# Signature Schemes based on Supersingular Isogeny Problems

Steven D. Galbraith[1], Christophe Petit[2], and Javier Silva Velón[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.SilvaVelon@maths.ox.ac.uk

**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 [12, 13, 16, 22, 23]. Unlike other elliptic curve cryptosystems, the only known quantum algorithm for these problems, due to Biasse-Jao-Sankar [6], 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 [12] gave a collision-resistant hash function. Jao-De Feo [22] gave a key exchange protocol, De Feo-Jao-Plût [16] gave a public key encryption scheme and an interactive identification protocol, Jao-Soukharev [23] gave an undeniable signature, and Xi-Tian-Wang [38] 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 [16] interactive identification protocol by using the Fiat-Shamir transform to turn it into a non-interactive 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 [16], 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 also has running time of $\tilde{O}(p^{1/4})$. The

recent paper by Galbraith-Petit-Shani-Ti [19] 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. 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 from the Fiat-Shamir protocol in the usual way.

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 $\phi$ 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 $\psi$. 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 [27] to produce an isogeny $\mu : E_0 \to E_2$ that is completely independent of $\phi$. 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, 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 $\varphi$ is the degree of $\varphi$ as a morphism. An isogeny of degree $\ell$ is called an $\ell$-isogeny. If $\varphi$ is separable, then $\deg \varphi = \# \ker \varphi$. If there is a separable isogeny between two curves, we say that they are *isogenous*. 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 [37]. Given a subgroup $G$ of $E$, we can use Vélu's formulae [33] to explicitly obtain an isogeny $\varphi : E \to E'$ with kernel $G$ and such that $E' \cong E/G$. These formulas involve sums over $G$, so using them is efficient as long as $\#G$ is small or powersmooth. Given a prime $\ell$, the torsion group $E[\ell]$ contains exactly $\ell + 1$ cyclic subgroups of order $\ell$, each one corresponding to a different isogeny.

A composition of $n$ separable isogenies of degrees $\ell_i$ for $1 \leq i \leq n$ gives an isogeny of degree $N = \prod_i \ell_i$ with kernel a group $G$ of order $N$. For any permutation

$\sigma$ 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 $\varphi$, and which verifies that $\varphi\hat{\varphi} = \hat{\varphi}\varphi = [\deg \varphi]$.

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 $\varphi, \varphi'$ are *isomorphisms*, and that $E, E'$ are *isomorphic*. Isomorphism classes of elliptic curves over $\mathbb{F}_q$ can be labeled with their $j$-invariant [31, 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 $\infty$.

In the case of supersingular elliptic curves, there is always one 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 [15] gives an equivalence of categories between the $j$-invariants of 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 theorem above.

**Problem 1** *Let $p, \ell$ 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 $\ell^k$ from $E$ to $E'$.*

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

**Problem 2** *Let $p, \ell$ 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 $\ell^{k_1}$ and $\ell^{k_2}$, respectively, from $E$ to $E'$.*

The hardness assumption of the second problem has been used in [12] 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 [16] 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 [16] relies on problems 3 and 4 below (which De Feo-Jao-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 $\ell_1$. 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 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 [16].

**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 $\ell_1^{e_1}$. 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 various auxiliary points, 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. Find 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 [26], 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 [18]. 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 full 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 $\ell$-isogeny algorithm of Kohel-Lauter-Petit-Tignol [27, 30], which solves the following problem:

**Problem 7** *Let $p, \ell$ 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 $\ell^k$ 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 for $B \approx \frac{7}{2}\log p$ and $O(\log p)$ different prime powers, instead of ideals of norm a power of $\ell$. This is in fact the version that we will use 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 [7], whereas the best known algorithms for computing isogenies are still exponential. There is, however, a subexponential quantum algorithm for computing isogeny between ordinary curves [6].

### 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 $\ell$ isogeny between the corresponding vertices.

Isogeny graphs are regular 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 are also very good *expander graphs* [21]; 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 $\ell$ be as above, and let $j$ be a supersingular invariant in characteristic $p$. We define a random step of degree $\ell$ from $j$ as the process of randomly and uniformly choosing a neighbour of $j$ in the $\ell$-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 $\ell_i$ from $j_{i-1}$. We do not require the primes $\ell_i$ 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[3] of classical random walk theorems [21]:

**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 $A_{\ell_i}$ be the normalized adjacency matrix of the $\ell_i$-isogeny graph. Clearly $A_{\ell_i}$ 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 $\lambda_{2i}$ be the second largest eigenvalue of $A_{\ell_i}$ in absolute value.

If step $k$ is of degree $\ell_i$ 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

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

since $||v_0 - u||_2^2 = (1 - \frac{1}{N_p})^2 + \frac{N_p - 1}{N_p}(\frac{1}{N_p})^2 = \frac{(N_p-1)^2 + (N_p-1)}{N_p^2} = \frac{N_p - 1}{N_p} \leq 1$. (XXXX I think this is wrong: second equality does not hold. I suggest it is $\leq 1 - 2/N + 2/N^2 < 1$. XXXXXX) 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 protocols we will want the right-hand term to be smaller than $\frac{1}{2N_p} \approx \frac{6}{p}$, and at the same time we will want the powersmooth bound $B$ to be as small as possible. The following lemma shows that taking $B \approx 2 \log p$ suffices asymptotically.

[3] 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.

**Lemma 1** *There is a function $c_p = c(p)$ such that $\lim_{p\to\infty} c_p = 2$, and, for each $p$,*

$$\prod_{\substack{\ell_i \text{ prime} \\ e_i := \max\{e \mid \ell_i^e < c_p \log p\}}} \left(\frac{\ell_i + 1}{2\sqrt{\ell_i}}\right)^{e_i} > \frac{p}{6}.$$

We refer to Appendix A for the proof of this lemma. For concrete values, in particular for $p$ of length 128 and 256, it is easy to verify that the inequality holds for $B = c \log p$, for the values $c = 2.07$ and $c = 1.93$, 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 $\ell_i$ where $1 \leq i \leq n$. 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 $\ell$ (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

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

This representation is possible due to our convention the isogeny degrees are increasing and since the degrees are all small. To obtain the sequence of isogeny degrees from $N$ one factors using trial division and processes the primes in order of size.

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 $\Phi_{\ell_i}$ is the $\ell_i$-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 $\ell$ 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 $\ell^n$ isogeny. It suffices to send $\alpha$, 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 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 $\alpha$ (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 $\ell_i+1$ 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 $\ell_i$ 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.[4] 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

[4] In the most general case, when all primes $\ell_i$ 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.

$\ell_i = 2$ 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 $\ell_i$-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 $\{0, \ldots, \ell\}$. One can extend this canonical ordering to kernels of composite degrees in various simple ways (see also [4, 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 $\mathbb{F}_{p^2}$, in practice 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 $\infty$ can be canonically represented as $\mathbb{Q}\langle i, j\rangle$, where $i^2 = -1$, $j^2 = -p$ and $k := ij = -ji$. The maximal order $O_0$ with $\mathbb{Z}$-basis $\{1, i, \frac{1+k}{2}, \frac{i+j}{2}\}$ 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 $(1, i, j, k)$ 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 $\ell$ can be described by a single integer in $\{0, \ldots, \ell\}$. 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 $\prod_i (\ell_i + 1)^{e_i}$. 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 $\ell_i$ torsion (which may be over a large field extension), then to sort with respect to some canonical ordering the cyclic subgroups of order $\ell_i$ to identify the correct one, and finally to compute the next $j$-invariant with Vélu's formulae [36]. In the second case the verifier will need at each step given a $j$-invariant, to specialize one variable of the $\ell_i$-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 $\ell_i$. 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 [35] 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 $\ell_i$ 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 Security Definitions

An identification protocol will define the interaction between two parties, called a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, in which $\mathcal{P}$ tries to prove to $\mathcal{V}$ that they know the secret key corresponding to a given public key, without revealing any additional information about the secret. We use the terminology and notation of Abdalla-An-Bellare-Namprempre [1] (also see Bellare-Poettering-Stebila [5]).

A identification scheme is called *canonical* (or *three-move*) if is has the form of sigma-protocol between prover and verifier where the prover first sends a commitment $Y$, then receives a challenge $c$ from the verifier, and answers with a response $z$. The scheme is called *non-trivial* if $c$ is chosen from an exponentially large set in terms of the security parameter. See [1, 5] for more discussion.

We want an identification protocol to satisfy the following properties:

- **Correctness (or completeness):** if the prover really knows the secret, he should be able to convince the verifier.
- **Soundness:** if the prover does not know the secret, he should not be able to convince the verifier.
- **Honest Verifier Zero-knowledge:** the interaction does not reveal any information about the secret.

To define 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, it is possible 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.

For zero-knowledge, the idea is that if it is possible, given the challenge, to simulate the whole protocol without actually knowing the secret, then the protocol leaks no information about the secret. A simulator should be able to produce transcripts of the interaction that have indistinguishable distribution from the real interactions.

An identification scheme can be repeated several times in parallel to obtain another identification scheme with better soundness guarantees.

For signature schemes we use the standard definition of *existential unforgeability under chosen message attacks* [25]. An adversary can ask for polynomially many signatures of messages of his choice to a signing oracle $\mathsf{Sign}_{sk}(\cdot)$. 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.

### 2.5 Fiat-Shamir Transform

We use a common construction to build a signature scheme from an identification scheme, introduced by Fiat-Shamir [17].

The Fiat-Shamir idea is to make the interactive protocol non-interactive by using a random oracle to produce challenges and use the answers to the challenges as the signature.

The public key and secret key are the public key and the secret key from the identification protocol, respectively. In our basic identification schemes challenges are just one bit. To make our schemes non-trivial we will repeat the identification protocol $t$ times in parallel, so we have a list of commitments $\{a_i\}_{i=1}^t$, challenges $\{c_i\}_{i=1}^t$, where each $c_i$ is a bit, and responses $\{z_i\}_{i=1}^t$. The signature scheme derived from the identification protocol is as follows. Let $H$ be a random oracle that outputs a bit string of length $t$.

- $\mathsf{Gen}(\lambda)$: this is the same as in the identification protocol.
- $\mathsf{Sign}(sk, m)$: compute the commitments $\{a_i\}_{i=1}^t$ and then compute $h = H(m, \{a_i\}_{i=1}^t)$. Consider each bit of the hash as a challenge $c_i$ to the corresponding $a_i$, and produce an answer $z_i$. The signature is $(\sigma_1, \sigma_2) = (\{a_i\}_{i=1}^t, \{z_i\}_{i=1}^t)$.
- $\mathsf{Verify}(m, \sigma, pk)$: compute $H(m, \{a_i\}_{i=1}^t)$, and check whether each $z_i$ is a valid answer for the commitment $a_i$ and the corresponding bit of the hash as the challenge. If the answers are valid for all $i = 1, \ldots, t$, output 1, otherwise output 0.

Abdalla-An-Bellare-Namprempre [1] (also see Bellare-Poettering-Stebila [5]) 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.* With identification schemes of a certain type there is a major reduction in signature size which does not affect the security proof. Suppose that one can recover the values $a_i$ from the public key and the values $h$ and $z_i$. Then one can get the following signature scheme.

- $\mathsf{Gen}(\lambda)$: this is the same as in the identification protocol.
- $\mathsf{Sign}(sk, m)$: compute the commitments $\{a_i\}_{i=1}^t$ and then compute $h = H(m, \{a_i\}_{i=1}^t)$. Consider each bit of the hash as a challenge to the corresponding $a_i$, and produce an answer $z_i$. The signature is $(h, \sigma_2) = (h, \{z_i\}_{i=1}^t)$.
- $\mathsf{Verify}(m, \sigma, pk)$: compute $\{a_i\}_{i=1}^t$ from the public key and the signature. Compute $h' = H(m, \{a_i\}_{i=1}^t)$, and accept the signature if $h' = h$.

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.

There is interesting ongoing research on the security of Fiat-Shamir-type signatures against quantum computers. Clearly preimages, second preimages and collisions can be computed faster using quantum computers than with classical computers [10, 20] so the hash function output size must be increased accordingly. Some quantum attack models grant the attacker with the ability to perform signature queries in superposition, in which case many constructions that are secure classically become insecure [9, 24]. The quantum random oracle model was introduced as a replacement for the classical random oracle model, but adapting some classical proofs to this model has appeared to be very challenging [8]. Current research aims at proving or disproving the security of Fiat-Shamir signatures in this model [3, 8, 14], and at offering easy alternative ways to convert interactive identification schemes into secure signature schemes [32]. Of course even classically, a proof in the random oracle model can at best be considered as a security argument as the model is not sound in general [11].

In this paper our focus is on how to build signature schemes based on supersingular isogeny problems, so we leave these independent considerations aside and we focus on the classical security of our schemes, in the random oracle model.

## 3 First Signature Scheme

This section presents a signature scheme obtained from the interactive identification protocol of De Feo-Jao-Plût [16]. First we describe their scheme.

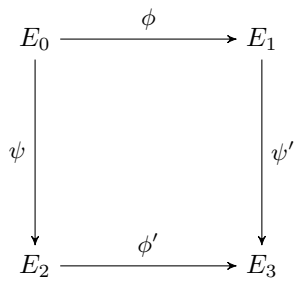
### 3.1 De Feo-Jao-Plût Interactive Protocol

Let $p$ be a large prime of the form $\ell_1^{e_1}\ell_2^{e_2} \cdot f \pm 1$, where $\ell_1, \ell_2$ 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 $E_1 = E_0/\langle P_1 \rangle$.

The secret is an $\ell_1^{e_1}$-isogeny $\phi : E_0 \to E_1$, whereas both these curves are public. A pair of generators $R_2, S_2$ of $E_0[\ell_2^{e_2}]$, and the images $\phi(R_2), \phi(S_2)$ are also public.

The interaction goes as follows:

1. The prover chooses a random primitive $\ell_2^{e_2}$-torsion point $P_2$, defines the curves $E_2 = E_0/\langle P_2 \rangle$ and $E_3 = E_0/\langle P_1, P_2 \rangle$, and uses Vélu's formulae to compute the diagram



and reveals $E_2$ and $E_3$ to the verifier.

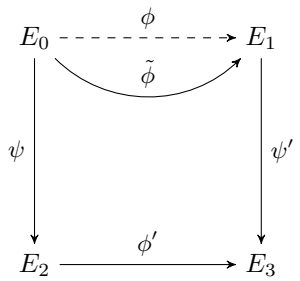
2. The verifier challenges the prover with a random bit $b \leftarrow \{0,1\}$.
3. If $b = 0$, the prover reveals $P_2$ and $\phi'(P_2)$.
   If $b = 1$, the prover reveals $\psi(P_1)$.

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 proved by De Feo-Jao-Plût [16].

**Theorem 3.** *If Problems 3 and 4 are computationally hard, then the interactive protocol defined above is an identification protocol.*

For future reference we sketch the ideas in the proof: Completeness is straightforward. For soundness: Suppose $\mathcal{A}$ is an adversary that takes as input the public key and succeeds in the identification protocol with a non-negligible advantage. 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. We choose one of the sessions of the interactive protocol, where $\mathcal{A}$ sends $E_2, E_3$ and receives a bit $b$. Suppose that $\mathcal{A}$ can answer both challenges $b = 0$ and $b = 1$ successfully (since $\mathcal{A}$ has non-negligible advantage there is some such session with non-negligible probability). By the standard oracle replay attack on $\mathcal{A}$ we get consistent responses to both challenges, and hence the following diagram.




From this, one has an explicit description of an isogeny $\tilde{\phi}$ 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 $\phi$, as desired.

Finally we need to prove zero-knowledge. For this one simulates 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.

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

We now fully specify our signature scheme based on De Feo-Jao-Plût identification protocol.

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.

*Key Generation Algorithm* Gen*:* On input a security parameter $\lambda$ generate a prime $p$ with at least $4\lambda$ bits, such that $p = \ell_1^{e_1}\ell_2^{e_2}f \pm 1$, with $\ell_1, \ell_2, f$ small (ideally $f = 1$, $\ell_1 = 2$, $\ell_2 = 3$) and $\ell_1^{e_1} \approx \ell_2^{e_2}$. Choose[5] 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 $H$ 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 $(E_0, P_1)$, and the public key is $(p, j_0, j_1, R_2, S_2, R_2', S_2', H)$.

*Signature Algorithm* Sign*:* 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 $\ell_1$-division polynomial to choose at each step of a non-backtracking walk). $\phi_i'$ into a bitstring $z_i$. Return the signature $\sigma = (h, z_1, \ldots, z_t)$.

*Verification Algorithm* Verify*:* On input a message $m$, a signature $\sigma$ 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

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

[5] Costello-Longa-Naehrig [13] 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.

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

**Theorem 4.** *The first signature scheme is secure in the random oracle model under a chosen message attack.*

PROOF: This follows immediately from Theorem 2 and Theorem 3. □

*Efficiency* As isogenies are of degree roughly $\sqrt{p}$, the scheme requires to use primes $p$ of size $4\lambda$ 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 1.

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 $\ell_1$ or $\ell_2$, 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 [28]. In both settings, the choice of hash function should be made carefully.

**Lemma 1.** *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 $\lambda$ bits of security is approximately $6\lambda^2$ 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 $\alpha_i$ 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 $\beta_i$ 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 $\ell_1$-th modular polynomial. This also can be done in $\lceil \log_2(\ell_1^{e_1}) \rceil$ bits.

For security level $\lambda$ one can take $t = \lambda$, $\ell_1^{e_1} \approx \ell_2^{e_2} \approx 2^\lambda$ 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 is revealed.

### 4.1 New Protocol 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 signature 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 $O_1$.

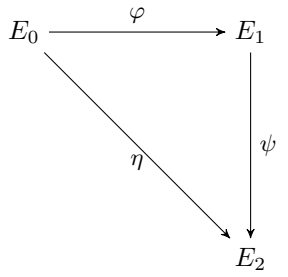
**Find new path:** Given an ideal $I$ corresponding to an isogeny $E_0 \to E_2$, to produce a new ideal $J$ corresponding to a "random" isogeny $E_0 \to E_2$ of powersmooth degree.

**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 $E_2$.

We now sketch the interaction between the prover and the verifier.

1. The prover performs a random walk of sufficiently large degree 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 $\psi$.
   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 $\ell$-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 $\eta$ 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.

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




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 5.** *If Problem 6 is computationally hard, then the interactive protocol defined above is an identification protocol.*

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 remainder of this section we first give a proof of Theorem 5, then we provide details of the algorithms involved in our identification protocol, and finally we describe the resulting signature scheme.

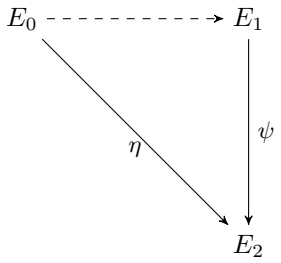
### 4.2 Proof of Theorem 5

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

**Completeness.** Let $\varphi$ 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.

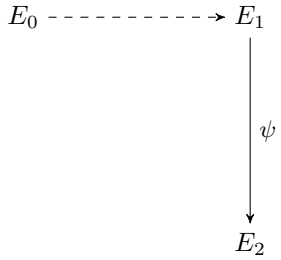
**2-special soundness.** Let $\mathcal{A}$ be a forger against the identification scheme. We describe an extractor algorithm that takes two curves $(E_0, E_1)$ and computes a path between them. Run $\mathcal{A}$ on input $(E_0, E_1)$ so that it outputs a commitment $E_2$. The extractor answers with challenge $b = 0$ and then re-winds $\mathcal{A}$ and answers with challenge $b = 1$. With non-negligible probability $\mathcal{A}$ outputs two valid answers $\psi : E_1 \to E_2$,

$\eta : E_0 \to E_2$ to these challenges. 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 $\psi$. This is summarized in the following diagram:



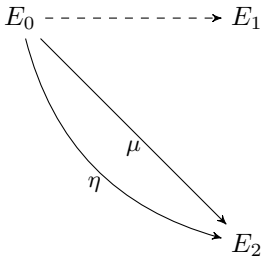

**Zero-knowledge.** We shall prove that there exists a probabilistic polynomial time simulator $\mathcal{S}$ that outputs transcripts indistinguishable from transcripts of interactions with an honest verifier.

– 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 $(E_2, 0, \psi)$.




In this case, it is clear that the distributions of every element in the transcript is 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 the protocol to produce another isogeny $\eta : E_0 \to E_2$. The simulator outputs the transcript $(E_2, 1, \eta)$.



The reason to output $\eta$ instead of $\mu$ is to ensure that the transcript distributions are indistinguishable from the distributions of real interaction transcripts.

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 $E$ in the graph and $N_p \approx p/12$ the number of vertices in the graph:

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

Therefore

$$\begin{aligned}
|\Pr(X_r = E) - \Pr(X_s = E)| &= \left|\Pr(X_r = E) - \frac{1}{N_p} - \left(\Pr(X_s = E) - \frac{1}{N_p}\right)\right| \leq \\
&\leq \left|\Pr(X_r = E) - \frac{1}{N_p}\right| + \left|\Pr(X_s = E) - \frac{1}{N_p}\right| \leq \\
&\leq \frac{1}{2N_p} + \frac{1}{2N_p} = \frac{1}{N_p} = \mathrm{negl}(\log p).
\end{aligned}$$

Therefore the distributions of $E_2$ are indistinguishable. Now, since $\eta$ is produced in the same way from $E_0$ and $E_2$, we have that the distributions of $\eta$ in both cases are indistinguishable.

### 4.3 Quaternion Isogeny Path Algorithm

In this section we sketch the quaternion isogeny algorithm from Kohel-Lauter-Petit-Tignol [27] 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.)

XXX Our analysis only heuristic too right??

The algorithm takes as input two maximal orders $O, O'$ in the quaternion algebra $B_{p,\infty}$, and it returns a sequence of $O$-left 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 $\ell^e$ 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. We stress that these powersmooth isogeny degrees are chosen by the signer and do not rely on any smoothness heuristics.

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 $O_0 = \langle 1, i, \frac{1+k}{2}, \frac{i+j}{2} \rangle$.

We focus on the case where $O = O_0$, 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 $O_0$. The equivalence class on 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 $q \in B_{o,\infty}$. Therefore our goal is, given a left $O_0$ ideal $I$, to compute another left $O_0$ ideal $J$ with powersmooth norm. The algorithm proceeds as follows:

1. Compute 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 [27, Lemma 5]: if $I$ is a left $O$-ideal of reduced norm $N$ and $\alpha$ 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 $\delta$ in Step 1, first a Minkowski-reduced basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of $I$ is computed and 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 $\delta$ is equal to $n(I)$ times a prime. Taking $m$ polynomial in $\log p$ suffices in practice, and with a very large probability it leads to $N$ prime of size $\tilde{O}(\sqrt{p})$ after $O(\log p)$ random trials [27, Section 3.1]. The Minkowski basis can be computed in $O(\log^2 B)$, where $B$ is a bound on the coefficients of the basis elements given as inputs for $I$ [29]. In our signature scheme we will have $\log B = O(\log p)$. A probable prime suffices in this context (actually Step 1 is not strictly needed but aims to simplify Step 2), so we can use Miller-Rabin test to discard composite numbers with a large probability. The test requires a single modular exponentiation (modulo number of size $\tilde{O}(\sqrt{p})$), 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 number tested are random 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 Step 1.

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

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

Step 2a is easy as most elements in $I$ will be suitable $\alpha$; in fact at least one basis element of $I$ will work.

(XXXX What is "large enough" in the below? How does this relate to the $\frac{7}{2}\log(p)$ issue? XXX)

In Step 2b the algorithm actually searches for $\beta_1 = a + bi + cj + dk$. 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). Heuristically it is sufficient to take $S_1 = \tilde{O}(p)$, and $O(\log p)$ approximately random numbers of this size must be tested for primality [27]. As shown above this has a cost of $\tilde{O}(\log^3 p)$ bit operations. Cornacchia's algorithm then requires $\tilde{O}(\log^2 p)$ bit operations to run.

Step 2c is just linear algebra modulo $N \approx \tilde{O}(\sqrt{p})$ and its cost can be neglected. As argued in [27] it has a negligible chance of failure, in which case one can just go back to Step 2b.

(XXXX Ditto "large enough" XXXXXX)

In Step 2d the algorithm a priori fixes $S_2$ large enough, then searches for integers $a, b, c, d, \lambda$ 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 $\beta_2 = Cj + Dk$. 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 $\lambda$. 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. Heuristically one can take $S_2 = \tilde{O}(p^3)$, and $O(\log p)$ approximately random numbers of size $\tilde{O}(p^2)$ will be tested for primality [27]. The overall cost of this step is therefore $\tilde{O}(\log^3 p)$ bit operations.

The costs of Step 2e and Step 3 can be neglected, leading to a cost of $\tilde{O}(\log^3 p)$ bit operations for the whole algorithm.

*Remark 3.* We stress that the output of this algorithm only depends on the ideal class of $I$ but not on $I$ itself. This is true since the algorithm first computes a Minkowski-reduced basis. It is important since in our use of the algorithm, the secret isogeny $\varphi$ could easily be recovered from $I$.

### 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 to be able to use 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 $(1, i, j, k)$ 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 [26, 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 ideal to 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 $n$. The main idea to determine the corresponding isogeny explicitly is to determine its kernel [37].

Let $\{\alpha_{j1}, \alpha_{j2}, \alpha_{j3}, \alpha_{j4}\}$ be a basis for the left $O_0$-ideal $I$ of norm $n$. Each element $\alpha_{jk} \in \langle 1, i, (1+k)/2, (i+j)/2\rangle$ can be written as $u + vi + wj + xk$ for some $u, v, w, x \in \mathbb{Q}$. We need to be able to determine, for a given point $P$, if $\alpha_{jk}(P) = 0$. The main issues is the denominators. Writing this as $\alpha = (u' + v'i + w'j + x'k)/z$ for some $u', v', w', x', z \in \mathbb{Z}$ we choose a point $P'$ such that $[z]P' = P$. Then $\alpha(P) = 0$ if and only if $[u']P' + [v']\phi(P') + [w']\pi(P') + [x']\pi(\phi(P')) = 0$.

To determine the complexity of such a computation it is necessary to bound the denominators, since the field of definition of the point $P'$ depends on this.

XXXX TO DO: How to bound the denominators???

Precisely, one can compute generators $P_i$, $i = 0, \ldots, n$ for all cyclic subgroups of $E_0[n]$, each one uniquely defining a degree $n$ isogeny which can be computed with Vélu's formulae.

The generator $P_i$ then corresponds to the basis $B_j$ if and only if $P_j$ is in the kernel of all corresponding basis maps. This algorithm requires at least $\tilde{O}(n^2 \log p)$ bit operations just to compute $E_0[n]$, hence its cost is prohibitive for large $n$.

When $n = \ell^e$ the degree $n$ isogeny can be decomposed into $e$ degree $\ell$ isogenies. If $I$ is the corresponding left $O_0$-ideal of norm $\ell^e$, then $I_i := I \bmod O_0\ell^i$ is a left $O_0$-ideal of norm $\ell^i$ corresponding to the first $i$ isogenies. Similarly if $P$ is a generator for the kernel of the degree $\ell^e$ isogeny then $\ell^{e-i+1}P$ is the kernel of the degree $\ell^i$ isogeny corresponding to the first $i$ steps. One can therefore perform the identification 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 $\ell^e$ torsion points, which in general may be defined over an $\ell^e$-extension of $\mathbb{F}_{p^2}$. To ensure that the $\ell^e$ 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 $\ell_i$, one reduces to the prime power case as follows. 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 indeed require that this is indeed the case in our protocols. Let $n_i := \prod_{j \le i} \ell_j^{e_j}$. 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 $\varphi_i$ which is a composition of all isogenies of degrees up to $\ell_i$. Using a Chinese 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 $I$, the following algorithm progressively identifies the corresponding isogeny sequence:

1. Let $\varphi_0$ be the identity endomorphism on $E_0$.
2. For each $i$:
   (a) Compute a generator $P_{ij}$ for each cyclic subgroup of order $\ell_i^{e_i}$ in $E_0[\ell_i^{e_i}]$.
   (b) Compute a basis $B_i = \{\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4}\}$ for $I \bmod O_0\ell_i^{e_i}$.
   (c) Find $j$ such that $P_{ij}$ is in the intersection kernel of the endomorphism maps corresponding to $\alpha_{ik}$.
   (d) Compute $\varphi_{i-1}(P_{ij})$.

(e) Compute the isogeny $\phi_i$ with kernel generated by $\varphi_{i-1}(P_{ij})$.
(f) Formally let $\varphi_i = \phi_i \circ \varphi_{i-1}$.

**Translating an isogeny path to an ideal** Let $E_0, E_1, \ldots, E_n$ an isogeny path and suppose $\varphi_i : E_0 \to E_i$ of degree $n_i$ (XX IS this right?? Is this what $n_i$ denotes???). The following algorithm progressively identifies the corresponding left-$O_0$ ideal $I$:

1. Let $I_0 = O_0$.
2. For each $i$:
   (a) Compute a generator $P_{ij}$ for each cyclic subgroup of order $\ell_i^{e_i}$ in $E_0[\ell_i^{e_i}]$.
   (b) Compute $\varphi_{i-1}(P_{ij})$ for all $j$.
   (c) Find $j$ such that $\varphi_{i-1}(P_{ij})$ is the kernel of the given isogeny of degree $\ell_i^{e_i}$.
   (d) Compute a basis $B_{ik} = \{\alpha_{ik1}, \alpha_{ik2}, \alpha_{ik3}, \alpha_{ik4}\}$ for each left $O_0$-ideal $I_{ik}$ of norm $n_i$ such that $I_{i-1} = I_{ik} \bmod O_0 n_{i-1}$.
   (e) Find $k$ such that the intersection kernel of the corresponding endomorphism maps contains $P_{ij}$, and let $I_i := I_{ik}$.

One can perform Step 2d of this algorithm as follows. Compute an embedding $\mu_\ell : B_{p,\infty} \hookrightarrow M_2(\mathbb{Q}_\ell)$ sending $O_0$ to $M_2(\mathbb{Z}_\ell)$. This essentially requires to compute a square root of $p$ in $\mathbb{Z}_\ell$. Any left ideal of $M_2(\mathbb{Z}_\ell)$ with norm $\ell^e$ is equal to $M_2(\mathbb{Z}_\ell)m_k$ with $m_k \in \left\{ \begin{pmatrix} \ell^{e-f} & r \\ 0 & \ell^f \end{pmatrix} : 0 \le f \le e\ ,\ 0 \le r < \ell^f \right\}$ [34, Theorem 2.2.3]. By computing $\mu_\ell^{-1}(m_k) \bmod O_0 \ell_i^{e_i}$ for $m_k$ a matrix as above, we obtain elements $\alpha_k \in B_{p,\infty}$, which can be used to compute $I_{ik} := I_{i-1}\alpha_k + I_{i-1}\ell_i^{e_i}$. Note that elements in $\mathbb{Z}_\ell$, including the embedding itself, need only be computed with precision $\ell^e$.

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. As we do not want to use special primes $\ell_i^{e_i}$ 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 $\ell_i^{e_i}$ can be evaluated using $O(\ell_i^{e_i})$ field operations. Each loop in the first algorithm above (from quaternion ideals to isogenies) therefore requires $O(\log^2 p)$ field operations hence $\tilde{O}(\log^4 p)$ bit operations, and the whole algorithm will require $\tilde{O}(\log^5 p)$ bit operations. A loop in the second algorithm above (from isogenies to quaternion ideals) requires $O(\log^3 p)$ field operations as such. This cost can be decreased to $O(\log^2 p)$ field operations if instead of applying isogenies to all generators in Step 2b, we only apply them on a basis of the $\ell_i^{e_i}$-torsion, then deduce the other values by linear combinations. Therefore the cost of both algorithms becomes $\tilde{O}(\log^5 p)$ bit operations.

### 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* Gen*:* On input a security parameter $\lambda$ generate a prime $p$ with $2\lambda$ bits, which is congruent to 3 modulo 4. Fix $B$, $S_1$, $S_2$ as small as possible[6] 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 $\prod \left(\frac{2\sqrt{\ell_{k,i}}}{\ell_{k,i}+1}\right)^{e_{k,i}} < \frac{6}{p}$. 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 $(p, j_0, j_1, H)$ and the secret key is $E_0, E_1, O_1, I)$.

*Signature Algorithm* Sign*:* On input a message $m$ and keys $(SK, PK)$, 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 the first algorithm 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 the second algorithm of Section 4.4 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 $\sigma = (h, z_1, \ldots, z_t)$.

*Verification Algorithm* Verify*:* On input a message $m$, a signature $\sigma$ 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.

*Efficiency:* (XXX BELOW NEEDS TO BE RE-WRITTEN. NOT DONE YET XXX)

As the best 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 $B \approx \log S_i \approx 2 \log p$ ensures that the outputs of random walks are distributed uniformly enough. Random walks then require $2 \log p$ bits to represent, so signatures are $2 \log p \cdot \lambda + 2\lambda + \lambda \frac{1}{2}(2 \log p + \frac{7}{2} \log p) \approx \frac{23}{2}\lambda^2$ bits on average, depending on the challenge bits, private keys are $2\lambda$ bits and public keys are $3 \log p = 6\lambda$ bits. A signature mostly requires $2\lambda$ calls to the Algorithm of Sections 4.3 and 4.4, for a total cost of $\tilde{O}(\lambda^6)$. 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].

[6] The exact procedure is irrelevant here.

## 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 $\ell$-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.

Figures 1 and 2 compare the performance of our schemes with RSA and ECDSA signatures (assuming asymptotically optimal arithmetic), two dominant signature schemes in current cryptography architectures. Our signature sizes use the optimisation of Remark 1, but not the optimisation of Remark 2. Asymptotically, both our signature schemes compare favorably to RSA, except in verification costs for our second signature scheme and when a small public key exponent is used in RSA. At 128 and 256-bit security levels private keys are the same size as ECDSA private keys, and public key sizes are between ECC and RSA public keys. Signature sizes are somewhat large, even though asymptotically they will be smaller than RSA signatures. We find these first estimations encouraging and hope that future work will reduce them further.

| | Private Key Size | Public Key Size | Signature Size | Signing Costs | Verification Costs |
|---|---|---|---|---|---|
| RSA | $O(\lambda^3)$ | $O(\lambda^3)$ | $O(\lambda^3)$ | $\tilde{O}(\lambda^6)$ | $\tilde{O}(\lambda^3)$ |
| ECDSA | $2\lambda$ | $2\lambda$ | $4\lambda$ | $\tilde{O}(\lambda^2)$ | $\tilde{O}(\lambda^2)$ |
| Section 3 | $2\lambda$ | $20\lambda$ | $6\lambda^2$ | $\tilde{O}(\lambda^3)$ | $\tilde{O}(\lambda^3)$ |
| Section 4 | $2\lambda$ | $6\lambda$ | $11\lambda^2$ | $\tilde{O}(\lambda^6)$ | $\tilde{O}(\lambda^5)$ |

**Fig. 1.** Asymptotic efficiency comparison between our signature schemes, RSA and ECDSA, as a function of the security parameter $\lambda$. All sizes are in bits and computation costs are in bit operations. For RSA signatures we assume that small public key exponents are used, and for ECDSA we assume that a standardized curve is used and points are represented by their $x$-coordinates.

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.

| | 128 bit | | | 256 bit | | |
|---|---|---|---|---|---|---|
| | Private Key | Public Key | Signature | Private Key | Public Key | Signature |
| RSA | 3248 | 3248 | 3248 | 15424 | 15424 | 15424 |
| ECDSA | 256 | 256 | 256 | 512 | 512 | 512 |
| Section 3 | 256 | 2560 | 311296 | 512 | 5120 | 1245184 |
| Section 4 | 256 | 768 | 188416 | 512 | 3072 | 753664 |

**Fig. 2.** Concrete efficiency comparison between our signature schemes, RSA and ECDSA, at security levels of 128 and 256 bits. For RSA we used ECRYPT II's key length recommendations as computed by `www.keylength.com`. All sizes are in bits.

## 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. arXiv:math/0610550, October 2006.
3. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 474–483, 2014.
4. 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.
5. Mihir Bellare, Bertram Poettering, and Douglas Stebila. From Identification to Signatures, Tightly: A Framework and Generic Transforms. Cryptology ePrint Archive: Report 2015/1157, 2016.
6. 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.
7. 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.
8. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. *IACR Cryptology ePrint Archive*, 2010:428, 2010.
9. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 361–379, 2013.
10. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem. In *Encyclopedia of Algorithms*, pages 1662–1664. 2016.

11. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
12. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptology*, 22(1):93–113, 2009.
13. 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.
14. Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. The fiat-shamir transformation in a quantum world. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 62–81, 2013.
15. Max Deuring. Die typen der multiplikatorenringe elliptischer funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941. 10.1007/BF02940746.
16. 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.
17. 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.
18. Steven D. Galbraith. Constructing Isogenies Between Elliptic Curves Over Finite Fields. *LMS J. Comput. Math*, 2:118–138, 1999.
19. Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. Cryptology ePrint Archive, Report 2016/859, 2016. `http://eprint.iacr.org/2016/859`.
20. Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.
21. Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.
22. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, pages 19–34, 2011.
23. 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.
24. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. *CoRR*, abs/1602.05973, 2016.
25. Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.
26. David Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.
27. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion $\ell$-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17A:418–432, 2014.
28. Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Hash function requirements for schnorr signatures. *J. Mathematical Cryptology*, 3(1):69–87, 2009.
29. Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Transactions on Algorithms*, 5(4), 2009.
30. Christophe Petit. On the quaternion $\ell$-isogeny problem. Presentation slides from a talk at the University of Neuchâtel, March 2015.
31. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Springer Verlag, 1986.

32. 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.
33. Jacques Vélu. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Série A.*, 273:238–241, 1971.
34. Marie-France Vignéras. *The arithmetic of quaternion Algebra*. 2006.
35. Joachim von zur Gathen and Victor Shoup. Computing frobenius maps and factoring polynomials. *Computational Complexity*, 2:187–224, 1992.
36. Jacques Vélu. Isogénies entre courbes elliptiques. *Communications de l'Académie royale des Sciences de Paris*, 273:238–241, 1971.
37. William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'E.N.S.*, 2:521–560, 1969.
38. 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 $c = 2$, obtaining $\frac{1}{2}B = \log p$.