_0/\langle uR_2 + vS_2\rangle$ and $E_3 = E_1/\langle u\phi(R_2) + v\phi(S_2)\rangle$. When $b = 1$ we choose a random curve E_2 and a random point $R \in E_2[\ell_1^{e_1}]$ and we publish $E_2, E_3 = E_2/\langle R\rangle$ and answer with the point R (hence defining the isogeny). Although (E_2, E_3) are a priori not distributed correctly, the computational assumption of Problem 4 implies it is computationally hard to distinguish the simulation from the real game. Hence the scheme has computational zero knowledge.

Finally we prove the identification scheme is secure against impersonation under passive attacks. Let I be an impersonator for the scheme. Given a challenge instance $(E_0, E_1, R_1, S_1, R_2, S_2, \phi(R_2), \phi(S_2))$ for Problem 3 we run I on this tuple as the public key. We are required to provide I with a set of transcripts of honest executions of the scheme, but this is done using the simulation method used to show the sigma protocol has honest verifier zero knowledge. If I is able to succeed in its impersonation game then it breaks the soundness of the sigma protocol. We have already shown that if an adversary can break soundness then we can solve Problem 3. This completes the proof. $\square$

3.2 Signature Scheme based on De Feo-Jao-Plût Identification Protocol

One can apply the Fiat-Shamir transform from Section 2.5 to the De Feo-Jao-Plût identification scheme to obtain a signature scheme. One can also check that the scheme is recoverable and so one can apply the Fiat-Shamir variant from Remark 1. In this section we fully specify the signature scheme resulting from the transform of Remark 1, together with some optimisations.

Our main focus is to minimise signature size. Hence, we use the most space-efficient variant of the Fiat-Shamir transform. Next we need to consider how to minimise the amount of data that needs to be sent to specify the isogenies. Several approaches were considered in Section 2.3. For the pair of vertical isogenies it seems to be most compact to represent them using a representation of the kernel (this is more efficient than specifying two paths in the isogeny graph), however this requires additional points in the public key. For the horizontal isogeny there are several possible approaches, but we think the most compact is to use the representation in terms of specifying roots of the modular polynomial. One can easily find other implementations that allow different tradeoffs of public key size versus signature size.

Key Generation Algorithm: On input a security parameter λ generate a prime p with at least 4λ bits, such that $p = \ell_1^{e_1}\ell_2^{e_2}f \pm 1$, with ℓ_1, ℓ_2, f small (ideally $f = 1, \ell_1 = 2, \ell_2 = 3$) and $\ell_1^{e_1} \approx \ell_2^{e_2}$. Choose[6] a supersingular elliptic curve E_0 with j-invariant j_0. Fix points $R_2, S_2 \in E_0(\mathbb{F}_{p^2})[\ell_2^{e_2}]$ and a random primitive $\ell_1^{e_1}$-torsion point $P_1 \in E_0[\ell_1^{e_1}]$. Compute the isogeny $\phi : E_0 \to E_1$ with kernel generated by P_1, and let j_1 be the j-invariant of the image curve. Set $R_2' = \phi(R_2)$, $S_2' = \phi(S_2)$. Choose a hash function H

[6] Costello-Longa-Naehrig [9] choose a special j-invariant in $\mathbb{F}_p$ for efficiency reasons in their implementation of the supersingular key exchange protocol. One could also choose a random j-invariant by performing a random isogeny walk from any fixed j-invariant.

with $t = t(\lambda)$ bits of output (depending on the security requirements we may choose $t = \lambda$ or $t = 2\lambda$). The secret key is P_1, and the public key is $(p, j_0, j_1, R_2, S_2, R'_2, S'_2, H)$. One can reduce the size of the public key by using different representations of isogeny paths, but for simplicity we use this variant.

Signature Algorithm: For $i = 1, \ldots, t$, choose random integers $0 \leq \alpha_i < \ell_2^{e_2}$. Compute the isogeny $\psi_i : E_0 \to E_{2,i}$ with kernel generated by $R_2 + [\alpha_i]S_2$ and let $j_{2,i} = j(E_{2,i})$. Compute the isogeny $\psi'_i : E_1 \to E_{3,i}$ with kernel generated by $R'_2 + [\alpha_i]S'_2$ and let $j_{3,i} = j(E_{3,i})$. Compute $h = H(m, j_{2,1}, \ldots, j_{2,t}, j_{3,1}, \ldots, j_{3,t})$ and parse the output as t challenge bits b_i. For $i = 1, \ldots, t$, if $b_i = 0$ then set $z_i = \alpha_i$. If $b_i = 1$ then compute $\psi_i(P_1)$ and compute a representation z_i of the j-invariant $j_{2,i} \in \mathbb{F}_{p^2}$ and the isogeny with kernel generated by $\psi_i(P_1)$ (for example, as a sequence of integers representing which roots of the ℓ_1-division polynomial to choose at each step of a non-backtracking walk, or using a compact representation of $\psi_i(P_1)$ in reference to a canonical basis of $E_{2,i}[\ell_1^{e_1}]$). Return the signature $\sigma = (h, z_1, \ldots, z_t)$.

Verification Algorithm: On input a message m, a signature σ and a public key PK, recover the parameters p, E_0, E_1. For each $1 \leq i \leq t$, using the information provided by z_i, one recomputes the j-invariants $j_{2,i}, j_{3,i}$. In the case $b_i = 0$ this is done using $z_i = \alpha_i$ by computing the isogeny from E_0 with kernel generated by $R_2 + [\alpha_i]S_2$ and the isogeny from E_1 with generated by $R'_2 + [\alpha_i]S'_2$. When $b_i = 1$ then the value $j_{2,i}$ is provided as part of z_i, together with a description of the isogeny from $E_{2,i}$ to $E_{3,i}$.

One then computes

$$h' = H(m, j_{2,1}, \ldots, j_{2,t}, j_{3,1}, \ldots, j_{3,t})$$

and checks that the value equals h from the signature. The signature is accepted if this is true and is rejected otherwise.

Theorem 5. *If Problems 3 and 4 are computationally hard then the first signature scheme is secure in the random oracle model under a chosen message attack.*

PROOF: This follows immediately from Theorem 3 and Theorem 4. $\square$

One can also apply the Unruh transform described in Section 2.6 to obtain a signature scheme that is proven secure against quantum adversaries.

Theorem 6. *If Problems 3 and 4 are computationally hard for a quantum computer then the signature scheme obtained from the Unruh transform on the sigma protocol of Section 3.1 is a secure signature scheme against quantum adversaries in the random oracle model.*

Efficiency As isogenies are of degree roughly $\sqrt{p}$, the scheme requires to use primes p of size 4λ to defeat meet-in-the-middle attacks. Assuming H is some fixed hash function and therefore not sent, the secret key is simply $x(P_1) \in \mathbb{F}_{p^2}$ and so requires $2 \log p = 8\lambda$ bits.

The public key is p and then $j_0, j_1, x(R_2), x(S_2), x(R'_2), x(S'_2) \in \mathbb{F}_{p^2}$ which requires $13 \log_2(p) \approx 52\lambda$ bits. The signature size is analysed in Lemma 2.

In terms of computational complexity. The basic operations are repeated $O(\lambda)$ times (one for each challenge bit) and each operation requires computing isogenies that are a composition of around $O(\lambda)$ isogenies of degree ℓ_1 or ℓ_2, each of which is a small number of field operations. Assuming quasi-linear cost $\tilde{O}(\log(p^2)) = \tilde{O}(\lambda)$ for the field operations, the computational complexity of the signing and verifying algorithms is $\tilde{O}(\lambda^3)$ bit operations.

Remark 2. The question of the output length t of the hash function depends on the security requirements. For non-repudiation it is necessary that H be collision-resistant, and so one takes $t = 2\lambda$. But if one is only concerned with security against forgery then one can take $t = \lambda$. This is similar to the case of Schnorr signatures, as mentioned by Schnorr and discussed in detail by Neven-Smart-Warinschi [24]. In both settings, the choice of hash function should be made carefully.

Lemma 2 *The average signature size of this scheme is*

$$t \ + \ \tfrac{t}{2}\lceil \log_2(\ell_2^{e_2})\rceil + \tfrac{t}{2}\left(2\lceil \log_2(p)\rceil + \lceil \log_2(\ell_1^{e_1})\rceil\right)$$

bits. The minimum signature size for λ bits of security is approximately $6\lambda^2$ bits. If non-repudiation is required then the minimum signature size is approximately $12\lambda^2$ bits.

PROOF: On average half the bits b_i of the hash value are zero and half are one. When $b_i = 0$ we send an integer α_i such that $0 \leq \alpha_i < \ell_2^{e_2}$, which requires $\lceil \log_2(\ell_2^{e_2})\rceil$ bits. When $b_i = 1$ we need to send $j_{2,i} \in \mathbb{F}_{p^2}$, which requires $2\lceil \log_2(p)\rceil$ bits, followed by a representation of the isogeny. One can represent a generator of the kernel of the isogeny with respect to some canonical generators P_1', Q_1' of $E_{2,i}[\ell_1^{e_1}]$ as β_i such that $0 \leq \beta_i < \ell_1^{e_1}$, thus requiring $\lceil \log_2(\ell_1^{e_1})\rceil$ bits. Alternatively one can represent the non-backtracking sequence of j-invariants in terms of an ordering on the roots of the ℓ_1-th modular polynomial. This also can be done in $\lceil \log_2(\ell_1^{e_1})\rceil$ bits.

For security level λ one can take $t = \lambda$, $\ell_1^{e_1} \approx \ell_2^{e_2} \approx 2^{2\lambda}$ and so $p \approx 2^{4\lambda}$. Hence the signature size is, at best, approximately $(\lambda/2)(2\lambda + 8\lambda + 2\lambda) = 6\lambda^2$ bits. □

4 Second Signature Scheme

We now present our main result. The main advantage of this scheme compared with the one in the previous section is that its security is based on the general problem of computing an isogeny between two supersingular curves, or equivalently on computing the endomorphism ring of a supersingular elliptic curve. Unlike the scheme in the previous section, the prime has no special property and no auxiliary points are revealed.

4.1 Identification Scheme Based on Endomorphism Ring Computation

The concept is similar to the graph isomorphism zero-knowledge protocol, in which we reveal one of two graph isomorphisms, but never enough information to deduce the secret isomorphism.

As recalled in Section 2.3, although it is believed that computing endomorphism rings of supersingular elliptic curves is a hard computational problem in general, there are some particular curves for which it is easy. Therefore let E_0 be a curve for which computing the endomorphism ring is easy. Take a random isogeny (walk in the graph) $\varphi : E_0 \to E_1$ and, using this knowldege, compute $\mathrm{End}(E_1)$. The public information is (E_0, E_1) and the secret is $\mathrm{End}(E_1)$. Under the assumption that computing the endomorphism ring is hard, the secret key is secure.

Our scheme will require three algorithms, that are explained in detail in later sections.

Translate isogeny path to ideal: Given $E_0, O_0 = \mathrm{End}(E_0)$ and a chain of isogenies from E_0 to E_1, to compute $O_1 = \mathrm{End}(E_1)$ and a left-O_0-ideal I whose right order is O_1.

Find new path: Given an ideal I corresponding to an isogeny $E_0 \to E_2$, to produce a new ideal J corresponding to a "random" isogeny $E_0 \to E_2$ of powersmooth degree.

Translate ideal to isogeny path: Given E_0, O_0, E_2, I to compute a sequence of prime degree isogenies giving the path from E_0 to E_2.

Figure 1 gives the interaction between the prover and the verifier. Let L be the product of prime powers ℓ^e up to $B = 4\log(p)$. In other words, let $\ell_1, \ldots, \ell_r$ be the list of all primes up to B and let $L = \prod_{i=1}^{r} \ell_i^{e_i}$ where $\ell_i^{e_i} \leq B < \ell_i^{e_i+1}$. One can see that this is a canonical, recoverable identification protocol, but it is not non-trivial.

---

1. The prover performs a random walk of degree L in the graph, obtaining a curve E_2 and an isogeny $\psi : E_1 \to E_2$, and reveals E_2 to the verifier.
2. The verifier challenges the prover with a random bit $b \leftarrow \{0, 1\}$.
3. If $b = 0$, the prover answers with ψ.
   If $b = 1$, the prover does the following:
   – Compute $\mathrm{End}(E_2)$ and translate the isogeny path between E_0 and E_2 into a corresponding ideal I giving the path in the quaternion algebra.
   – Use the powersmooth version of the quaternion ℓ-isogeny algorithm to compute another path between $\mathrm{End}(E_0)$ and $\mathrm{End}(E_2)$ in the quaternion algebra, which is independent of E_1, represented by an ideal J.
   – Translate the ideal J to an isogeny path η from E_0 to E_2.
4. The verifier accepts the proof if the answer to the challenge is indeed an isogeny between E_1 and E_2 or between E_0 and E_2, respectively.

Fig. 1. New Identification Scheme

The isogenies involved in this protocol are summarized in the following diagram:

$$
\begin{tikzcd}
E_0 \arrow[r, "\varphi"] \arrow[dr, "\eta"'] & E_1 \arrow[d, "\psi"] \\
& E_2
\end{tikzcd}
$$

The two translation algorithms mentioned above in the $b = 1$ case will be described in Section 4.4. They rely on the fact that $\mathrm{End}(E_0)$, $\mathrm{End}(E_1)$ and $\mathrm{End}(E_2)$ are known. The algorithms are efficient when the degree of the random walk is powersmooth. For this reason all isogenies in our protocols will be of powersmooth degree. The powersmooth version of the quaternion isogeny algorithm of Kohel-Lauter-Petit-Tignol will be described and analyzed in Section 4.3. The random walks are taken of sufficiently large degree such that their output has close to uniform distribution, by Theorem 1 and Lemma 1.

We repeat the process to reduce the cheating probability. The computational hardness of Problem 5 remains essentially the same if the curves are chosen according to a distribution that is close to uniform. We can then prove:

Theorem 7. *Let λ be a security parameter and $t \geq \lambda$. If Problem 6 is computationally hard, then the identification scheme obtained from t parallel executions of the protocol in Figure 1 is a non-trivial, canonical, recoverable identification scheme that is secure against impersonation under passive attacks.*

The advantage of this protocol over the protocol proposed in the previous section is that it relies on a more standard and potentially harder computational problem.

In the rest of this section we first give a proof of Theorem 7, then we provide details of the algorithms involved in our scheme, and finally in Section 4.5 we describe the resulting signature scheme.

4.2 Proof of Theorem 7

We shall prove that the protocol in Figure 1 is complete, 2-special sound and zero-knowledge.

Completeness. Let φ be an isogeny between E_0 and E_1 of B-powersmooth degree, for $B = O(\log p)$. If the challenge received is $b = 0$, it is clear that the prover knows a valid isogeny $\psi : E_1 \to E_2$, so the verifier accepts the proof. If $b = 1$, the prover follows the procedure describe above and the verifier accepts. In the next subsections we will show that this procedure is polynomial time.

2-special soundness. Let $\mathcal{A}$ be a forger against the identification scheme that plays the 2-soundness game specified in Definition 2. We describe an extractor algorithm that takes two curves (E_0, E_1) and computes a path between them. Run $\mathcal{A}$ on input (E_0, E_1). Then $\mathcal{A}$ outputs $(\textsc{cmt}, \{\textsc{ch}_1, \textsc{ch}_2\}, \{\textsc{rsp}_1, \textsc{rsp}_2\})$ such that $\mathcal{V}(\textsc{pk}, \textsc{cmt}\|\textsc{ch}_i\|\textsc{rsp}_i) =$

1 for all $i \in \{1, \ldots, n\}$. Since $\mathrm{CH}_1 \neq \mathrm{CH}_2$ there is a location j such that $\mathrm{CH}_{1,j} \neq \mathrm{CH}_{2,j}$. The corresponding components of the responses RSP_1 and RSP_2 therefore give two isogenies $\psi : E_1 \to E_2$, $\eta : E_0 \to E_2$. Given these two valid answers an extraction algorithm can compute an isogeny $\phi : E_0 \to E_1$ as $\phi = \hat{\psi} \circ \eta$, where $\hat{\psi}$ is the dual isogeny of ψ. The extractor outputs ϕ, which satisfies the relation $(E_0, E_1, \phi : E_0 \to E_1)$ corresponding to solutions of Problem 6. This is summarized in the following diagram.

$$\begin{tikzcd}
E_0 \arrow[r, dashed] \arrow[dr, "\eta"'] & E_1 \arrow[d, "\psi"] \\
& E_2
\end{tikzcd}$$

Honest-verifier zero-knowledge. We shall prove that there exists a probabilistic polynomial time simulator $\mathcal{S}$ that outputs transcripts indistinguishable from transcripts of interactions with an honest verifier, in the sense that the two distributions are statistically close. The simulator starts by taking a random coin $b \leftarrow \{0, 1\}$.

– If $b = 0$, take a random walk from E_1 of length L, obtaining a curve E_2 and an isogeny $\psi : E_1 \to E_2$. The simulator outputs the transcript $(E_2, 0, \psi)$.

$$\begin{tikzcd}
E_0 \arrow[r, dashed] & E_1 \arrow[d, "\psi"] \\
& E_2
\end{tikzcd}$$

In this case, it is clear that the distributions of every element in the transcript are the same as in the real interaction, as they are generated in the same way. This is possible because, when $b = 0$, the secret is not required for the prover to answer the challenge.

– If $b = 1$, take a random walk from E_0 of length L to obtain a curve E_2 and an isogeny $\mu : E_0 \to E_2$, then proceed as in Step 3 of Figure 1 to produce another isogeny $\eta : E_0 \to E_2$. The simulator outputs the transcript $(E_2, 1, \eta)$.

$$
\begin{tikzcd}
E_0 \arrow[r, dashed] \arrow[dr, "\mu"] \arrow[dr, bend right, "\eta"'] & E_1 \\
& E_2
\end{tikzcd}
$$

The reason to output η instead of μ is to ensure that the transcript distributions are indistinguishable from the distributions in the real scheme.

We first study the distribution of E_2. Let X_r be the output of the random walk from E_1 to produce E_2 in the real interaction, and let X_s be the output of the random walk from E_0 to produce E_2 in the simulation.

By Theorem 1, we have, for any curve E_i in the graph (here $N_p \approx p/12$ is the number of vertices in the graph)

$$\left|\Pr(X_r = E_i) - \frac{1}{N_p}\right| \leq \frac{1}{2N_p^2}, \qquad \left|\Pr(X_s = E_i) - \frac{1}{N_p}\right| \leq \frac{1}{2N_p^2}.$$

Therefore

$$\sum_i |\Pr(X_r = E_i) - \Pr(X_s = E_i)| \leq N_p \cdot \max_i |\Pr(X_r = E_i) - \Pr(X_s = E_i)| \leq$$
$$\leq N_p \cdot \left(\frac{1}{2N_p^2} + \frac{1}{2N_p^2}\right) = \frac{1}{N_p} = \mathrm{negl}(\log p).$$

In other words, the statistical distance, between the distribution of E_2 in the real signing algorithm and the simulation, is negligible. Now, since η is produced in the same way from E_0 and E_2 as in the real scheme, we have that the statistical distance between the distributions of η is also negligible. This follows from Lemma 3 in Section 4.3, which states that the output of the quaternion path algorithm does not depend on the input ideal, only on its ideal class.

4.3 Quaternion Isogeny Path Algorithm

In this section we sketch the quaternion isogeny algorithm from Kohel-Lauter-Petit-Tignol [23] and we evaluate its complexity when $p = 3 \bmod 4$. (In the original paper the algorithm is only claimed to run in heuristic probabilistic polynomial time.)

The algorithm takes as input two maximal orders O, O' in the quaternion algebra $B_{p,\infty}$, and it returns a sequence of left O-ideals $I_0 = O \subset I_1 \subset \ldots \subset I_e$ such that the right order of I_e is in the same equivalence class as O'. In addition, the output is such that the index of I_{i+1} in I_i is a small prime for all i. The authors focus on the case where the norm of I_e is ℓ^e for some integer e, but they mention that the algorithm can be extended to the case of powersmooth norms. We will only describe and use the powersmooth version. In our application there are some efficiency advantages from using isogenies whose degree is a product of small powers of distinct primes, rather than a large power of a small prime.

Note that the ideals returned by the quaternion isogeny path algorithm (or equivalently the right orders of these ideals) correspond to vertices of the path in the quaternion algebra graph, and to a sequence of j-invariants by Deuring's correspondence. In the next subsection we will describe how to make this correspondence explicit; here we focus on the quaternion algorithm itself.

An important feature of the algorithm is that paths between two arbitrary maximal orders O and O' are always constructed as a concatenation of two paths from each maximal order to a special maximal order, which in our protocol we take equal to $O_0 = \langle 1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2} \rangle$.

We focus on the case wrhere $O = O_0$, and assume that instead of a second maximal O' we are given the corresponding left O_0-ideal I as input. This will be sufficient for our use of the algorithm. We assume that I is given by a $\mathbb{Z}$ basis of elements in O_0. Denote by $n(\alpha)$ and $n(I)$ the reduced norm of an element or ideal respectively. The equivalence class of maximal orders defines an equivalence class of O_0-ideals, where two ideals I and J are in the same class if and only if $I = Jq$ with $q \in B_{p,\infty}^*$. Therefore our goal is, given a left O_0 ideal I, to compute another left O_0 ideal J with powersmooth norm. Without loss of generality we assume there is no integer $s > 1$ such that $I \subset sO_0$, and that $I \neq O_0$. The algorithm proceeds as follows:

1. Compute an element $\delta \in I$ and an ideal $I' = I\bar{\delta}/n(I)$ of prime norm N.
2. Find $\beta \in I'$ with norm NS where S is powersmooth.
3. Output $J = I'\bar{\beta}/N$.

Steps 1 and 3 of this algorithm rely on the following simple result [23, Lemma 5]: if I is a left O-ideal of reduced norm N and α is an element of I, then $I\bar{\alpha}/N$ is a left O-ideal of norm $n(\alpha)/N$. Clearly, I and J are in the same equivalence class.

To compute δ in Step 1, first a Minkowski-reduced basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of I is computed. To obtain Lemma 3 below we make sure that the Minkowski basis is uniformly randomly chosen among all such bases. Then random elements $\delta = \sum_i x_i \alpha_i$ are generated with integers x_i in an interval $[-m, m]$, until the norm of δ is equal to $n(I)$ times a prime. A probable prime suffices in this context (actually Step 1 is not strictly needed but aims to simplify Step 2), so we can use the Miller-Rabin test to discard composite numbers with a large probability.

Step 2 is the core of the algorithm and actually consists of the following substeps:

2a. Find α such that $I' = O_0 N + O_0 \alpha$.
2b. Find $\beta_1 \in O_0$ with powersmooth norm S_1.
2c. Find $\beta_2 \in \mathbb{Z}\mathbf{j} + \mathbb{Z}\mathbf{k}$ such that $\alpha = \beta_1 \beta_2 \bmod NO_0$.
2d. Find β_2' and $\lambda \in \mathbb{Z}_N^*$ with powersmooth norm S_2 such that $\beta_2' = \lambda\beta_2 \bmod NO_0$.
2e. Set $\beta = \beta_1 \beta_2'$.

In Step 2a we need $\alpha \in I$ such that $\gcd(n(\alpha), N^2) = N$. This is easily achieved by starting with some $\alpha \notin \mathbb{Z}$, checking the gcd condition and dividing by some factor if necessary.

In Step 2b the algorithm actually searches for $\beta_1 = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$. A large enough powersmooth number S_1 is fixed a priori, then the algorithm generates small random values of c, d until the norm equation $a^2+b^2 = S_1-p(c^2+d^2)$ can be solved efficiently using Cornacchia's algorithm (for example, until the right hand side is a prime equal to 1 modulo 4).

Step 2c is just linear algebra modulo N. As argued in [23] it has a negligible chance of failure, in which case one can just go back to Step 2b.

In Step 2d the algorithm a priori fixes S_2 large enough, then searches for integers a, b, c, d, λ with $\lambda \notin N\mathbb{Z}$ such that $N^2(a^2+b^2) + p\left((\lambda C + cN)^2 + (\lambda D + dN)^2\right) = S_2$ where we have $\beta_2 = C\mathbf{j} + D\mathbf{k}$. If necessary S_2 is multiplied by a small prime such that $(C^2+D^2)S_2$ is a square modulo N, after which the equation is solved modulo N, leading to two solutions for λ. An arbitrary solution is chosen, and then looking at the equation modulo N^2 leads to a linear space of solutions for $(c, d) \in \mathbb{Z}_N$. The algorithm chooses random solutions until the equation

$$a^2 + b^2 = \left(S_2 - p^2\left((\lambda C + cN)^2 + (\lambda D + dN)^2\right)\right)/N^2$$

can be efficiently solved with Cornacchia's algorithm.

The overall algorithm is summarized in Algorithm 1. We now prove two lemmas on this algorithm. The first lemma shows that the output of this algorithm only depends on the ideal class of I but not on I itself. This is important in our second signature scheme, as otherwise part of the secret isogeny φ could potentially be recovered from η. The second lemma gives a precise complexity analysis of the algorithm, where [23] only showed probabilistic polynomial time complexity. Both lemmas are of independent interest.

Lemma 3 *The output distribution of the quaternion isogeny path algorithm only depends on the equivalence class of its input. (In particular, the output distribution does not depend on the particular ideal class representative chosen for this input.)*

PROOF: Let I_1 and I_2 be two left O_0-ideals in the same equivalence class, namely there exists $q \in B_{p,\infty}^*$ such that $I_2 = I_1 q$. We show that the distribution of the ideal I' computed in Step 1 of the algorithm is identical for I_1 and I_2. As the inputs are not used anymore in the remaining of the algorithm this will prove the lemma.

In the first step the algorithm computes a Minkowski basis of its input, uniformly chosen among all possible Minkowski bases. Let $B_1 = \{\alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{14}\}$ be a Minkowski basis of I_1. Then by multiplicativity of the norm we have that $B_2 = \{\alpha_{11}q, \alpha_{12}q, \alpha_{13}q, \alpha_{14}q\}$ is a Minkowski basis of I_2. The algorithm then computes random elements $\delta = \sum_i x_i \alpha_i$ for integers x_i in an interval $[-m, m]$. Clearly, for any element δ_1 computed when the input is I_1, there corresponds an element $\delta_2 = \delta_1 q$ computed when the input is I_2. This is repeated until the norm of δ is a prime times $n(I)$. As $n(I_2) = n(I_1)n(q)$ the stopping condition is equivalent for both. Finally, an ideal I of prime norm is computed as $I\bar{\delta}/n(I)$. Clearly when $\delta_2 = \delta_1 q$ we have $\frac{I_2\bar{\delta}_2}{n(I_2)} = \frac{I_1 q\bar{q}\bar{\delta}_1}{n(q)n(I_1)} = \frac{I_1\bar{\delta}_1}{n(I_1)}$. This shows that the prime norm ideal computed in Step 1 only depends on the equivalence class of the input. □

Algorithm 1 Quaternion isogeny path algorithm

Input: $\mathcal{O}_0 = \langle 1, \mathbf{i}, \frac{1+\mathbf{k}}{2}, \frac{\mathbf{i}+\mathbf{j}}{2} \rangle$, I a left $\mathcal{O}_0$ ideal.
Output: J left $\mathcal{O}_0$-ideal of powersmooth norm such that $I = Jq$ for some $q \in B_{p,\infty}$.

$\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ Minkowski-reduced basis of I.
$\alpha_i \leftarrow \{\pm\alpha_i\}$ for $i = 1, 2, 3, 4$.
loop
  $\{x_1, x_2, x_3, x_4\} \leftarrow [-m, m]^4$. Start with $m = \lceil \log p \rceil$ and do exhaustive search in the box, increasing m if necessary.
  $\delta := \sum_{i=1}^{4} x_i \alpha_i$
  if $N := n(\delta)/n(I)$ is prime **then return** $N, I' := I\bar{\delta}/n(I)$
Set an a priori powersmooth bound $s = \frac{7}{2} \log p$, and numbers S_1, S_2 with $S_1 > p \log p$, $S_2 > p^3 \log p$ and s-powersmooth product $S_1 S_2$.
Write $I' = \mathcal{O}_0 N + \mathcal{O}_0 \alpha$.
while a, b are not found **do**
  $c, d \leftarrow [-m, m]^2$, for $m = \lfloor \sqrt{S_1/2p} \rfloor$. Increase S_1 and s if necessary.
  $a, b \leftarrow$ Solution of $a^2 + b^2 = S_1 - p(c^2 + d^2)$ (solve using Cornacchia's algorithm).
$\beta_1 = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$
Set β_2 as a solution of $\alpha = \beta_1 \beta_2 \bmod N\mathcal{O}_0$.
Write $\beta_2 = C\mathbf{j} + D\mathbf{k}$. Try small primes r in increasing order until we find one such that $\left(\frac{(C^2+D^2)S_2 r}{N} \right) = 1$, and set $S_2 = S_2 r$. Update s accordingly.
$\lambda \leftarrow$ Solution of $p\lambda^2(C^2 + D^2) = S_2 \bmod N$.
while a, b are not found **do**
  $c, d \leftarrow$ Solution of $p\lambda^2(C^2 + D^2) + 2p\lambda N(Cc + Dd) = S_2 \bmod N^2$.
  $a, b \leftarrow$ Solution of $a^2 + b^2 = \left(S_2 - p^2\left((\lambda C + cN)^2 + (\lambda D + dN)^2\right)\right)/N^2$ (solve using Cornacchia's algorithm). Increase S_2 and s if necessary.
$\beta_2' = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$
$J = I' \overline{\beta_1 \beta_2'}/N$

Lemma 4 *Let $X := \max |c_{ij}|$ where $c_{ij} \in \mathbb{Z}$ are integers such that $c_{i1} + c_{i2} i + c_{i3} \frac{i+k}{2} + c_{i4} \frac{i+j}{2}$ for $1 \leq i \leq 4$ forms a $\mathbb{Z}$-basis for I. If $\log X = O(\log p)$ then under heuristic smoothness assumptions Algorithm 1 runs in time $\tilde{O}(\log^3 p)$, and produces an output of norm S with $\log(S) \approx \frac{7}{2} \log(p)$ which is $(\frac{7}{2} \log p)$-powersmooth.*

PROOF: The Minkowski basis can be computed in $O(\log^2 X)$, for example using the algorithm of [25].

As shown in [23, Section 3.1] for generic ideals the norms of all Minkowski basis elements are in $O(\sqrt{p})$. In the first loop we initially set $m = \lceil \log p \rceil$. Assuming heuristically that the numbers N generated behave like random numbers we expect the box to produce some prime number. The resulting N will be in $\tilde{O}(\sqrt{p})$. For some non generic ideals the Minkowski basis may contain two pairs of elements with norms respectively significantly smaller or larger than $O(\sqrt{p})$; in that case we can expect to finish the loop for smaller values of m by setting $x_3 = x_4 = 0$, and to obtain some N of a smaller size.

Rabin's pseudo-primality test performs a single modular exponentiation (modulo a number of size $\tilde{O}(\sqrt{p})$), and is passed by composite numbers with a probability at most $1/4$, and can be repeated r times to decrease this probability to $1/4^r$. Assuming heuristically that the numbers tested behave like random numbers the test will only be repeated a significant amount of times on actual prime numbers, so in total it will be repeated $O(\log p)$ times. This leads to a total complexity of $\tilde{O}(\log^3 p)$ bit operations for the first loop.

The other two loops involve solving equations of the form $x^2 + y^2 = M$. For such an equation to have solutions it is sufficient that M is a prime with $M = 1 \bmod 4$, a condition that is heuristically satisfied after $2 \log M$ random trials. Choosing S_1 and S_2 as in the algorithm ensures that the right-hand term of the equation is positive, and (assuming this term behaves like a random number of the same size) is of the desired form for some choices (c, d), at least heuristically. Cornacchia's algorithm runs in time $\tilde{O}(\log^2 M)$, which is also $\tilde{O}(\log^2 p)$ in the algorithm. The pseudo-primality tests will require $\tilde{O}(\log^3 p)$ operations in total, and their cost will dominate both loops.

Computing β_2 is just linear algebra modulo $N \approx \tilde{O}(\sqrt{p})$ and this cost can be neglected. The last two steps can similarly be neglected.

As a result, we get an overall cost of $\tilde{O}(\log^3 p)$ bit operations for the whole algorithm.

Let $s = \frac{7}{2} \log p$. We have $n(J) = n(I')n(\beta_1)n(\beta_2')/N^2$ so $\log n(J) \approx \frac{1}{2} \log p + \log p + 3 \log p - \log p = \frac{7}{2} \log p$. Moreover $\prod_{p_i^{e_i} < s} p_i^{e_i} \approx (s)^{s/\log s} \approx p^{7/2}$ so we can expect to find $S_1 S_2$ that is s-powersmooth and of the correct size. □

4.4 Step-by-Step Deuring Correspondence

We now discuss algorithms to convert isogeny paths into paths in the quaternion algebra, and vice versa. This will be necessary in our protocols as we are sending curves and isogenies, whereas the process uses the quaternion isogeny algorithm.

All the isogeny paths that we will need to translate in our signature scheme will start from the special j-invariant $j_0 = 1728$ mentioned above. We recall from Section 2.3 that this corresponds to the curve E_0 with equation $y^2 = x^3 + x$ and endomorphism ring $\mathrm{End}(E_0) := \langle 1, \phi, \frac{1+\pi\phi}{2}, \frac{\pi+\phi}{2} \rangle$. Moreover there is an isomorphism of quaternion algebras sending $(1, \mathbf{i}, \mathbf{j}, \mathbf{k})$ to $(1, \phi, \pi, \pi\phi)$.

For any isogeny $\varphi : E_0 \to E_1$ of degree n, we can associate a left $\mathrm{End}(E_0)$-ideal $I = \mathrm{Hom}(E_1, E_0)\varphi$ of norm n, corresponding to a left O_0-ideal with the same norm in the quaternion algebra $B_{p,\infty}$. Conversely every left O_0-ideal arises in this way [22, Section 5.3]. In our protocol we will need to make this correspondence explicit, namely we will need to pair up each isogeny from E_0 with the correct O_0 ideal. Moreover we need to do this for "large" degree isogenies to ensure a good distribution via our random walk theorem.

Translating an ideal to an isogeny path Let E_0 and $O_0 = \mathrm{End}(E_0)$ be given, together with a left O_0-ideal I corresponding to an isogeny of degree n. We assume I is given as a $\mathbb{Z}$-basis $\{\alpha_1, \ldots, \alpha_4\}$. The main idea to determine the corresponding isogeny explicitly is to determine its kernel [31].

Assume for the moment that n is a small prime. One can compute generators for all cyclic subgroups of $E_0[n]$, each one uniquely defining a degree n isogeny which can be computed with Vélu's formulae. A generator P then corresponds to the basis $\{\alpha_1, \ldots, \alpha_4\}$ if and only if $\alpha_j(P) = 0$ for all $1 \leq j \leq 4$. To evaluate $\alpha(P)$ with $\alpha \in I$ and $P \in E_0[n]$, we first write $\alpha = (u + v\mathbf{i} + w\mathbf{j} + x\mathbf{k})/2$, then we compute P' such that $[2]P' = P$ and finally we evaluate $[u]P' + [v]\phi(P') + [w]\pi(P') + [x]\pi(\phi(P'))$. As this algorithm potentially tests all possible cyclic subgroups of order n, its cost is prohibitive for large n.

When $n = \ell^e$ the degree n isogeny can be decomposed into a composition of e degree ℓ isogenies. If I is the corresponding left O_0-ideal of norm ℓ^e, then $I_i := I \bmod O_0\ell^i$ is a left O_0-ideal of norm ℓ^i corresponding to the first i isogenies. Similarly if P is a generator for the kernel of the degree ℓ^e isogeny then $\ell^{e-i+1}P$ is the kernel of the degree ℓ^i isogeny corresponding to the first i steps. One can therefore perform the matching of ideals with kernels step-by-step with successive approximations of I or P respectively. This algorithm is more efficient than the previous one, but it still requires to compute ℓ^e torsion points, which in general may be defined over a degree ℓ^e extension of $\mathbb{F}_{p^2}$. To ensure that the ℓ^e torsion is defined over $\mathbb{F}_{p^2}$ one can choose p such that $\ell^e \mid (p \pm 1)$ as in the De Feo-Jao-Plût protocols; however for general p this translation algorithm will still be too expensive.

We solve this efficiency issue by using powersmooth degree isogenies in our protocols. When $n = \prod_i \ell_i^{e_i}$ with distinct primes ℓ_i, one reduces to the prime power case as follows. For simplicity we assume that 2 does not divide n. The isogeny of degree n can be decomposed into a sequence of prime degree isogenies. For simplicity we assume the isogeny steps are always performed in increasing degree order; we can require that this is indeed the case in our protocols. Let $n_i := \prod_{j \leq i} \ell_j^{e_j}$. If I is the left O_0-ideal of norm n, then $I_i := I \bmod O_0 n_i$ is a left O_0-ideal of norm n_i corresponding to the isogeny φ_i which is a composition of all isogenies of degrees up to ℓ_i. Using a Chinese Remainder Theorem-like representation, points in $E_0[n]$ can be represented as a sequence of points in $E_0[\ell_i^{e_i}]$. Given a left-O_0 ideal I, Algorithm 2 progressively identifies the corresponding isogeny sequence.

In our protocols we will have $\ell_i^{e_i} = O(\log n) = O(\log p)$; moreover we will be using $O(\log p)$ different primes. The complexity of Algorithm 2 under these assumptions is given by the following lemma. Note that almost all primes ℓ_i are such that $\sqrt{B} < \ell_i \leq B$ and so $e_i = 1$, hence we ignore the obvious ℓ-adic speedups that can be obtained in the rare cases when ℓ_i is small.

Lemma 5 *Let $n = \prod \ell_i^{e_i}$ with $\log n, \ell_i^{e_i} = O(\log p)$. Then Algorithm 2 can be implemented to run in time $\tilde{O}(\log^5 p)$ bit operations for the first loop, and $\tilde{O}(\log^4 p)$ for the rest of the algorithm.*

PROOF: Without any assumption on p the $\ell_i^{e_i}$ torsion points will generally be defined over $\ell_i^{e_i}$ degree extension fields, hence they will be of $O(\log^2 p)$ size. However the isogenies themselves will be rational, i.e. defined over $\mathbb{F}_{p^2}$. This means their kernel is defined by a polynomial over $\mathbb{F}_{p^2}$. Isogenies over $\mathbb{F}_{p^2}$ of degree d can be evaluated at any point in $\mathbb{F}_{p^2}$ using $O(d)$ field operations in $\mathbb{F}_{p^2}$.

Let $d = \ell_i^{e_i}$. To compute a basis of the d-torsion, we first factor the division polynomial over $\mathbb{F}_{p^2}$. This polynomial has degree $O(d^2) = O(\log(p)^2)$. Using the algorithm

Algorithm 2 Translating ideal to isogeny path

Input: $\mathcal{O}_0 = \mathrm{End}(E_0) = \langle 1, \psi, \frac{1+\pi\psi}{2}, \frac{\pi+\psi}{2} \rangle$, $I = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$, $n = \prod_{i=1}^{r} \ell_i^{e_i}$ with $2 \not| n$.
Output: the isogeny corresponding to I through Deuring's correspondence.

```
for i = 1, ..., r do
    Compute a basis {P_i1, P_i2} for the ℓ_i^{e_i} torsion on E_0
    for j = 1, 2 do
        Compute P'_ij such that P_ij = [2]P'_ij
φ_0 = [1]_{E_0}
for i = 1, ..., r do
    for k = 1, 2, 3, 4 do
        α_ik = α_k with its coefficients reduced modulo ℓ_i^{e_i}.
        Write α_ik = (u_ik + v_ik i + w_ik j + x_ik k)/2.
        for j = 1, 2 do
            P_ijk = [u_ik]P'_ij + [v_ik]φ(P'_ij) + [w_ik]π(P'_ij) + [x_ik]π(φ(P'_ij))
    Solve ECDLP to compute Q_i of order ℓ_i^{e_i} such that α_ik(Q_i) = 0 for all k
    Compute φ_i = Isogeny with kernel ⟨Q_i⟩ (compute with Vélu's formulae).
    Set φ_i = φ_i φ_{i-1}
Output φ_0, φ_1, ..., φ_r.
```

in [21] this can be done in $\tilde{O}(\log^4 p)$ bit operations. Since the isogenies are defined over $\mathbb{F}_{p^2}$, this will give factors of degree at most $(d-1)/2$, each one corresponding to a cyclic subgroup. We then randomly choose some factor with a probability proportional to its degree, and we factor it over its splitting field, until we have found a basis of the d-torsion. After $O(1)$ random choices we will have a basis of the d-torsion. Each factorization costs $\tilde{O}(\log^5 p)$ using the algorithm in [29], and verifying that two points generate the d-torsion can be done with $O(d)$ field operations. It then takes $O(d)$ field operations to compute generators for all kernels. As $r = O(\log p)$ we deduce that the first loop requires $\tilde{O}(\log^5 p)$ bit operations.

Computing P_{ijk} involves Frobenius operations and multiplications by scalars bounded by d (and so $O(\log p)$ bits). This requires $O(\log p)$ field operations, that is a total of $\tilde{O}(\log^3 p)$ bit operations. Any cyclic subgroup of order $\ell_i^{e_i}$ is generated by a point $Q_i = aP_{i1} + bP_{i2}$, and the image of this point by α_{ik} is $aP_{i1k} + bP_{i2k}$. One can determine the integers a, b by an ECDLP computation or by testing random choices. There are roughly $\ell_i^{e_i} = O(\log p)$ subgroups, and testing each of them requires at most $O(\log\log p)$ field operations, so finding Q_i requires $\tilde{O}(\log p)$ field operations. Computing the isogeny can be done in $O(\log p)$ field operations using Vélu's formulae. As $r = O(\log p)$ we deduce that the second loop requires $\tilde{O}(\log^4 p)$ bit operations. □

We stress that in our signature algorithm, Algorithm 2 will be run $O(\log p)$ times. However the torsion points are independent of both the messages and the keys, so they can be precomputed. Hence the "online" running time of Algorithm 2 is $\tilde{O}(\log(p)^4)$ bit operations per execution.

Translating an isogeny path to an ideal Let $E_0, E_1, \ldots, E_r$ an isogeny path and suppose $\varphi_i : E_0 \to E_i$ is of degree $n_i = \prod_{j \le i} \ell_j^{e_j}$. We define $I_0 = O_0$. Then for

$i = 1, \ldots, r$ we compute an element $\alpha_i \in I_{i-1}$ and an ideal $I_i = I_{i-1}\ell_i^{e_i} + I_{i-1}\alpha_i$ that corresponds to the isogeny ϕ_i. At step i, we use a basis of I_{i-1} to compute a quadratic form f_i that is the norm form of the ideal I_{i-1}. The roots of this quadratic form modulo $\ell_i^{e_i}$ correspond to candidates for α_i and hence I_i. Note that this correspondence is not injective: a priori there will be $O((\ell_i^{e_i})^3)$ roots but there are only $O(\ell_i^{e_i})$ corresponding ideals including the correct one. Our strategy is to pick random solutions to the quadratic form until the maps α_i and ϕ_i have the same kernels.

Algorithm 3 Translating isogeny path to ideal

Input: $E_0, E_1, \ldots, E_r$ isogeny path, $\phi_i : E_{i-1} \to E_i$ of degree $\ell_i^{e_i}$.
Output: the ideal path $I_0, \ldots, I_r$ corresponding to the isogeny path.
1: Let $I_0 = \mathcal{O}_0$
2: **for** $i = 1, \ldots, r$ **do**
3: Find Q_i of order $\ell_i^{e_i}$ that generates the kernel of ϕ_i
4: Compute $[\beta](Q_i)$ for all $\beta \in \{1, \mathbf{i}, \frac{\mathbf{i}+\mathbf{j}}{2}, \frac{1+\mathbf{k}}{2}\}$
5: Let $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ a basis of I_{i-1}
6: Let $f_i(w, x, y, z) = n(w\beta_1 + x\beta_2 + y\beta_3 + z\beta_4)$
7: **repeat**
8: Pick a random solution to $f_i(w, x, y, z) = 0 \bmod \ell_i^{e_i}$
9: Set $\alpha_i = w\beta_1 + x\beta_2 + y\beta_3 + z\beta_4$
10: **until** $[\alpha_i](Q_i) = \infty$
11: Set $I_i = I_{i-1}\ell_i^{e_i} + I_{i-1}\alpha_i$
12: Perform basis reduction on I_i

In our protocols we will have $\ell_i^{e_i} = O(\log n) = O(\log p)$; moreover we will be using $O(\log p)$ different primes. The complexity of Algorithm 3 under these assumptions is given by the following lemma.

Lemma 6 *Let $n = \prod \ell_i^{e_i}$ with $\log n, \ell_i^{e_i} = O(\log p)$, and assume all the isogenies are defined over $\mathbb{F}_{p^2}$. Then Algorithm 3 can be implemented to run in expected time $\tilde{O}(\log^4 p)$.*

PROOF: We remind that without any assumption on p the $\ell_i^{e_i}$ torsion points will generally be defined over $\ell_i^{e_i}$ degree extension fields, hence they will be of $O(\log^2 p)$ size. Isogenies of degree d can be evaluated at any point using $O(d)$ field operations.

The isogeny ϕ_i is naturally given by a polynomial ψ_i such that the roots of ψ_i correspond to the x-coordinates of affine points in $\ker \varphi_i$. To identify a generator Q_i we first factor ψ_i over $\mathbb{F}_{p^2}$. Using the algorithm in [29] this can be done with $\tilde{O}(\log^3 p)$ bit operations. We choose a random irreducible factor with a probability proportional to its degree, we use this polynomial to define a field extension of $\mathbb{F}_{p^2}$, and we check whether the corresponding point is of order $\ell_i^{e_i}$. If not we choose another irreducible factor and we repeat. We expect to only need to repeat this $O(1)$ times, and each step requires $\tilde{O}(\log p)$ bit operations. So the total cost for Step 3 is $\tilde{O}(\log^3 p)$.

Step 4 requires $O(\log \log p)$ field operations to compute a point Q_i' such that$[2]Q_i' = Q_i$. After that it mostly requires $O(\log p)$ field operations to compute the Frobenius map. The total cost of this step is therefore $\tilde{O}(\log^3 p)$.

To compute a random solution to f_i modulo $\ell_i^{e_i}$, we choose uniformly random values for w, x, y, and when the resulting quadratic equation in z has solutions we choose a random one. As $\ell_i^{e_i} = O(\log p)$ the cost of this step can be neglected. Computing $[\alpha_i](Q_i)$ requires $O(\log\log p)$ operations over a field of size $O(\log^2 p)$. On average we expect to repeat the loop $O(\ell_i^{e_i}) = O(\log p)$ times, resulting in a total cost of $\tilde{O}(\log^3 p)$. Computing each f_i costs $\tilde{O}(\log p)$ bit operations.

The value α_i is represented as a $\mathbb{Z}$-linear combination with respect to the $\mathbb{Z}$-basis of O_0, and the size of the coefficients is increased by $\log_2(\ell_i^{e_i})$ over the size of the coefficients in the representation of the β_j. The product $I_{i-1}\alpha_i$ thus has coefficients more than doubled in bitlength. However, there exists a representation of the ideal with small coefficients, since the degree of the ideal is bounded in terms of the degree of the isogeny from E_0 to E_i. Hence there exists a reduced basis with smaller coefficients.

Overall, one can show that all the coefficients of quaternion algebra elements in the algorithm have size $O(\log(p))$ bits. As $r = O(\log p)$ the total cost of the algorithm is $\tilde{O}(\log^4 p)$. $\square$

Note that the output is an ideal represented with coefficients of size $O(\log p)$, and so the required condition $\log B = O(\log p)$ in Lemma 4 is satisfied.

4.5 Signature Scheme based on Endomorphism Ring Computation

In this section we give the details of our second signature scheme based on our new identification protocol, with security relying on computing the endomorphism ring of a supersingular elliptic curve.

Key Generation Algorithm: On input a security parameter λ generate a prime p with 2λ bits, which is congruent to 3 modulo 4. Fix B, S_1, S_2 as small as possible[7] such that $S_k := \prod_i \ell_{k,i}^{e_{k,i}}$, $\ell_{k,i}^{e_{k,i}} < B$, $\gcd(S_1, S_2) = 1$, and $\prod \left(\frac{2\sqrt{\ell_{k,i}}}{\ell_{k,i}+1}\right)^{e_{k,i}} < \frac{72}{p^2}$. Perform a random isogeny walk of degree S_1 from the curve E_0 with j-invariant $j_0 = 1728$ to a curve E_1 with j-invariant j_1. Compute $O_1 = \mathrm{End}(E_1)$ and the ideal I corresponding to this isogeny. Choose a hash function H with at least $t = t(\lambda)$ bits of output (in practice, depending on the security requirement, either $t = \lambda$ or $t = 2\lambda$). The public key is $\textsc{pk} = (p, j_0, j_1, H)$ and the secret key is $\textsc{sk} = (E_0, E_1, O_1, I)$.

Signature Algorithm: On input a message m and keys ($\textsc{pk}, \textsc{sk}$), recover the parameters p and j_1. For $i = 1, \ldots, t$, generate a random isogeny walk w_i of degree S_2, ending at a j-invariant $j_{2,i}$. Compute $h := H(m, j_{2,1}, \ldots, j_{2,t})$ and parse the output as t challenge bits b_i. For $i = 1, \ldots, t$, if $b_i = 1$ use w_i and Algorithm 2 of Section 4.4 to compute the corresponding path in the quaternion algebra, then use the algorithm of Section 4.3 to compute a "fresh" path between O_0 and $O_{2,i}$, and finally use Algorithm 3 to compute an isogeny path w_i' from j_0 to $j_{2,i}$. If $b_i = 0$ set $z_i := w_i$, otherwise set $z_i := w_i'$. Return the signature $\sigma = (h, z_1, \ldots, z_t)$.

[7] The exact procedure is irrelevant here.

Verification Algorithm: On input a message m, a signature σ and a public key PK, recover the parameters p and j_1. For each $1 \leq i \leq t$ one uses z_i to compute the image curve $E_{2,i}$ of the isogeny. Hence the verifier recovers the signature components $j_{2,i}$ for $1 \leq i \leq t$. The verifier then recomputes the hash $H(m, j_{2,1}, \ldots, j_{2,t})$ and checks that the value is equal to h, accepting the signature if this is the case and rejecting otherwise.

We now show that this scheme is a secure signature.

Theorem 8. *If Problem 6 is computationally hard then the second signature scheme is secure in the random oracle model under a chosen message attack.*

PROOF: As shown in Section 4.2, if Problem 6 is computationally hard then the identification scheme (sigma protocol) has 2-special soundness and honest-verifier zero-knowledge. It follows by the same arguments as in Section 4 that the identification scheme is secure against impersonation under passive attacks. Theorem 3 then implies that the signature scheme is secure in the random oracle model. $\square$

One can also apply the Unruh transform described in Section 2.6 to obtain a signature scheme that is proven secure against quantum adversaries.

Theorem 9. *If Problem 6 is computationally hard for a quantum computer then the signature scheme obtained from the Unruh transform on the sigma protocol of Section 4.1 is a secure signature scheme against quantum adversaries in the random oracle model.*

Efficiency: As the best classical algorithm for computing the endomorphism ring of a supersingular elliptic curve runs in time $\tilde{O}(\sqrt{p})$ one can take $\log p = 2\lambda$. By Lemma 1 taking $B \approx 4 \log p$ ensures that the outputs of random walks are distributed uniformly enough. Random walks then require $4 \log p$ bits to represent, so signatures are

$$t + \frac{t}{2}\left(4\lceil \log p \rceil + \frac{7}{2}\lceil \log p \rceil\right)$$

bits on average, depending on the challenge bits. For λ bits of security, we choose $t = \lambda$, so the signature length is approximately $\lambda + (\lambda/2)(8\lambda + 7\lambda) \approx 15\lambda^2/2$. For non-repudiation, we choose $t = 2\lambda$, and the signature length is about $15\lambda^2$ bits.

Private keys are 2λ bits and public keys are $3 \log p = 6\lambda$ bits. A signature mostly requires 2λ calls to the Algorithm of Sections 4.3 and 4.4, for a total cost of $\tilde{O}(\lambda^5)$. Verification requires to check $O(\lambda)$ isogeny walks, each one comprising $O(\lambda)$ steps with a cost $\tilde{O}(\lambda^3)$ each, hence a total cost of $\tilde{O}(\lambda^5)$ bit operations.

Optimization with Non Backtracking Walks: In our description of the signature scheme we have allowed isogeny paths to "backtrack". We made this choice to simplify the convergence analysis of random walks and because it does not affect the asymptotic complexity of our schemes significantly. However in practice at any concrete security parameter, it will be better to use non-backtracking random walks as they will converge more quickly to a uniform distribution [2].

5 Conclusion

We have presented the first two signature schemes based on supersingular isogeny problems. Both schemes are built from a parallel execution of an identification scheme with bounded soundness, using the Fiat-Shamir transform. Our first scheme is built directly from the De Feo-Jao-Plût identification protocol with some optimization, the second one is more involved and crucially relies on the quaternion ℓ-isogeny algorithm of Kohel-Lauter-Petit-Tignol. The first scheme is significantly more efficient, but the second one is based on an arguably more standard and potentially harder computational problem.

Our schemes rely on problems that can potentially resist to quantum algorithms. However this family of problems are also are rather new in cryptography. Among all of them, we believe that the problem of computing the endomorphism ring of a supersingular elliptic curve (on which our second signature scheme relies) is the most natural one to consider from an algorithmic theory point of view, and it was the subject of Kohel's PhD thesis in 1996. The problem is also potentially harder than Problems 3 and 4 considered in previous works (and used in our first signature scheme). Yet, even that problem is far from having received the same scrutiny as more established cryptography problems like discrete logarithms or integer factoring. We hope that this paper will encourage the community to study its complexity.

References

1. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433. Springer, 2002.
2. Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 9(4):585–603, 2007.
3. Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key compression for isogeny-based cryptosystems. In *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography*, AsiaPKC '16, pages 1–10, New York, NY, USA, 2016. ACM.
4. Mihir Bellare, Bertram Poettering, and Douglas Stebila. From identification to signatures, tightly: A framework and generic transforms. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 435–464. Springer, 2016.
5. Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2014.
6. Gaetan Bisson and Andrew V. Sutherland. Computing the endomorphism ring of an ordinary elliptic curve over a finite field. *IACR Cryptology ePrint Archive*, 2009:100, 2009.
7. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.

8. Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Mathematical Cryptology*, 8(1):1–29, 2014.
9. Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 572–601, 2016.
10. Max Deuring. Die typen der multiplikatorenringe elliptischer funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941.
11. L. Dewaghe. Un corollaire aux formules de Vélu. Preprint, 1995.
12. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology*, 8(3):209–247, 2014.
13. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.
14. Steven D. Galbraith. Constructing Isogenies Between Elliptic Curves Over Finite Fields. *LMS J. Comput. Math*, 2:118–138, 1999.
15. Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91. Springer, 2016.
16. Steven Goldfeder, Melissa Chase, and Greg Zaverucha. Efficient post-quantum zero-knowledge and signatures (draft). eprint 2016/1110, 2016.
17. Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.
18. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, pages 19–34, 2011.
19. David Jao and Vladimir Soukharev. Isogeny-based quantum-resistant undeniable signatures. In *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, pages 160–179, 2014.
20. Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.
21. Kiran S. Kedlaya and Christopher Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.
22. David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.
23. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17A:418–432, 2014.
24. Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Hash function requirements for schnorr signatures. *J. Mathematical Cryptology*, 3(1):69–87, 2009.
25. Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Transactions on Algorithms*, 5(4), 2009.
26. Christophe Petit. On the quaternion ℓ-isogeny problem. Presentation slides from a talk at the University of Neuchâtel, March 2015.
27. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Springer Verlag, 1986.
28. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 755–784, 2015.
29. Joachim von zur Gathen and Victor Shoup. Computing Frobenius maps and factoring polynomials. *Computational Complexity*, 2:187–224, 1992.
30. Jacques Vélu. Isogénies entre courbes elliptiques. *Communications de l'Académie royale des Sciences de Paris*, 273:238–241, 1971.

31. William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'E.N.S.*, 2:521–560, 1969.
32. Sun Xi, Haibo Tian, and Yumin Wang. Toward quantum-resistant strong designated verifier signature from isogenies. *International Journal of Grid and Utility Computing*, 5(2):292–296, September 2012.

A Proof of Lemma 1

We have

$$\prod_{\substack{\ell_i^{e_i} < B \\ \ell_i \text{ prime} \\ e_i \text{ maximal}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right)^{e_i} > \prod_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right) > \prod_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \left(\frac{\sqrt{\ell_i}}{2}\right).$$

Taking logarithm, using the prime number theorem and replacing the sum by an integral we have

$$\log \prod_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \left(\frac{\sqrt{\ell_i}}{2}\right) = \sum_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \frac{1}{2} \log \ell_i - \sum_{\substack{\ell_i < B \\ \ell_i \text{ prime}}} \log 2 \approx \frac{1}{2} \int_1^B \log x \frac{1}{\log x} dx - \frac{B}{\log B} =$$
$$= \frac{1}{2} B - \frac{B}{\log B} \approx \frac{1}{2} B.$$

if B is large enough. Then, we choose $c = 4$, obtaining $\frac{1}{2} B = 2 \log p > \log(p^2/72)$.