SÉTA: Supersingular Encryption from Torsion Attacks

Cyprien Delpech de Saint Guilhem[1,2], Péter Kutas[3], Christophe Petit[3], and Javier Silva[4]

[1] imec-COSIC, KU Leuven, Belgium
[2] Dept Computer Science, University of Bristol, United Kingdom
[3] School of Computer Science, University of Birmingham, United Kingdom
[4] Universitat Pompeu Fabra, Barcelona, Spain cyprien.delpechdesaintguilhem@kuleuven.be, p.kutas@bham.ac.uk, christophe.f.petit@gmail.com, javier.silva@upf.edu

Abstract. We present *SÉTA*,[5] a new family of public-key encryption schemes with post-quantum security based on isogenies of supersingular elliptic curves. We first define a family of trapdoor one-way functions for which the computation of the inverse is based on an attack by Petit (ASIACRYPT 2017) on the problem of computing an isogeny between two supersingular elliptic curves, given the images of torsion points by this isogeny. We use this method as a decryption mechanism to build first a OW-CPA scheme, then we make use of generic transformations to obtain IND-CCA security in the quantum random oracle model, both for a PKE scheme and a KEM. Compared to alternative schemes based on SIDH, our protocols have the advantage of relying on arguably harder problems.

1 Introduction

Isogeny-based cryptography. In recent years, there has been an increasing interest in cryptosystems whose security rely on supersingular isogeny problems, since these are believed to be hard to solve even with quantum computers. This makes them appropriate candidates for post-quantum cryptography, a field which has received increasing attention because of the recent standardization process initiated by NIST.[6]

More precisely, the central problem considered in isogeny-based cryptography is, given two elliptic curves, to compute an isogeny between them. For the right choice of parameters, the best quantum algorithms for solving this problem still run in exponential time [5]. Variants of this problem have been used to build cryptographic primitives such as hash functions [8], encryption schemes [13,2], key encapsulation mechanism (KEM)s [2] and signatures [19,36].

Encryption schemes. In 2011, Jao and De Feo [21] introduced the first public-key encryption (PKE) scheme based on isogenies of supersingular elliptic curves, built from a key agreement protocol. Their work was inspired by a previous construction by Stolbunov [30] for ordinary elliptic curves, with a switch to supersingular curves to thwart sub-exponential quantum algorithms that exist in the ordinary case [9].

The key agreement protocol follows a "Diffie–Hellman-like" structure: Alice and Bob start from a public curve E_0 and choose random secret isogenies φ_A, φ_B to reach curves E_A, E_B. Then they send the curves to each other and finally use their respective secrets to arrive at a common curve E_{AB} as shown in Figure 1. While the commutativity of the diagram is preserved as long as the two isogenies have co-prime degrees, in the supersingular case it is not immediately possible for, say, Bob to evaluate his isogeny φ_B on Alice's curve E_A without some extra help. To solve this, Jao and De Feo proposed sending additional information in the protocol in the form of images of torsion points under the secret isogenies. With the help of these points, they ensured that each party could evaluate their secret isogeny on the other's curve.

[5] To be pronounced [ʃeːtɒ] meaning "walk" in Hungarian.

[6] U.S. Department of Commerce, National Institute of Standards and Technology, Post-Quantum Cryptography project, 2016. Available at https://csrc.nist.gov/projects/post-quantum-cryptography, last retrieved September 13th, 2019.

$$\begin{tikzcd}
 & E_A \arrow[dr, "\tilde{\varphi}_B"] & \\
E_0 \arrow[ur, "\varphi_A"] \arrow[dr, "\varphi_B"] & & E_{AB} \\
 & E_B \arrow[ur, "\tilde{\varphi}_A"] &
\end{tikzcd}$$

Fig. 1. Sketch of the SIDH key agreement protocol.

However, the isogeny problem upon which the security of the scheme is based now differs from the original problem in several ways. First, it is a decisional problem, consisting on distinguishing E_{AB} from random, given E_0, E_A, E_B. This is analogous to the relation between the discrete logarithm and decisional Diffie–Hellman problems. Second, the adversary now has access to the images of some torsion points under the secret isogenies, which in principle could make the recovery of these isogenies easier. In addition, Jao and De Feo propose to use special primes and rather small degree isogenies in their protocols to accelerate computations.

The introduction of this new hardness assumption fostered the proposal of new related assumptions that were used to prove the security of isogeny-based schemes. Many of these assumptions shared the need to reveal extra points. In particular, this family of assumptions and parameter choices are used in the SIKE submission to the NIST process [2].

A worrisome attack on SIKE variants. In 2017, Petit [24] studied the impact of the extra points in the hardness of these problems. He showed that for some choices of parameters, the problem could in fact be solved in polynomial time with classical algorithms. More precisely, Petit's algorithm solves the following problem: let E_0 be a special curve, for which the endomorphism ring is known, and let $\varphi : E_0 \to E$ be an isogeny of degree D. Let P, Q be a basis of the N-torsion of E_0. Then, given $E_0, E, \varphi(P), \varphi(Q)$, the problem is to compute φ. The algorithm's running time depends on the choices of D and N.

So far, Petit's techniques cannot be applied to the parameters proposed by Jao and De Feo, hence the proposed schemes [21,13,2] remain secure. Nevertheless and in anticipation of potential further cryptanalysis progress, it is desirable to design alternative cryptographic protocols that only rely on "pure" isogeny problems, where in particular no additional torsion point action is revealed to an attacker. This has so far only been achieved for signature schemes [29,19,12] and hash functions [8]. A special case is CSIDH [7], a key agreement protocol that relies on the pure isogeny problem, but is restricted to supersingular elliptic curves over $\mathbb{F}_p$, and can be broken in quantum subexponential time.

More generally, any relaxation of the assumptions used in building isogeny-based PKE schemes and KEMs is of interest from a theoretical point of view, and could become crucial if further cryptanalysis progress occurs.

Our contributions. We provide new PKE schemes and KEMs based on isogeny problems. Key recovery security for our schemes only relies on the "pure" isogeny problem for supersingular curves, and the standard OW-CPA and IND-CCA security rely on problems that are arguably harder to solve than those used in SIDH and SIKE.

We now briefly sketch the core idea of our constructions. Petit's algorithm crucially uses the fact that the endomorphism ring of E_0 is known in SIDH/SIKE. We exploit this fact to turn the attack into a decryption mechanism.

– Let E_0 be a special curve as above. Alice takes a random isogeny $\varphi_s : E_0 \to E_s$ and publishes E_s as her public key, keeping φ_s as her secret key. A canonical method to compute a basis P, Q of the N-torsion of any E_s is also fixed as part of the scheme.

– When Bob wants to send a message m to Alice, he encodes it into an isogeny $\varphi_m : E_s \to E_m$, creating the following diagram.

$$E_0 \xrightarrow{\varphi_s} E_s \xrightarrow{\varphi_m} E_m$$

He sends $(j(E_m), \varphi_s(P), \varphi_s(Q))$ as the ciphertext.
– To decrypt a message, Alice uses her secret isogeny φ_s and knowledge of the endomorphism ring of E_0 to compute endomorphisms of E_s. She can then recover the secret φ_m by running the attack on the ciphertext.

Note that the endomorphism ring of E_s remains hidden to the adversary, so, even though the parameters are chosen to enable Petit's attack, it cannot be run unless $\mathrm{End}(E_s)$ is recovered. The task of recovering the endomorphism ring of a randomly sampled supersingular curve is also a hard problem, for which only exponential-time algorithms exist. The problem is in fact heuristically equivalent to computing isogenies between two randomly sampled supersingular curves [25,15]. As a consequence, an alternative secret key cannot be derived when given only E_0 and E_s.

Outline We first recall the required background on isogenies and quaternion algebras in Section 2. We also state relevant computational problems and formal security definitions that we will use, and briefly recall the SIDH/SIKE constructions. We then present a generalisation of the Charles-Goren-Lauter hash function and describe our construction as a trapdoor one-way function (OWF), together with its inversion mechanism, in Section 3. We also deal with a potential timing dependency that arises from an uncommon case in which Petit's algorithm takes longer to recover the isogeny. We identify when this happens and tune our parameters to avoid this case completely. In Section 4, we present the two SÉTA PKE schemes; one is OW-CPA secure and we make use of a generic OAEP-style transformation to achieve IND-CCA security in the quantum random oracle model for the other. In Section 5 we make use of other transformations to obtain IND-CCA secure KEMs. There, we present two alternative routes: one uses the transformations of [20], which works out of the box but have the disadvantage of the security reduction not being tight; the other uses the work of [26], which has tighter reductions but requires the starting scheme to verify an additional property called sparse pseudorandomness. We prove this property under the hardness of another isogeny problem. In Section 6, we collect all the parameter restrictions and show how to choose appropriate parameters. We conclude the paper in Section 7.

2 Preliminaries

We denote the security parameter by λ. We write PPT for probabilistic polynomial time. The notation $y \leftarrow \mathcal{A}(x; r)$ means that the algorithm $\mathcal{A}$, with input x and randomness r, outputs y. An algorithm $\mathcal{A}$ with oracle access to a function O is represented as $\mathcal{A}^{\mathsf{O}(\cdot)}$. The notation Pr[sampling : event] means the probability of the event on the right happening after sampling elements as specified on the left. Given a set $\mathcal{S}$, we denote sampling a uniformly random element x of $\mathcal{S}$ by $x \xleftarrow{\$} \mathcal{S}$. We denote the cardinality of $\mathcal{S}$ by $\#\mathcal{S}$. A probability distribution X has min-entropy $H_\infty(X) = b$ if any event occurs with probability at most 2^{-b}. For $n \in \mathbb{N}$, we use the notation $[n] = \{0, \dots, n-1\}$ when the context clearly indicates that this is a set. Given an integer $n = \prod_i \ell_i^{e_i}$, where the ℓ_i are its prime factors, we say that n is *B-powersmooth* if $\ell_i^{e_i} < B$ for all i. We denote by $\mathbb{Z}_n$ the set of residue classes modulo n. Throughout this paper, we let $p > 3$ denote a prime number.

2.1 Supersingular elliptic curves

We first recall definitions and results concerning supersingular elliptic curves.

Let q be a power of $p$and let E_1, E_2 be elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny $\varphi : E_1 \to E_2$ is a surjective morphism which sends the point of infinity of E_1 to the point of infinity of E_2. An isogeny is also a group homomorphism from $E_1(\overline{\mathbb{F}_q})$ to $E_2(\overline{\mathbb{F}_q})$ with a finite kernel. The degree of the isogeny is its degree as a finite map of curves. If the isogeny φ is separable, then $\#\ker\varphi = \deg\varphi$. If there exists an isogeny

φ from E_1 to E_2, then there exists a unique isogeny $\hat{\varphi}$ from E_2 to E_1 with the property that $\varphi \circ \hat{\varphi} = [n]$ where n is the degree of the isogeny and $[n]$ denotes here the multiplication by n map on E_2. Such isogenies φ and $\hat{\varphi}$ are called dual of each other. We call two curves isogenous if there exists an isogeny between them. By the previous remark, this relation is symmetric.

Let E be an elliptic curve defined over $\mathbb{F}_q$. An isogeny from E to itself is called an endomorphism of E. Under addition and composition, endomorphisms of E form a ring denoted $\mathrm{End}(E)$. A theorem of Deuring states that such an endomorphism ring is either an order in an imaginary quadratic field (such curves are called ordinary) or a maximal order in a quaternion algebra (such curves are called supersingular).

It is a well-known theorem of Tate that two curves defined over $\mathbb{F}_q$ are isogenous by an isogeny defined over $\mathbb{F}_q$ if and only if their number of $\mathbb{F}_q$-rational points is equal. Supersingular curves can always be defined (up to isomorphism) over $\mathbb{F}_{p^2}$ and a curve is supersingular if and only if the number of points is congruent to 1 mod p. Supersingularity is thus preserved under isogenies.

Kernels of isogenies and Vélu's formulas. An isogeny is a group homomorphism whose kernel is a finite subgroup of the starting curve. Moreover, let E be an elliptic curve defined over finite field $\mathbb{F}_q$ and let G be a finite subgroup of $E(\overline{\mathbb{F}_q})$. Then there exists a unique (up to automorphisms of the target curve) separable isogeny whose kernel is exactly G. Due to this uniqueness property we will denote the image curve by E/G. Furthermore, given a subgroup G whose order is powersmooth, the curve E/G can be computed efficiently using Vélu's formulas [35].

Elliptic curve j-invariant. An elliptic curve E defined over $\mathbb{F}_{p^2}$ can always be written in short Weierstrass form $E : y^2 = x^3 + Ax + B$, for $A, B \in \mathbb{F}_{p^2}$. We can therefore identify any curve with its two coefficients: $E \sim (A, B)$. Given such a curve, its j-invariant is defined as $j(E) = 1728\frac{4A^3}{4A^3+27B^2}$. As its name suggests, this quantity is invariant under any isomorphism over $\overline{\mathbb{F}_{p^2}}$. In this work, we denote by $\mathcal{J}_p$ the set of j-invariants of supersingular curves defined over $\mathbb{F}_{p^2}$. We then identify the set of isomorphism classes of supersingular elliptic curves over $\mathbb{F}_{p^2}$ with $\mathcal{J}_p$.

The Weil pairing. For $N \in \mathbb{N}$, let $E[N] := \{P \in E(\overline{\mathbb{F}_{p^2}}) : [N]P = \infty\}$ denote the N-torsion of E, where ∞ denotes the point at infinity in $E(\overline{\mathbb{F}_{p^2}})$. The Weil pairing is a map $e : E[N] \times E[N] \to \mathbb{F}_{p^2}$ that is both bilinear and non-degenerate:

$$\begin{gathered} e(P_1 + P_2, Q) = e(P_1, Q) \cdot e(P_2, Q) \\ e(P, Q_1 + Q_2) = e(P, Q_1) \cdot e(P, Q_2) \\ \forall P \in E[N] \setminus \{\infty\},\ \exists Q \in E[N] : e(P, Q) \neq 1. \end{gathered}$$

Canonical curves. We take the same approach as [19, Appendix A] to fix canonical choices of curves. Given $j \in \mathbb{F}_{p^2}$, we define the curve E_j as $E_j \sim (0, 1)$ when $j = 0$, $E_j \sim (1, 0)$ when $j = 1728$ and $E_j \sim (\frac{3j}{1728-j}, \frac{2j}{1728-j})$ otherwise.

Isogeny graphs. Let $\ell \neq p$ be a prime number. Define the graph $G_\ell = G_\ell(\mathbb{F}_{p^2})$ to have vertex set $V = \mathcal{J}_p$. We have that $\#V = \lfloor \frac{p}{12} \rfloor + k$, where $k \in \{0, 1, 2\}$. Given two vertices $j_1, j_2 \in V$, with representative curves E_1, E_2 such that $j(E_i) = j_i$, there is an edge in G_ℓ between j_1 and j_2 if and only if there is an equivalence class of ℓ-isogenies between E_1 and E_2, where two isogenies $\varphi, \psi : E_1 \to E_2$ are equivalent if there exists an automorphism α of E_2 such that $\psi = \alpha\varphi$.

Edges of $G_\ell(\mathbb{F}_{p^2})$ can also be defined by the modular polynomial $\Phi_\ell(x, y) \in \mathbb{Z}[x, y]$ [27]. It is symmetric, meaning that $\Phi_\ell(x, y) = \Phi_\ell(y, x)$, and is of degree $\ell + 1$ in both x and y. It holds that $\Phi_\ell(j_1, j_2) = 0$ if and only if there is an ℓ-isogeny equivalence class between two curves with j-invariants j_1 and j_2, and thus an edge in G_ℓ. Therefore, given a vertex $j \in V$, its neighbours are exactly those j-invariants which are roots of the univariate polynomial $\Phi_\ell(x, j)$. As Φ_ℓ is of degree $\ell + 1$ in x and all the j-invariants are in $\mathbb{F}_{p^2}$, we see that G_ℓ is an $(\ell + 1)$-regular graph.

2.2 Quaternion algebras and endomorphism rings of supersingular elliptic curves

A quaternion algebra is a four-dimensional central simple algebra over a field K. When the characteristic of K is not 2, then A admits a basis $1, i, j, ij$ such that $i^2 = a,\ j^2 = b, ij = -ji$ where $a, b \in K \setminus \{0\}$. The numbers a, b characterise the quaternion algebra up to isomorphism, thus we denote the aforementioned algebra by the pair (a, b). A quaternion algebra is either a division ring or it is isomorphic to $M_2(K)$, the algebra of 2×2 matrices over K.

Let A be a quaternion algebra over $\mathbb{Q}$. Then $A \otimes \mathbb{Q}_p$ is a quaternion algebra over $\mathbb{Q}_p$ (the field of p-adic numbers) and $A \otimes \mathbb{R}$ is a quaternion algebra over the real numbers. A is said to split at p (resp. at ∞) if $A \otimes \mathbb{Q}_p$ (resp. $A \otimes \mathbb{R}$) is a full matrix algebra. Otherwise it is said to ramify at p (resp. at ∞). A quaternion algebra over $\mathbb{Q}$ is split at every but finitely many places, and the list of these places defines the quaternion algebra up to isomorphism. An order in a quaternion algebra over $\mathbb{Q}$ is a four-dimensional $\mathbb{Z}$-lattice which is also a subring containing the identity (it is the non-commutative generalization of the ring of integers in number fields). An order is called maximal if it is maximal with respect to inclusion.

The endomorphism ring of a supersingular elliptic curve over $\mathbb{F}_{p^2}$ is a maximal order in the quaternion algebra $B_{p,\infty}$, which ramifies at p and at ∞. Moreover, for every maximal order in $B_{p,\infty}$ there exists a supersingular elliptic curve whose endomorphism ring is isomorphic to it.

It is easy to see that, when $p \equiv 3 \pmod 4$, this quaternion algebra is isomorphic to the quaternion algebra $(-p, -1)$. In that case, the integral linear combinations of $1, i, \frac{ij+j}{2}, \frac{1+i}{2}$ form a maximal order $\mathcal{O}_0$ which corresponds to an isomorphism class of supersingular curves, namely the class of curves with j-invariant 1728 (e.g. the curve $E : y^2 = x^3 + x$). It is easy to see that all elements $ai + bj + cij + d$ with $a, b, c, d \in \mathbb{Z}$ are contained in $\mathcal{O}_0$.

2.3 Computational problems for supersingular isogenies

Given an elliptic curve, a first problem is to compute its endomorphism ring.

Problem 1 (Endomorphism ring computation). 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.

Next, given two elliptic curves over $\mathbb{F}_q$, another problem is whether there exists an isogeny between them and, if it does, how to compute it. Existence can be decided in polynomial time by computing the number of points on the respective curves. When there exists a low-degree isogeny between them, this isogeny can be guessed and computed easily. The interesting problem is therefore to compute an isogeny between two curves when no such isogeny with low degree exists. Moreover, one typically restricts to the case of smooth degree isogenies, as the output isogeny can then be naturally represented as a composition of low degree rational maps.

Problem 2. Let d be a smooth number. Let E_1 and E_2 be elliptic curves over $\mathbb{F}_q$ connected by an isogeny of degree d. Compute an isogeny between E_1 and E_2.

Note that, heuristically at least, the restriction to smooth degree isogenies does not change the hardness of Problem 2, which is equivalent to Problem 1 [25,15]. Furthermore, fixing E_1 arbitrarily and letting E_2 vary does not change the complexity. In our protocols, security against key recovery attacks will rely on the hardness of these problems only, unlike in SIDH and SIKE. The OW-CPA security of our protocols also relies on the following related problem, in which images of torsion points by a degree d isogeny are revealed.

Problem 3 (Random-start computational supersingular isogeny (RCSSI) problem). Given p and integers d and N, let E_1 be a uniformly random supersingular elliptic curve over $\mathbb{F}_{p^2}$ and $\varphi : E_1 \to E_2$ be a random isogeny of degree d sampled from a distribution X with min-entropy $H_\infty(X) = O(\lambda)$. Let P, Q be a basis of the torsion group $E_1[N]$. Given $E_1, P, Q, E_2, \varphi(P)$ and $\varphi(Q)$, compute φ.

We stress that Problem 3 is a variant of the CSSI problem, introduced in [13, Problem 5.2], which differs in two aspects. The first difference is that the starting curve E_1 is uniformly random instead of being a special fixed curve. When E_1 is a special curve for which the endomorphism ring is known, there are parameters d and N for which Problem 3 may be easy, as shown in [24]. However, selecting E_1 at random means that computing its endomorphism ring is exactly an instance of Problem 1. Since that problem is also believed to be hard on average, the attack of Petit [24] does not apply against Problem 3, even for unbalanced d and N.

The second difference is the specification of the entropy of the distribution from which the challenge isogeny is sampled. Note that in the statement of Problem 3 we have allowed arbitrary distributions with sufficient min-entropy for convenience, but in fact we will only require the problem to be hard for specific distributions. In Appendix A, we explain that this modification to the original CSSI problem is in fact not specific to our protocols, as a similar modification seems to be needed to formally prove the security of the NIST submission SIKE [2] derived from SIDH.

The above problems are the only computational problems needed to construct new PKE and KEM schemes based on our new trapdoor OWF. However, the reduction is not tight for the KEM, and a tighter reduction can be obtained by relying on an additional problem.

In [32, Definitions 2 and 3], the authors consider the problem of, given two curves E_1, E_2 such that there exists an isogeny φ between them, and a basis $\{P, Q\}$ of the N-torsion of E_1, computing $\varphi(P), \varphi(Q)$. We consider the decisional variant of this problem. However, we cannot expect indistinguishability between images of torsion points and random points of the N-torsion of E_2, as there are some properties that the former will always satisfy. We therefore impose these on the latter.

Problem 4. Let E_1 be a random supersingular elliptic curve, and let P, Q be a basis of $E[N]$. Let $E_2 = E_1/\ker\varphi$ for some random d-isogeny φ from E_1, sampled from a distribution X with min entropy $H_\infty(X) = O(\lambda)$, and assume that $gcd(N, d) = 1$. Consider the following distributions:

- $(\overline{P}, \overline{Q})$, where $\overline{P} = \varphi(P), \overline{Q} = \varphi(Q)$.
- $(\overline{P}, \overline{Q})$, where $\overline{P}, \overline{Q} \xleftarrow{\$} E_2[N]$, conditioned on $e(\overline{P}, \overline{Q}) = e(P, Q)^{\deg\varphi}$.

The problem is, given $E_1, E_2, P, Q, \overline{P}, \overline{Q}$, to distinguish between these two distributions.

Remark 1. We note that sampling elements of the second distribution is efficient, as it essentially amounts to choosing a matrix in $GL_2(\mathbb{Z}_N)$ with the correct determinant. See Appendix B for a more detailed analysis.

2.4 SIDH and SIKE protocols

Here we give a high level description of SIDH and SIKE. We start with the original SIDH protocol of Jao and De Feo [21]. In the setup one chooses two small primes ℓ_A and ℓ_B and a prime p which is of the form $p = \ell_A^{e_A}\ell_B^{e_B}f - 1$ where f is a small cofactor and e_A and e_B are large (in SIKE [2] they use $\ell_A^{e_A} = 2^{216}$, $\ell_B^{e_B} = 3^{137}$ and $f = 1$). Let E be the elliptic curve with j-invariant 1728. Let P_A, Q_A be a basis of $E[\ell_A^{e_A}]$ and let P_B, Q_B be a basis of $E[\ell_B^{e_B}]$. The protocol is as follows:

1. Alice chooses a random cyclic subgroup of $E[\ell_A^{e_A}]$ generated by $A = [x_A]P_A + [y_A]Q_A$ and Bob chooses a random cyclic subgroup of $E[\ell_B^{e_B}]$ generated by $B = [x_B]P_B + [y_B]Q_B$.
2. Alice computes the isogeny $\varphi_A : E \to E/\langle A\rangle$ and Bob computes the isogeny $\varphi_B : E \to E/\langle B\rangle$
3. Alice sends the curve $E/\langle A\rangle$ and the points $\varphi_A(P_B)$ and $\varphi_A(Q_B)$ to Bob and Bob similarly sends $(E/\langle B\rangle$, $\varphi_B(P_A)$, $\varphi_B(Q_A))$ to Alice
4. Alice and Bob both use the images of the torsion points to compute the shared secret which is the curve $E/\langle A, B\rangle$ (e.g. Alice can compute $\varphi_B(A) = [x_A]\varphi_B(P_A) + [y_A]\varphi_B(Q_A)$ and $E/\langle A, B\rangle = E_B/\langle\varphi_B(A)\rangle$).

This key exchange protocol also leads to a PKE scheme in the same way as the Diffie–Hellman key exchange leads to ElGamal encryption. Let Alice's private key be the isogeny $\varphi_A : E \to E/\langle A\rangle$ and her public key be the curve $E/\langle A\rangle$ together with the images of the torsion points $\varphi_A(P_B)$ and $\varphi_A(Q_B)$. Encryption and decryption work as follows:

1. To encrypt a bitstring m, Bob chooses a random subgroup generated by $B = [x_B]P_B + [y_B]Q_B$ and computes the corresponding isogeny $\varphi_B : E \to E/\langle B\rangle$. He computes the shared secret $E \to E/\langle A, B\rangle$ and hashes the j-invariant of $E/\langle A, B\rangle$ to a binary string s. The ciphertext corresponding to m is the tuple $(E/\langle B\rangle, \varphi_B(P_A), \varphi_B(Q_A), c := m \oplus s)$
2. In order to decrypt Bob's message, Alice computes $E/\langle A, B\rangle$ and from this information computes s. Then she retrieves the message by computing $c \oplus s$.

This PKE scheme is IND-CPA secure [21,13,2]. In the SIKE submission [2], it is transformed using the constructions in [20, Section 3] to produce an IND-CCA secure KEM in the random oracle model.

2.5 Security definitions

We now recall standard security definitions for PKE schemes and KEMs. We note that in the definitions in this section, the adversary has quantum access to the random oracles, but only classical access to any other oracles.

We will first recall some weak security notions that are used as a starting point in the generic transformations discussed in later sections. The first one is the notion of partial-domain one-wayness, which states that some part of the input of a function f is hard to recover given the corresponding output.

Definition 1 ([33], Definition 6). *Let $f : \{0,1\}^{\lambda+k_1} \times \{0,1\}^{k_0} \to \{0,1\}^m$ be a function. We say that f is* partial-domain one-way *if, for any quantum PPT adversary $\mathcal{A}$,*

$$\Pr\left[s \xleftarrow{\$} \{0,1\}^{\lambda+k_1}, t \xleftarrow{\$} \{0,1\}^{k_0}, \tilde{s} \leftarrow \mathcal{A}(f(s,t)) : \tilde{s} = s\right] \leq \mathsf{negl}(\lambda)$$

We also consider OW-CPA security for PKE schemes, meaning that it should be hard to decrypt a ciphertext of a random message. Note that this is weaker than the usual IND-CPA security.

Definition 2 ([20], Definition 1). *Let $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption sche-me with message space $\mathcal{M}$. We say that the encryption scheme is secure against quantum* one-way chosen-plaintext attack (OW-CPA) *if, for any quantum PPT adversary $\mathcal{A}$,*

$$\Pr\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda), m^* \xleftarrow{\$} \mathcal{M}, \\ c^* \leftarrow \mathsf{Enc}_{pk}(m^*), \tilde{m} \leftarrow \mathcal{A}(pk, c^*)\end{matrix} : \tilde{m} = \mathsf{Dec}_{sk}(c^*)\right] \leq \mathsf{negl}(\lambda).$$

We include below a definition for security against key recovery. This is usually not defined separately from OW-CPA security as the latter implies the former. We include it here to later highlight that our encryption schemes enjoy additional protection guarantees against key recovery attacks.

Definition 3 (Security against key recovery). *Let $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme with key space $\mathcal{K}$ and message space $\mathcal{M}$. We say that the encryption scheme is secure against key recovery if, for any quantum PPT adversary $\mathcal{A}$,*

$$\Pr\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda) \\ sk' \leftarrow \mathcal{A}(pk)\end{matrix} : \forall m \in \mathcal{M}, \mathsf{Dec}_{sk'}\left(\mathsf{Enc}_{pk}(m)\right) = m\right] \leq \mathsf{negl}(\lambda).$$

Starting from primitives satisfying the definitions above, generic transformations can produce PKE schemes and KEMs, respectively, with very strong levels of security. More precisely, we will obtain IND-CCA security in the quantum random oracle model in both cases.

Definition 4 ([33], Definition 5). *Let $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption sche-me. We say that the encryption scheme is secure against quantum* indistinguishable chosen-ciphertext attack (IND-CCA) *if, for any quantum PPT adversary $\mathcal{A}$,*

$$\left|\Pr\left[\begin{matrix}(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda), m_0, m_1 \leftarrow \mathcal{A}^{\mathsf{O}(\cdot)}(pk), \\ b \xleftarrow{\$} \{0,1\}, c^* \leftarrow \mathsf{Enc}_{pk}(m_b), \tilde{b} \leftarrow \mathcal{A}^{\mathsf{O}(\cdot)}(pk, c^*),\end{matrix} : \tilde{b} = b\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{O}(c)$ returns $\mathsf{Dec}_{sk}(c)$ for $c \neq c^*$.

Definition 5 ([20], Definition 3). *Let* (KGen, Enc, Dec) *be a KEM with symmetric key space* $\mathcal{K}$. *We say that the KEM is secure against quantum* IND-CCA *if, for any quantum PPT adversary* $\mathcal{A}$*,*

$$\left| \Pr\left[\begin{matrix} (pk, sk) \leftarrow \mathsf{KGen}(1^\lambda), b \xleftarrow{\$} \{0,1\}, \\ (K_0^*, c^*) \leftarrow \mathsf{Enc}(pk), K_1^* \xleftarrow{\$} \mathcal{K}, \tilde{b} \leftarrow \mathcal{A}^{\mathsf{O}(\cdot)}(K_b^*, c^*) \end{matrix} : \tilde{b} = b \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{O}(c)$ *returns* $\mathsf{Dec}_{sk}(c)$ *for* $c \neq c^*$.

3 Injective trapdoor OWFs from supersingular isogenies

We first present a generalisation of the CGL hash function [8] and then introduce a family of trapdoor OWFs constructed from it. We show that, for certain parameters, we can efficiently sample a statistically uniform function from the family and that any such function is injective and one-way. Finally, we show that sampling a function at random yields a trapdoor, i.e. a secret isogeny, which we can use to efficiently invert the function.

3.1 Charles-Goren-Lauter hash function

We now present the CGL hash function family $\mathcal{H}^{p,\ell,n} = \{h_j^{\ell,n} : [\ell]^n \to \mathcal{J}_p\}_{j \in \mathcal{J}_p}$ as introduced in [8] (we omit ℓ, n from notation when the context is clear). To select a hash function from the family, one selects a j-invariant $j \in \mathcal{J}_p$ which fixes a canonical curve $E/\mathbb{F}_{p^2}$ with $j(E) = j$. The graph $G_\ell(\mathbb{F}_{p^2})$ is $(\ell+1)$-regular so there are $\ell+1$ isogenies of degree ℓ connecting E to other vertices; a canonical one of these is ignored once and for all and the other ℓ are numbered arbitrarily. Then, given an n-symbol message $m = b_1 b_2 \ldots b_n$, with $b_i \in [\ell]$, hashing starts by choosing a degree-ℓ isogeny from E according to symbol b_1 to arrive at a first curve E_1. Not allowing backtracking, there are then only ℓ isogenies out of E_1 and one is chosen according to b_2 to arrive at a second curve E_2. Continuing in the same way, m determines a unique walk of length n along the edges of G_ℓ.

The output of the CGL hash function is then the j-invariant of the final curve in the path, i.e. $h_j(m) \coloneqq j(E_n)$ where the walks starts at vertex j and is defined as described above. We see that starting at a different vertex j' results in a different hash function $h_{j'}$ and we indeed get a family $\mathcal{H}$ of hash functions.

We modify this hash function family in three ways. First, we consider a generalisation where we do not ignore one of the $\ell+1$ isogenies from the starting curve E. That is, we take n-symbol inputs $m = b_1 b_2 \ldots b_n$ where $b_1 \in [\ell+1]$ and $b_i \in [\ell]$ for $i \geq 1$; this introduces a one-to-one correspondence between all possible cyclic isogenies of degree ℓ^n originating from E and all elements of the set $[\ell+1] \times [\ell]^{n-1}$.

Secondly, we consider a generalisation where the walk takes place over multiple graphs G_{ℓ_i}. Given an integer $D_m = \prod_{i=1}^n \ell_i^{e_i}$ where the ℓ_i are its prime factors, we introduce the notation $\mu(D_m) \coloneqq \prod_{i=1}^n (\ell_i + 1) \cdot \ell_i^{e_i - 1}$. We then take the message m to be an integer element of the set $[\mu(D_m)]$ represented as a tuple $(m_1, \ldots, m_n)$ where each $m_i \in [(\ell_i+1) \cdot \ell_i^{e_i-1}]$. Each m_i is hashed using a function from the CGL family $\mathcal{H}^{p,\ell_i,e_i}$, that is, hashed along the graph G_{ℓ_i}. To ensure continuity, the j-invariants are chained along the hash functions, that is, we write $j_i = h_{j_{i-1}}^{\ell_i,e_i}(m_i)$, where j_{i-1} is the hash output from the previous element m_{i-1}. Thus, only j_0 parametrises the overall hash function, which we denote by j, as all the others are implicitly defined. As before, this generalization returns the final j-invariant $j_n = h_{j_{n-1}}^{\ell_n,e_n}(m_n)$ as the hash of m.

Thirdly, we also modify the CGL hash function to return the images of two given points under $\phi \coloneqq \sigma \circ \varphi : E_{j_{i-1}} \to E_{j_i}$, where $\varphi_i : E_{j_{i-1}} \to E_i$ is of degree $\ell_i^{e_i}$ and $\sigma : E_i \to E_{j_i}$ is an appropriate isomorphism. Thus each hash function of the family is defined as $h_{j_{i-1}}^{\ell_i,e_i} : (m, P, Q) \mapsto (j_i, \phi_i(P), \phi_i(Q))$, which we combine with our multiple-graph version presented above where the points are also chained through the different functions and only the first points are given as inputs. For the rest of this work, as we will only make use of this family of generalised functions, we therefore refer by $\mathcal{H}^{p,d_m}$ to the hash function family

$$\mathcal{H}^{p,D_m} = \left\{ h_j^{D_m} : [\mu(D_m)] \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2 \to \mathcal{J}_p \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2 \right\}_{j \in \mathcal{J}_p}.$$

3.2 A new one-way function family

Given p, D_m and N, we define a family of functions $\mathcal{F}^{p,D_m,N} : \mathcal{J}_p \times [\mu(D_m)] \to \mathcal{J}_p \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$, which uses the generalised CGL hash function family $\mathcal{H}^{p,D_m}$. We define the function $f_j(m)$ to first compute the canonical curve E_j and compute a canonical basis (P_j, Q_j) of the N-torsion group $E_j[N]$ (which is efficient if N is powersmooth). Next, the function computes $(j_c, P_c, Q_c) = h_j^{D_m}(m, P_j, Q_j)$. Succinctly, we have

$$f_j : m \mapsto \left(h_j^{D_m}(m, P_j, Q_j)\right)$$

Statistically random sampling from the family. The starting curve E_0 with j-invariant $j(E_0) = 1728$ is fixed as part of the global parameters of the family $\mathcal{F}^{p,D_m,N}$. To select a random f_j from $\mathcal{F}$, a random isogeny of degree D_s, with cyclic kernel K_s, is chosen. This fixes $E_s \approx E_0/K_s$, and the corresponding j-invariant $j_s = j(E_s)$, thus fixing the function $f_{j_s} : [\mu(D_m))] \to \mathcal{J}_p \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$.

Theorem 1 ([19], Theorem 1). *For degree $D_s = \prod_i \ell_i^{e_i}$, the distribution of the j-invariant j_s sampled as the last j-invariant of a random walk of length D_s is within statistical distance $\prod_i \left(\frac{2\sqrt{\ell_i}}{\ell_i+1}\right)^{e_i}$ of uniform.*

Following [19, Lemma 1], taking $D_s = \prod_i \ell_i^{e_i}$ with ℓ_i ranging through all primes less than $2(1+\epsilon)\log p$ and $e_i = \max\{e \in \mathbb{N} : \ell_i^e < 2(1+\epsilon)\log p\}$ leads to a statistical distance of less than $1/p^{1+\epsilon}$, for arbitrary ϵ. This also ensures that D_s is B-powersmooth, for $B \approx 2(1+\epsilon)\log p$, which allows for efficient computation of degree-D_s isogenies.

Injectivity. We observe that, for the right choice of parameters, the functions are injective.

Lemma 1. *Let $N^2 > 4D_m$, then any function $f_j \in \mathcal{F}^{p,D_m,N}$ is injective.*

Proof. Suppose that a function f_j is not injective, i.e. that there are two distinct isogenies φ and φ' of degree D_m from E_j to E_c, corresponding to two distinct messages, with the same action on $E_j[N]$, implied by the colliding images of P_j and Q_j. Then, following [23, Section 4], their difference is also an isogeny between the same curves whose kernel contains the entire N-torsion. This, together with [28, Lemma V.1.2], imply that $4D_m \geq \deg(\varphi - \varphi') \geq N^2$. Taking $N^2 > 4D_m$ ensures that in fact $\varphi = \varphi'$ and therefore that f_j is injective. □

One-wayness. We now prove that the functions from this family are one-way under the hardness of an isogeny problem.

Lemma 2. *Let D_s be such that the distribution of j_s is statistically close to uniform. A function $f_j \in \mathcal{F}^{p,D_m,N}$ sampled at random as explained above, is quantum one-way under the hardness of Problem 3 with isogeny degree $d = D_m$ and torsion degree N.*

Proof. Suppose that there is a PPT quantum adversary $\mathcal{A}$ that can break the one-wayness of f_j; that is, given j and $(j_c, P_c, Q_c) = f_j(m^*)$ for $m^* \xleftarrow{\$} [\mu(D_m)]$, $\mathcal{A}$ can recover m^* with non-negligible probability. We build a reduction $\mathcal{B}$ which receives a challenge $(E_1, P_1, Q_1, E_2, P_2, Q_2)$ for Problem 3, with X being the uniform distribution over isogenies of degree D_m, and returns an isogeny $\varphi : E_1 \to E_2$ such that $\varphi(P_1) = P_2$ and $\varphi(Q_1) = Q_2$.

We first observe that since E_1 is uniformly distributed, then so is its j-invariant and its distribution is statistically close to that expected by $\mathcal{A}$ for j, so $\mathcal{A}$ is not able to distinguish such distributions. We also observe that the distribution of isogenies resulting from hashing a uniform $m^* \xleftarrow{\$} [\mu(D_m)]$ is exactly the distribution X of D_m-isogenies. The reduction therefore passes $j(E_1)$ and $(j(E_2), P_2, Q_2)$ to $\mathcal{A}$, who will return a corresponding input m with high probability. By reproducing the hashing of m, the reduction $\mathcal{B}$ can then recompute an isogeny $\tilde{\varphi}$ which is equivalent to φ. Note here that if m is a correct pre-image of (j_c, P_c, Q_c) under the function f_j, then we are certain that it is the only one as, by Lemma 1, f_j is injective. With its knowledge of E_1, P_1 and Q_1, $\mathcal{B}$ can then compute φ and return it. □

3.3 Computing inverses

In this section, we show how to use the algorithm of [24] to invert a given function $f_j \in \mathcal{F}^{p,D_m,N}$. We are given (j_c, P_c, Q_c) as the output of $f_j(m)$ for some unknown m, and also the random isogeny $\phi_s : E_0 \to E_j$ of degree D_s used to select E_j at random. This gives us the composed isogeny $\phi = \phi_m \circ \phi_s : E_0 \to E_m$ of degree $D = D_m \cdot D_s$, where ϕ_m is the walk determined by m, used in the computation of $f_j(m)$.

Computing ϕ_m given a suitable endomorphism of E_0. In this section we assume that we know $\theta \in \mathrm{End}(E_0)$ and $d \in \mathbb{Z}$ such that $\mathrm{Tr}(\theta) = 0$ and $\deg(\phi \circ \theta \circ \hat{\phi} + [d]) = N$. Furthermore we assume that D is odd and that $-4\deg(\theta)$ is not a square modulo every prime divisor of D. We will explain how to find such θ as part of the global parameters for our schemes in Section 3.4; here we describe how to invert the function given such a θ.

Let $\psi = \phi \circ \theta \circ \hat{\phi} + [d] \in \mathrm{End}(E_m)$. We can compute ψ by the following method described in [24]. The endomorphism ψ has degree N and we know its action on $E_m[N]$, thus we can compute its kernel (since it is contained in $E_m[N]$). Since we are able to compute ψ we can compute the kernel of $\ker(\phi \circ \theta \circ \hat{\phi}) \cap E_m[D]$ efficiently. Now let $G = \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_m[D]$. Lemma 3 below shows that in fact $G = \ker(\hat{\phi})$; from this we can recover first $\ker(\phi)$ and then $\ker(\phi_m)$, separating out ϕ_s. This then allows us to recover $m \in [\mu(D_m)]$ which corresponds to $\ker(\phi_m)$. Algorithm 1 summarises these steps in pseudocode.

Algorithm 1 Computing inverses

Require: $\mathsf{c}, \phi_s, \theta \in \mathrm{End}(E_0), d \in \mathbb{Z}$.
Ensure: $m \in [\mu(D_m)]$ such that $f_{j_s}(m) = \mathsf{c}$.
1: Parse c as $(j_c, P_c, Q_c) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$.
2: Compute the canonical curve $E_m = E_j$.
3: Let $\phi = \phi_m \circ \phi_s : E_0 \to E_m$.
4: Let $\psi = \phi \circ \theta \circ \hat{\phi} + [d] \in \mathrm{End}(E_m)$. $\triangleright$ Choices of θ and d ensure $\deg \psi = N$.
5: Compute $K_1 = \ker \psi \subset E_m[N]$ using d, θ, ϕ_s and $P_c, Q_c \in E_m[N]$.
6: Compute $K_2 = \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_m[D] = \ker(\psi - d) \cap E_m[D] = \ker(\hat{\phi})$.
7: Compute $\ker(\phi_m)$ using $\ker(\hat{\phi})$.
8: **return** $m \in [\mu(D_m)]$ that corresponds to $\ker(\phi_m)$.

Lemma 3. *Let θ be such that $-\deg(\theta)$ is a quadratic nonresidue modulo every prime dividing D. Then G is cyclic and furthermore $G = \ker(\hat{\phi})$.*

Proof. It is clear that $\ker(\hat{\phi}) \subset G$ since it is contained in $\ker(\phi \circ \theta \circ \hat{\phi})$ and in $E_m[D]$ as well. We now show that G is cyclic. Let M be the largest divisor of D such that $E_m[M] \subset G$. Then ϕ can be decomposed as $\phi_{D/M} \circ \phi_M$. Then by [24, Lemma 5] the kernel of ϕ_M is fixed by θ. In the proof of [24, Lemma 6] it is shown that a subgroup of $E_0[M]$ can only be fixed by an endomorphism θ if $\mathrm{Tr}(\theta)^2 - 4\deg(\theta)$ is a square modulo M. Choosing θ as above therefore ensures that $M = 1$ which implies that G is cyclic. The order of G is a divisor of D since G is cyclic and every element of G has order dividing D. However, G contains $\ker(\hat{\phi})$ which is a group of order D. This implies that $G = \ker(\hat{\phi})$. □

Avoiding a timing dependency. The condition that $-\deg(\theta)$ is a quadratic nonresidue modulo every prime dividing D may seem strange at first since in [24] the case when G is not cyclic is also considered. Without this condition, M will not always be equal to 1 and in that case the most time-consuming part of the algorithm is guessing a θ-invariant subgroup of $E_0[M]$—this is exponential in the number of prime factors of M and it can be expensive since D is powersmooth. In [24] it is shown that the expected running time of the attack remains polynomial time. This is however not sufficient for our purposes, as inversion could take a very long time on some inputs, and the variable inversion time creates a dependency between the input and the inversion time. By evoking this extra condition on θ and increasing the parameters slightly, we avoid a timing dependency entirely.

Detection of invalid inputs. When provided with a valid ciphertext c, Algorithm 1 will always return the corresponding plaintext. To detect invalid inputs we proceed as follows. If any of the steps fails we return $\perp$ to indicate that the ciphertext is invalid. If the algorithm returns an output $\tilde{m}$ then we recompute the image $\tilde{\mathsf{c}}$ from it; if that matches the original c, then we return $\tilde{m}$ as a valid message; otherwise we return $\perp$.

3.4 Computation of the endomorphism

We now provide an algorithm for finding $\theta \in \mathrm{End}(E_0)$ which does not depend on ϕ_s or ϕ_m, only on their degrees, and can therefore be run as part of global parameter generation. This is essentially just a small modification of [24, Algorithm 2] but it is technical and may be skipped at a first reading.

The ring $\mathrm{End}(E_0)$ has an integral basis $\{1, i, \frac{ij+j}{2}, \frac{1+i}{2}\}$, with $i^2 = -p$ and $j^2 = -1$. As seen in Section 2.2 the endomorphism ring contains the $\mathbb{Z}$-linear combinations of i, j, ij. We will be looking for θ in the form $ai + bj + cij$ with $a, b, c \in \mathbb{Z}$. This means that we are looking for a solution of the following Diophantine equation:

$$D^2(pa^2 + pb^2 + c^2) + d^2 = N \tag{1}$$

Furthermore, we need that $-4\deg(\theta)$ is a quadratic nonresidue modulo every prime divisor of D.

We make certain parameter restrictions which are partly necessary and partly for convenience. First we choose D to be odd since $-4\deg(\theta)$ is obviously a square modulo 2. We choose N to be a square modulo D^2, so the equation will be solvable modulo D^2 and we choose $N > D^5$. Let $D = \prod_{i=1}^{k} \ell_i^{e_i}$ be the prime decomposition of D, and let us denote by $T := \prod_{i=1}^{k} \ell_i$ the product of all distinct prime factors of D. We will also add the restriction that $D > T^3$. Let $A := pa^2 + pb^2 + c^2$. Algorithm 2 below computes a solution to Equation 1 such that $-A$ is a quadratic nonresidue modulo every prime number dividing D.

The following lemmas address the correctness and efficiency of Algorithm 2.

Lemma 4. *Let A be the output of Algorithm 2. Then $-A$ is a quadratic nonresidue modulo all ℓ_i.*

Proof. Let r_i, s_{ℓ_i} and u be as in Algorithm 2. Let r be an integer such that $r \equiv r_i \pmod{\ell_i}$. Then we show that for every i, the integer $\frac{-N+(D^2r+u)^2}{D^2}$ is not a quadratic residue modulo ℓ_i which implies that $-A$ is not a quadratic residue modulo every ℓ_i since $T\ell + r \equiv r_i \pmod{\ell_i}$ for every integer ℓ.

We have that
$$\frac{-N + (D^2r + u)^2}{D^2} = \frac{-N + u^2}{D^2} + D^2r^2 + 2ur.$$

By our choice of r we have that

$$\frac{-N + u^2}{D^2} + D^2r^2 + 2ur \equiv \frac{-N + u^2}{D^2} + 2ur_i \equiv s_{\ell_i} \pmod{\ell_i},$$

which is a quadratic nonresidue by the choice of s_{ℓ_i}. $\square$

Lemma 5. *Under plausible heuristic assumptions Algorithm 2 finds a solution to Equation 1 with the required properties in polynomial time.*

Proof. Lemma 4 implies that $-(pa^2 + pb^2 + c^2)$ is a quadratic nonresidue modulo every ℓ_i. Observe that if $\ell < \frac{T}{2}$ we have that $N - (D^2(T\ell + r) + u)^2 > 0$ because of the conditions $N > D^5$ and $D > T^3$. This implies that whenever $\ell < \frac{T}{2}$ we have that $\frac{A-c^2}{p}$ in Step 13 is a positive number. Moreover, we can estimate the size of $\frac{A-c^2}{p}$ since $A = \frac{N-(D^2(T\ell+r)+u^2)^2}{D^2} \approx D^3$, which implies that $\frac{A-c^2}{p} \approx D^2$. By the Prime number theorem

Algorithm 2 Computing θ

Require: D, N, p as above. Let T be the product of primes dividing D.
Ensure: solution to equation 1 such that $-A$ is a quadratic nonresidue modulo every prime dividing D.

```
1: Find u such that u^2 ≡ N (mod D^2) .
2: for every prime ℓ_i dividing D do
3:     Let s_{ℓ_i} be a quadratic nonresidue modulo ℓ_i.
4:     r_i ← (s_{ℓ_i} − (−N+u^2)/D^2)(2u)^{−1} (mod ℓ_i).
5: Compute a residue r modulo T with the property that r ≡ r_i (mod ℓ_i).
6: ℓ ← 0.
7: d ← D^2(Tℓ + r) + u.
8: A ← (N−d^2)/D^2.
9: if A is not a square modulo p then
10:    ℓ ← ℓ + 1.
11:    go to Step 7.
12: else
13:    Find c such that c^2 ≡ A (mod p).
14:    if (A−c^2)/p is a prime congruent 1 modulo 4 then
15:        Solve the equation a^2 + b^2 = (A−c^2)/p.
16:    else
17:        ℓ ← ℓ + 1.
18:        go to Step 7.
19: return (a, b, c, d)
```

and the Chebotarev density theorem we have that the number of primes smaller than D^2 and congruent to 1 modulo 4 is $O\left(\frac{D^2}{\log(D^2)}\right)$. Thus, after $O(\log p)$ iterations (which is much smaller than $\frac{T}{2}$) we will get that $\frac{A-c^2}{p}$ is a sum of two squares.

Finally, representing a prime number (congruent to 1 modulo 4) as a sum of two squares can be accomplished in polynomial time using Cornacchia's algorithm. All the other steps clearly run in polynomial time. □

Remark 2. The proof implies that instead of having the two conditions $N > D^5$ and $D > T^3$ we could have had the condition $N > D^4T^3$.

4 Public-key encryption schemes

We now build a PKE scheme using the family of trapdoor OWFs of Section 3 and show that it is OW-CPA secure; then we modify it to achieve IND-CCA security.

4.1 OW-CPA encryption scheme

We define the $\mathsf{S\acute{E}TA}_{\mathsf{OW\text{-}CPA}}$ PKE scheme as the tuple $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms described below.

Parameters. Let λ denote the security parameter. Let $p = O(\lambda)$ denote an integer prime number which defines the field $\mathbb{F}_{p^2}$. Let E_0 be a fixed supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with j-invariant $j(E_0) = 1728$. Let D_s, D_m and N be integers chosen according to the requirements of Section 3, and such that $\log \mu(D_m) = \text{poly}(\lambda)$. Let $\theta \in \text{End}(E_0)$ be computed as in Section 3.4. We let $\mathsf{params} = (\lambda, p, j_0, D_s, D_m, N, \theta)$.

Key generation. The $\mathsf{KGen}(\mathsf{params})$ algorithm proceeds as follows:

1. Sample a random cyclic subgroup $K_s \subset E_0(\overline{\mathbb{F}_{p^2}})$ of size D_s.

2. Compute the isogeny $\phi_s : E_0 \to E_s := E_0/\langle K_s \rangle$.
3. Compute the j-invariant $j_s = j(E_s)$ and its canonical curve E_{j_s}.
4. Set $\mathsf{pk} := j_s$ and $\mathsf{sk} := K_s$.
5. Return $(\mathsf{pk}, \mathsf{sk})$.

Encryption. The $\mathsf{Enc}(\mathsf{params}, \mathsf{pk}, m)$ algorithm proceeds as follows. For a given $m \in \{0,1\}^{n_m}$, where $n_m = \lfloor \log_2 \mu(D_m) \rfloor$, first cast m as an integer in the set $[\mu(D_m)]$ and then:

1. Parse $\mathsf{pk} = j_s \in \mathcal{J}_p$.
2. Compute $(j_c, P_c, Q_c) \leftarrow f_{j_s}(m)$, where $f_{j_s} \in \mathcal{F}^{p,D_m,N}$.
3. Embed (j_c, P_c, Q_c) as a binary string $\mathsf{c} \in \{0,1\}^{n_c}$ where n_c is sufficiently large to represent one j-invariant in $\mathcal{J}_p$ and two points in $E_{j_c}[N]$ (see end of Section 4.2).
4. Return c.

Decryption. The $\mathsf{Dec}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{c})$ algorithm proceeds as follows:

1. Given $\mathsf{params}, \mathsf{sk}$ and $\mathsf{c} \in \{0,1\}^{n_c}$, parse c as $(j_c, P_c, Q_c) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$; if that fails, return $\perp$.
2. Follow Algorithm 1 to recover $\tilde{m} \in [\mu(D_m)]$; if this fails, set $\tilde{m} = \perp$.
3. If $\tilde{m} \neq \perp$; verify that $f_{j_s}(\tilde{m}) \stackrel{?}{=} \mathsf{c}$. If not, set $\tilde{m} = \perp$.
4. If $\perp$ was recovered, return $\perp$.
5. Otherwise, from $\tilde{m} \in [\mu(D_m)]$, recover $m \in \{0,1\}^{n_m}$ and return it.

Theorem 2. *Let D_s be such that the distribution of j_s is statistically close to uniform. If Problem 3 with $p, d = D_m,$, N and X such that $H_\infty(X) = \lambda$ is hard for quantum PPT adversaries, then the PKE scheme above is quantum OW-CPA secure.*

Proof. In the notation of Definition 2, we have $\mathcal{M} = \{0,1\}^{n_m}$. We see that a randomly sampled $m \stackrel{\$}{\leftarrow} \mathcal{M}$ directly embedded as an integer $m \in [\mu(D_m)]$ yields a distribution Y with min-entropy $H_\infty(Y) = \lambda$ on isogenies of degree D_m starting from E_s. Similarly to the proof of Lemma 2, the challenge of opening a given ciphertext c reduces to recovering the secret isogeny of Problem 3 with $X = Y$. ◻

4.2 IND-CCA encryption scheme

We now show how to construct $\mathsf{S\acute{E}TA}_{\mathsf{IND\text{-}CCA}}$, an IND-CCA secure encryption scheme based on our OWF of Section 3. We do so with the post-quantum OAEP transformation of [33, Section 5] which we recall below together with its security theorem. Let

$$f : \{0,1\}^{\lambda+k_1} \times \{0,1\}^{k_0} \to \{0,1\}^{n_c}$$

be an invertible injective function. The function f is the public key of the scheme, its inverse f^{-1} is the secret key. The scheme makes use of three hash functions

$$\begin{aligned} G &: \{0,1\}^{k_0} \to \{0,1\}^{k-k_0}, \\ H &: \{0,1\}^{k-k_0} \to \{0,1\}^{k_0}, \\ H' &: \{0,1\}^{k} \to \{0,1\}^{k}, \end{aligned}$$

modelled as random oracles, where $k = \lambda + k_0 + k_1$. Given those, the encryption scheme is defined as follows:

– Enc: given a message $m \in \{0,1\}^\lambda$, choose $r \leftarrow \{0,1\}^{k_0}$ and set

$$s = m||0^{k_1} \oplus G(r), \qquad t = r \oplus H(s),$$
$$c = f(s,t), \qquad d = H'(s||t),$$

and output the ciphertext (c, d).

– Dec: given a ciphertext (c, d), use the secret key to compute $(s, t) = f^{-1}(c)$. If $d \neq H'(s||t)$ output $\perp$. Otherwise, compute $r = t \oplus H(s)$ and $\overline{m} = s \oplus G(r)$. If the last k_1 bits of $\overline{m}$ are 0, output the first n bits of $\overline{m}$, otherwise output $\perp$.

Theorem 3 ([33], Theorem 2). *If f is a quantum partial-domain one-way function, then the OAEP construction is IND-CCA secure in the quantum random oracle model.*

By proving the following lemma, we show that one can use the above quantum OAEP transformation to construct an IND-CCA secure encryption using our OWF.

Lemma 6. *The function f defined in Section 3.2 is a quantum partial-domain one-way function.*

Proof. We note that in our case, partial domain inversion is the same as domain inversion where only the first part of the path is required. More precisely, factor D_m as $D'_m \cdot D''_m$ such that $\gcd(D'_m, D''_m) = 1$, $2^{\lambda+k_1} \leq \mu(D'_m)$ and $2^{k_0} \leq \mu(D''_m)$ and then embed each of s and t in the respective factors. If D'_m is appropriately set, then recovering s from $c = f(s, t)$ is hard under the same assumption as Theorem 2 with D_m replaced by D'_m. □

Efficiency. An OW-CPA secure ciphertext is composed of a j-invariant $j_c \in \mathbb{F}_{p^2}$, which can be represented with $2 \log p$ bits, and two torsion points $P_c, Q_c \in E_{j_c}[N]$, each of which can be represented with $2 \log N$ bits by identifying each N-torsion point with a pair of elements in $\mathbb{Z}_N$. Further compression is possible, representing both torsion points with $3 \log N$ bits, using the techniques in [10, Section 6.1]. The IND-CCA version adds an output of a hash function H', which has the same size as the input of the one-way function f. Thus the total size in bits of the ciphertext is

$$2 \log p + 4 \log N + k.$$

4.3 Comparison with SIDH-based encryption scheme

Prior to this work, the main method to obtain a PKE scheme from supersingular isogenies was to adapt the original key agreement protocol of [21] in an ElGamal fashion as described in [13, Section 3.3]. In that scheme, the public-key generation can be seen as a partial key agreement where one party generates their secret isogeny and publishes the target curve, together with the images of a torsion basis, as its long-term static key. The main advantage of our new encryption scheme compared to the above is that their security is based on a priori easier isogeny problems: future cryptanalysis progress might affect SIKE without affecting our schemes.

More precisely, the IND-CPA security of the original schemes of [21,13] and the SIKE version [2] rely on the supersingular DDH and CDH problems, respectively; that is, given E_0, E_A, E_B as in Figure 1 and the corresponding images of torsion points, respectively distinguish E_{AB} from random or compute E_{AB}. In this work, we approach the original "discrete logarithm"-like assumption (given two curves, compute an isogeny between them) as we prove OW-CPA security under the hardness of this problem when also given images of torsion points. While OW-CPA is a weaker notion than IND-CPA, the generic OAEP transformation in the random oracle model provides us with IND-CCA security. We note that SIKE also uses the random oracle model, even for IND-CPA security.

Importantly, in SIDH-based schemes the starting curve E_0 is fixed for efficiency reasons and therefore the additional hardness of Problem 3 that comes from a random starting curve does not apply. Furthermore, the curves E_0 and E_A are somewhat close in the underlying isogeny graph because of the chosen degrees. In contrast, the security of our schemes of Sections 4.1 and 4.2 benefit from the full hardness of Problem 3 and the use of significantly longer isogenies.

Considering the notion of key recovery, (Definition 3), we can see that, in the case of [13], it is directly related to the hardness of CSSI [13, Problem 5.2] as recovering the secret isogeny enables any attacker to complete the key agreement and decrypt the message. Not only does this problem include the torsion point images which means that it can be weak against Petit's attacks [24], but the static nature of the key also opens the scheme to active attacks [18].

In contrast, our scheme of Section 4.1 does not suffer from such attacks; the torsion point images that we reveal depend only on the plaintext. Indeed, the key recovery problem for our scheme consists of recovering an equivalent isogeny between the curves E_0 and E_s *without additional torsion information* (Problem 2), or, equivalently directly computing the endomorphism ring of E_s (Problem 1)—either of these options would allow to directly evaluate the endomorphism $\phi_s \circ \theta \circ \hat{\phi}_s \in \mathrm{End}(E_s)$ in the inversion algorithm of Section 3.3. This guarantees stronger key-recovery security to our schemes in contrast to SIDH and its variants. We formalise this in the following result.

Theorem 4. *If there exists a quantum PPT adversary $\mathcal{A}$ against the key recovery security of the scheme of Section 4.1 then there exists a PPT adversary $\mathcal{B}$ against Problem 2 for the curves E_0 and E_s.*

Proof. Given E_1 and E_2 as in the statement above, $\mathcal{B}$ computes $j_s = j(E_2)$ and submits j_s to $\mathcal{A}$ as the public key. When $\mathcal{A}$ returns an alternative secret key sk', $\mathcal{B}$ checks that it is valid and returns it as a solution to Problem 2. □

5 Key encapsulation mechanism

We discuss two generic transformations that we can apply to our encryption scheme to obtain an IND-CCA secure KEM in the quantum random oracle model (QROM). The first one works for any OW-CPA encryption scheme, but has the drawback of losing a large tightness factor in the security reduction. The second one does have tighter reductions, but requires an additional property from the starting scheme, called sparse pseudorandomness. We prove that our scheme verifies this property under the hardness of Problem 4. We refer to [20,4] for transformations in the classical random oracle model.

5.1 A non-tight transformation

We first describe the $\mathsf{QFO}_m^{\not\perp}$ transformation from [20], which takes a OW-CPA secure PKE scheme and produces an IND-CCA secure key encapsulation mechanism. This is based on a previous transformation by Targhi-Unruh [33], which in turn is essentially a QROM secure version of the Fujisaki-Okamoto transformation [16]. Following the recommendation in [4, Section 16], we choose the variant with implicit rejection, that is, when the ciphertext is invalid, the decapsulation algorithm outputs a wrong key instead of $\perp$.

Let $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, with message space $\mathcal{M} = \{0,1\}^\lambda$ and randomness space $\mathcal{R}$. Also, let

$$G : \{0,1\}^\lambda \to \mathcal{R}, \qquad H : \{0,1\}^* \to \{0,1\}^\lambda, \qquad H' : \{0,1\}^\lambda \to \{0,1\}^\lambda$$

be three hash functions, modelled as random oracles. The $\mathsf{QFO}_m^{\not\perp}$ transformation outputs the following KEM:

$\overline{\mathsf{KGen}}(1^\lambda)$:	$\overline{\mathsf{Enc}}(pk)$:	$\overline{\mathsf{Dec}}(dk, c, d)$:
$(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$	$m \xleftarrow{\$} \mathcal{M}$	$m = \mathsf{Dec}_{sk}(c)$
$s \xleftarrow{\$} \mathcal{M}$	$c = \mathsf{Enc}_{pk}(m; G(m))$	**if** $c \neq \mathsf{Enc}_{pk}(m; G(m))$ **or** $H'(m) \neq d$
return (pk, dk)	$d = H'(m)$	**return** $K = H(s, c, d)$
$dk = (pk, sk, s)$	$K = H(m)$	**else return** $K = H(m)$.
	return (K, c, d)	

Fig. 2. The $\mathsf{QFO}_m^{\not\perp}$ transformation

Theorem 5 (Theorems 4.4 and 4.6 from [20]). *Let* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with perfect correctness that is OW-CPA secure. Then the* $\mathsf{QFO}_m^{\not\perp}$ *transformation above produces a KEM that is IND-CCA secure in the quantum random oracle model. More precisely, for any quantum PPT adversary* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}_{\mathsf{KEM},\mathcal{A}}^{\mathsf{IND-CCA}}(\lambda) \le 8q^{3/2}\left(\mathsf{Adv}_{\mathsf{PKE},\mathcal{B}}^{\mathsf{OW-CPA}}(\lambda)\right)^{1/4},$$

where q *is the number of queries made to any of the random oracles.*

We note that the theorem in [20] is more general and covers the case in which the PKE scheme has a decryption error, but we do not need this. Observe that there is a large tightness factor that is lost. In the next section, we present another transformation with a tighter reduction.

Corollary 1. *The scheme described in Section 4.1, combined with the* $\mathsf{QFO}_m^{\not\perp}$ *transformation, is a quantum IND-CCA secure KEM under the hardness of Problem 3.*

Proof. Direct application of Theorems 2 and 5. □

Efficiency. The overhead of the transformation over the OW-CPA scheme is the addition of the two hashes $H(m), H'(m)$ (our PKE scheme is deterministic, so we do not need the hash $G(m)$).

5.2 A tighter transformation

Sparse pseudorandomness. In [26], the authors provide a transformation from a weakly secure public key encryption scheme to a CCA-secure KEM. Their security reductions are tight, but unlike other proposals, they require an additional property from the original PKE scheme, which they call sparse pseudorandomness. Informally, this means that the ciphertexts of a random message are computationally indistinguishable from uniformly random elements of the ciphertext space (pseudorandomness), and at the same time the probability of a random element of the ciphertext space being a valid ciphertext is negligible (sparseness).

Definition 6 (Definition 3.2 from [26]). *A deterministic public-key encryption scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *with plaintext space* $\mathcal{M}$ *and ciphertext space* $\mathcal{C}$, *is* sparse pseudorandom *if the following two properties are satisfied.*

- *Sparseness:*
$$\mathsf{Sparse}_{\mathsf{PKE}}(\lambda) := \max_{(pk,sk)\in\mathsf{KGen}(1^\lambda)} \frac{\#\mathsf{Enc}_{pk}(\mathcal{M})}{\#\mathcal{C}} \le \mathsf{negl}(\lambda).$$
- *Pseudorandomness: for any PPT adversary* $\mathcal{A}$,
$$\mathsf{Adv}_{\mathsf{PKE},\mathcal{A}}^{\mathsf{PR}}(\lambda) := \left| \begin{array}{l} \Pr\left[\begin{array}{c} (pk,sk) \leftarrow \mathsf{KGen}(1^\lambda); \\ m^* \xleftarrow{\$} \mathcal{M}; \\ c^* = \mathsf{Enc}_{pk}(m^*) \end{array} : 1 \leftarrow \mathcal{A}(pk, c^*)\right] \\ - \Pr\left[\begin{array}{c} (pk,sk) \leftarrow \mathsf{KGen}(1^\lambda); \\ c^* \xleftarrow{\$} \mathcal{C} \end{array} : 1 \leftarrow \mathcal{A}(pk, c^*)\right] \end{array} \right| \le \mathsf{negl}(\lambda).$$

The SXY transformation We now describe the SXY transformation introduced in [26], which takes a deterministic PKE scheme that is sparse pseudorandom and produces an IND-CCA secure key encapsulation mechanism.

Let $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a sparse pseudorandom deterministic public-key encryption scheme with message space $\mathcal{M} = \{0,1\}^\lambda$ and ciphertext space $\mathcal{C}$. Also, let

$$H : \{0,1\}^\lambda \to \{0,1\}^\lambda, \qquad H' : \{0,1\}^\ell \times \mathcal{C} \to \{0,1\}^\lambda$$

be two hash functions, modelled as random oracles. The SXY transformation outputs the following KEM:

$\overline{\mathsf{KGen}}(1^\lambda)$:	$\overline{\mathsf{Enc}}(pk)$:	$\overline{\mathsf{Dec}}(dk, c)$:
$(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$	$m \xleftarrow{\$} \mathcal{M}$	$m = \mathsf{Dec}_{sk}(c)$
$s \xleftarrow{\$} \{0,1\}^\ell$	$c = \mathsf{Enc}_{pk}(m)$	**if** $m = \perp$ **return** $K = H'(s, c)$
$dk = (pk, sk, s)$	$K = H(m)$	**if** $c \neq \mathsf{Enc}_{pk}(m)$ **return** $K = H'(s, c)$
return (pk, dk)	**return** (K, c)	**else return** $K = H(m)$.

Fig. 3. The SXY transformation

Theorem 6 (Theorem 4.2 and Lemma 3.1 from [26]). *Let* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a deterministic PKE scheme with perfect correctness that is sparse pseudorandom. Assume that the ciphertext space* $\mathcal{C}$ *is efficiently sampleable. Then the SXY transformation above produces a KEM that is IND-CCA secure in the quantum random oracle model. More precisely, for any quantum PPT adversary* $\mathcal{A}$ *there exists and adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{IND-CCA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{PR}}_{\mathsf{PKE},\mathcal{B}}(\lambda) + \mathsf{Sparse}_{\mathsf{PKE}}(\lambda) + 2^{\frac{-\ell+1}{2}} q_{H'},$$

where $q_{H'}$ *is the number of queries made to* H'.

Efficiency. The overhead of the transformation over the OW-CPA scheme is the addition of the hash $K = H(m)$.

Finally, we prove that our encryption scheme is sparse pseudorandom, and therefore we can apply the SXY transformation to it. Recall that our encryption function is defined as

$$\mathsf{Enc}_{pk}(m) = (j(E_m), \varphi_m(P), \varphi_m(Q)),$$

where $pk = E_s$ is a supersingular elliptic curve, $\{P, Q\}$ is a basis of the N-torsion of E_s, and $\varphi_m : E_s \to E_m$ is the isogeny corresponding to the CGL hash function with input m. The message space is $\mathcal{M} = \{0,1\}^n$. To guarantee that the two conditions above are satisfied, we must carefully choose the ciphertext space $\mathcal{C} \subset \mathcal{V} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$, where $\mathcal{V} = \mathcal{J}_p$ is the set of vertices of the supersingular isogeny graph. In particular, to have pseudorandomness we must ensure that there is no way to distinguish random elements of $\mathcal{C}$ from valid ciphertexts. We impose the following conditions on $\mathcal{C}$:

- An element $(j(\overline{E}), \overline{P}, \overline{Q}) \in \mathcal{C}$ must satisfy that $\overline{E}$ is isogenous to E and $\overline{P}, \overline{Q} \in \overline{E}[N]$.
- The elements $\overline{P}, \overline{Q}$ must be of order N and linearly independent.
- Note that $e(\varphi_m(P), \varphi_m(Q)) = e(P, Q)^{D_m}$, where e is the Weil pairing. Therefore $(j(\overline{E}), \overline{P}, \overline{Q}) \in \mathcal{C}$ must satisfy that $e(\overline{P}, \overline{Q}) = e(P, Q)^{D_m}$.

Note that the third condition implies the second when N and D_m are coprime, which is the case for our constructions.

We now prove that our scheme is sparse pseudorandom.

Lemma 7. *Let* $\epsilon > 0$. *Assume that* $p^{1-\varepsilon} N^3 > \mu(D_m)$ *and* D_m *is large enough to ensure that the output of a random walk of degree* D_m *is close to uniform. Then the encryption scheme defined above is sparse in* $\mathcal{C}$.

Proof. Our aim is to prove that $\#\mathsf{Enc}_{pk}(\mathcal{M})/\#\mathcal{C}$ is negligible. Since the encryption function is injective, we have that $\#\mathsf{Enc}_{pk}(\mathcal{M}) = \#\mathcal{M} = 2^{\lfloor \log \mu(D_m) \rfloor}$. On the other hand, $\#\mathcal{C}$ can be factored in the number of valid j-invariants times the number of valid pairs of points for each curve.

We observe that, if D_m is large enough, the mixing property of expander graphs ensures that the probability of ending a random walk of degree D_m at any j-invariant on the graph is bounded away from 0. Therefore the number of valid j-invariants is the size of the graph, which is $\lfloor p/12 \rfloor + k$ where $k \in \{0, 1, 2\}$.

For the number of valid pairs, we fix a supersingular j-invariant $j(\overline{E}) \in \mathcal{V}$. We observe that $\overline{E}[N] = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$, and we are interested in finding how many choices of $(\overline{P}, \overline{Q}) \in \overline{E}[N] \times \overline{E}[N]$ correspond to a valid ciphertext, that is, that they verify the pairing condition. There are roughly N^3 such pairs, as we have N^4 pairs in the torsion and we impose one equation on them.

Therefore

$$\frac{\#\mathsf{Enc}_{pk}(\mathcal{M})}{\#\mathcal{C}} \approx \frac{\mu(D_m)}{\frac{p}{12} N^3} > \frac{12}{p^{\varepsilon}},$$

which is negligible in the security parameter. □

Proving pseudorandomness information-theoretically does not seem possible, given the result above, so we rely on a hardness assumption.

Lemma 8. *The encryption scheme defined above is pseudorandom under the hardness of Problem 4.*

Proof. The pseudorandomness game is exactly distinguishing between the two distributions in Problem 4. □

Corollary 2. *The scheme described in Section 4.1, combined with the SXY transformation, is a quantum IND-CCA secure KEM under the hardness of Problem 4.*

Proof. Direct application of Theorem 6, and Lemmas 7 and 8. □

5.3 Comparison to SIKE.

Most of the differences between our KEM and SIKE are inherited from those discussed in Section 4.3, and in particular security of our KEM relies on arguably harder problems.

Generic transformations are used in both SIKE and our KEM to achieve IND-CCA security. SIKE makes use of the transformations in [20], which work out of the box. We do the same in Section 5.1, but in Section 5.2 we apply another transformation from [26], which requires a stronger starting property. To the best of our knowledge, this approach has not been explored for the SIKE family.

6 Parameter selection

In this section, we first summarise the conditions which the parameters of our scheme need to satisfy in order to achieve OW-CPA security and for decryption to be effective. We then suggest concrete parameters.

6.1 Parameter requirements

Recall that λ is the security parameter, p is the characteristic of the field, D_s, D_m are the degrees of the secret key and message isogenies, respectively, N is the order of torsion points whose image is revealed, and T is the product of all distinct prime factors of $D = D_s D_m$.

Algorithmic requirements. We choose $\log p = O(\lambda)$ for efficient arithmetic. We need to efficiently represent N-torsion points, so we require that N is powersmooth, with a powersmooth bound as small as possible. Key generation and encryption depend on performing a random walk in the isogeny graph. This can be done efficiently as long as the degree of the corresponding isogenies is powersmooth. The conditions for efficient decryption and avoiding a timing dependency are discussed in Section 3. In Table 1, we list the conditions required for the efficiency of our algorithms.

Requirement	Condition
Efficiency of computations	$\log p = O(\lambda)$
Representation of N-torsion points	N powersmooth
Efficiency of key generation	D_s powersmooth
Efficiency of encryption	D_m powersmooth
Existence of θ	$D \equiv 1 \bmod 2$
Injectivity of functions	$N^2 > 4D_m$
Solvability of Diophantine equation	$D > p$ and $N > D^4$
Inversion is constant time	$N > D^5$, $D > T^3$ and $N \bmod D$ is square

Table 1. List of parameter conditions for efficiency.

Security requirements. Next, we focus on the conditions required for security. We start by reviewing the hardness of the computational problems involved, which we presented in Section 2.3.

If the degree d were not specified in Problem 2, it could be solved in classical $\tilde{O}(\sqrt{p})$ time [17,14]. By specifying d, one can instead apply a claw-finding algorithm by computing all isogenies of degree $\sqrt{d}$ starting from E_1 and then looking for a collision with isogenies of degree $\sqrt{d}$ starting from E_2. Adapting the algorithms from [17,14] results in $\tilde{O}(\sqrt{d})$ classical running time.

Using Tani's quantum claw-finding algorithm [31] one could instead obtain a quantum algorithm with running time $\tilde{O}(\sqrt[3]{d})$, where time is quantified as the number of isogeny evaluation queries. However, based on the recent proposition of Adj et al. [1] that the van Oorschot–Wiener algorithm [34] is a better classical solution, Jaques and Schanck [22] argued that, in fact, running the query-optimal version of Tani's algorithm to achieve $\tilde{O}(\sqrt[3]{d})$ time would require enough hardware that could be repurposed to run van Oorschot–Wiener algorithm in time $\tilde{O}(\sqrt[4]{d})$. Adopting a reasonable constraint on such hardware, they therefore estimate that both the best classical and quantum algorithms require $\tilde{O}(\sqrt{d})$ time to solve Problem 2.

Problem 1 does not involve finding isogenies of a given degree and therefore the claw-finding technique used against Problem 2 cannot be used. Instead, this problem can be solved in classical $\tilde{O}(\sqrt{p})$ time [17] and in quantum $\tilde{O}(\sqrt[4]{p})$ time [5]. We note that when $d > p$, it may actually be more efficient to solve Problem 2 by first solving a related instance of Problem 1, independent of d and then computing the isogeny using the endomorphism ring instead of the claw-finding strategy. This may be the case in our setting, but since we are already considering explicitly the hardness of Problem 1, we ensure that the choice of p is appropriate for security.

For the ciphertext space to be sampleable, we require N to be powersmooth, as discussed in Remark 1 and Appendix B.

Remark 3. To achieve statistical uniformity of the j-invariants obtained through random walks, we must ensure that the walks are long enough, as discussed in Section 3.2. This amounts to choosing D_s, D_m as the product $\prod \ell_i^{e_i}$, where $\ell_i^{e_i}$ are all the highest prime powers smaller than $2\log p$, for all primes ℓ_i. However, recall that we also need N to be powersmooth, and at the same time $\gcd(D, N) = 1$, so we must distribute small primes between D and N. The simplest way is to alternate assigning a prime to D and one to N, in each case going up to the necessary bound imposed by the rest of the conditions. Alternative distributions of the primes could be considered to optimise computations.

Regarding the post-quantum OAEP transformation of Section 4.2, we first recall that we factor D_m as $D'_m \cdot D''_m$ such that $\gcd(D'_m, D''_m) = 1$, $2^{\lambda+k_1} \leq \mu(D'_m)$ and $2^{k_0} \leq \mu(D''_m)$. Thus we require the conditions $\lfloor \log \mu(D'_m) \rfloor \geq \lambda + k_1$ and $\lfloor \log \mu(D''_m) \rfloor \geq k_0$. We now determine k_0 and k_1. Since we need the output length of the hashes to be at least 2λ to avoid collision-finding attacks[7], we require $k_0 \geq 2\lambda$. Since we also want $k - k_0 = \lambda + k_1 \geq 2\lambda$, we set $k_1 \geq \lambda$. Thus, the conditions on D'_m and D''_m become $\lfloor \log \mu(D'_m) \rfloor \geq 2\lambda$ and $\lfloor \log \mu(D''_m) \rfloor \geq 2\lambda$. Table 2 summarises the conditions required for security.

[7] This seems to be an ongoing research. While the conservative choice would be to account for Grover's algorithm and take $t = 3\lambda$, there has been some arguments against quantum collision-finding algorithms in practice [3], so most works have suggested $t = 2\lambda$.

Requirements	Condition
Problem 1 is hard	$\log p \geq 4\lambda$
Problem 3 is hard for OW-CPA	$\log D_m \geq 2\lambda$
Problem 3 is hard for IND-CCA	$\log D'_m \geq 2\lambda$
Sampleable $\mathcal{C}$	N powersmooth
Statistical uniformity of j_s, j_m	See Remark 3
OAEP transform is secure	$\log \mu(D'_m) = \log \mu(D''_m) \geq 2\lambda$
Ciphertexts do not leak information	$\gcd(D, N) = 1$
Ciphertexts are sparse	$p \cdot N^3 > \mu(D_m)$

Table 2. List of parameter conditions necessary for security.

6.2 Concrete parameters

After reviewing parameter restrictions for efficiency and security we suggest concrete parameters. The list of parameters we need to specify is D_m, D_s, p, N and the endomorphism θ. To avoid specializing the problems in any way we choose a random large prime (450 bits) as opposed to a prime of a special form. First we give an example for the integer parameters and then we deal with θ separately. The numbers D_m, D_s and N are given by their prime decomposition to highlight their powersmoothness.

1. $D_s = (17^4) \cdot (23^3) \cdot (31^3) \cdot (37^2) \cdot 53 \cdot (71^2) \cdot (73^2) \cdot (89^1) \cdot (97^2) \cdot 101 \cdot 107 \cdot 113 \cdot 811 \cdot 1229 \cdot 1291 \cdot 2153 \cdot 2999 \cdot 3323 \cdot 4007 \cdot 5557 \cdot 5623$
2. $D_m = (17^4) \cdot (23^2) \cdot (31^2) \cdot (37^3) \cdot (53^2) \cdot (71^1) \cdot (73^2) \cdot (89^2) \cdot (97^1) \cdot 101 \cdot (107^2) \cdot 113 \cdot (811^2) \cdot 1229 \cdot 1291 \cdot 2153 \cdot 3313 \cdot 3517 \cdot 4889 \cdot 5209$
3. $N = (21^8) \cdot (29^8) \cdot (41^8) \cdot (43^8) \cdot (59^8) \cdot (61^8) \cdot (67^8) \cdot (83^8) \cdot (103^8) \cdot (139^4) \cdot (149^4) \cdot (233^4) \cdot (283^4) \cdot (311^4) \cdot (443^4) \cdot (491^4) \cdot (599^4) \cdot (619^4) \cdot (631^4) \cdot (761^2) \cdot (1321^2) \cdot (1327^2) \cdot (1373^2) \cdot (1433^2) \cdot (1571^4) \cdot (1579^4) \cdot (1733^4) \cdot (1741^4) \cdot (1753^2) \cdot (1787^2) \cdot (1931^4) \cdot (2083^2) \cdot (2843^2) \cdot (2857^2) \cdot (2579^4) \cdot (2591^4) \cdot (2621^4) \cdot (2971^4) \cdot (3001^4) \cdot (3011^2) \cdot (3217^4) \cdot (3221^4) \cdot (3541^4) \cdot (3617^2) \cdot (3967^2) \cdot (4021^2) \cdot (4691^2) \cdot (5413^2) \cdot (6791^2) \cdot (7057^2) \cdot (7307^2) \cdot (7487^2) \cdot (7523^2) \cdot (7883^2) \cdot (6151^2) \cdot (6173^2) \cdot (6197^2) \cdot (7127^2) \cdot (8713^2) \cdot (8867^2) \cdot (9431^2) \cdot (9209^2) \cdot (8951^2) \cdot (9397^2) \cdot (9463^2) \cdot (9547^2) \cdot (9643^2) \cdot (9931^2) \cdot (10957^2) \cdot (11443^2) \cdot (11447^2)$
4. $p =$ 23017678136010346213332577752065706892114306007377568563595997
   12828218867264882060938936126891411134546286806604551293695256541
   173852591

Now we turn our attention to θ. We implemented Algorithm 2 in MAGMA [6] to compute a suitable solution of Equation 1. We describe θ as a linear combination $ai + bj + cij$ as described in Section 3.3. To make verification easier we also disclose the value d in the solution of Equation 1:

1. $a =$ 47000468043585093198198624282434132830896002783759029074383774
   21082196898538929595378818129254297377088456585243627941929029192
   4182348665487
2. $b =$ 30985193965478054610126362437290833548435111205067023273851442
   48674792964230417880936080279712111562524815125415610483084803741
   5974030967808
3. $c =$ 30676687592556539096725306619083264341364898713699913576623186
   45291546831673839677853088182832098785291916003831085150626387002
   70268819
4. $d =$ 71661949387317897845939224015166218786859893202150351473026284
   32684449117257520669288979589497036094977019772975131377270923771
   55859302478387875305023424177755812219063102130559574446965608302
   61073811851770476170787462031458033843164639656685661083993117520
   16825531224628633496234647956882453339473372623136494929818982771
   2323916045170463515

Running Algorithm 2 took less than 2 minutes on a standard laptop, which makes the generation of θ efficient, as this only has to be computed once at the parameter generation phase.

There are various options for improving efficiency. One could potentially improve on Algorithm 2 to be able to find a solution to Equation 1 for better (smaller and less unbalanced) parameters. Another direction could be improving on Petit's attack, possibly relating it to another Diophantine equation. Finally, here we chose a random prime p and and random powersmooth and coprime N and D. The scheme could be potentially made more efficient by special choices of D, N and p.

7 Conclusion

In this paper, we introduced a new trapdoor mechanism for isogeny-based cryptography, which relies on using Petit's techniques to compute secret isogenies using extra torsion point information in a constructive way. First we used this idea to build a new trapdoor one-way function, and we then derived public-key encryption schemes and key encapsulation mechanisms secure in the quantum random oracle model. Compared to protocols that are directly derived from SIDH [13,2], our protocols rely on computational problems that are a priori harder to solve. In particular, key recovery only relies on the "pure" isogeny problem for supersingular elliptic curves, where a random prime is used, the curves are sampled at random, and no torsion point information is revealed.

Acknowledgements. The first author was supported for this work in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT and by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under contract No. N66001-15-C-4070. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ERC, United States Air Force, or DARPA. Work by the second and third authors was supported by an EPSRC New Investigator grant (EP/S01361X/1). The fourth author was supported by a PhD grant from the Spanish government, co-financed by the ESF (Ayudas para contratos predoctorales para la formación de doctores 2016). This work was partially done while the fourth author visited the University of Birmingham.

References

1. Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes, and Francisco Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. In *International Conference on Selected Areas in Cryptography*, pages 322–343. Springer, 2018.
2. Reza Azarderakhsh, Matthew Campagna, Craig Costello, LD Feo, Basil Hess, A Jalali, D Jao, B Koziel, B LaMacchia, P Longa, et al. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017.
3. Daniel J Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete. *SHARCS*, 9:105, 2009.
4. Daniel J. Bernstein and Edoardo Persichetti. Towards KEM unification. Cryptology ePrint Archive, Report 2018/526, 2018. `https://eprint.iacr.org/2018/526`.
5. Jean-Francois Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *Progress in Cryptology – INDOCRYPT 2014*, pages 428–442. Springer, 2014.
6. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *Journal of Symbolic Computation*, 24(3-4):235–265, 1997.
7. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.
8. Denis X Charles, Kristin E Lauter, and Eyal Z Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009.
9. Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

10. Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 679–706. Springer, 2017.
11. Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 572–601, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
12. Luca De Feo and Steven D Galbraith. SeaSign: Compact isogeny signatures from class group actions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 759–789. Springer, 2019.
13. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.
14. Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016.
15. Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In *Advances in Cryptology – EUROCRYPT 2018*, pages 329–368. Springer, 2018.
16. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual International Cryptology Conference*, pages 537–554. Springer, 1999.
17. Steven D Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS Journal of Computation and Mathematics*, 2:118–138, 1999.
18. Steven D Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *Advances in Cryptology – ASIACRYPT 2016*, pages 63–91. Springer, 2016.
19. Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 3–33. Springer International Publishing, 2017.
20. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2017.
21. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.
22. Samuel Jaques and John M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *Advances in Cryptology – CRYPTO 2019*, pages 32–61. Springer, 2019.
23. Chloe Martindale and Lorenz Panny. How to not break SIDH. Cryptology ePrint Archive, Report 2019/558, 2019. `https://eprint.iacr.org/2019/558`.
24. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 330–353. Springer International Publishing, 2017.
25. Christophe Petit and Kristin E Lauter. Hard and easy problems for supersingular isogeny graphs. Cryptology ePrint Archive, Report 2017/962, 2017. `https://eprint.iacr.org/2017/962`.
26. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 520–551. Springer, 2018.
27. J. H. Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151 of *Graduate Texts in Mathematics*. Springer-Verlag, New-York, 1994.
28. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
29. Anton Stolbunov. Cryptographic schemes based on isogenies. Doctoral thesis, 2012.
30. Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. in Math. of Comm.*, 4(2):215–235, 2010.
31. Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009.
32. Oleg Taraskin, Vladimir Soukharev, David Jao, and Jason LeGrow. An isogeny-based password-authenticated key establishment protocol. Cryptology ePrint Archive, Report 2018/886, 2018. `https://eprint.iacr.org/2018/886`.
33. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *Theory of Cryptography Conference*, pages 192–216. Springer, 2016.
34. Paul C Van Oorschot and Michael J Wiener. Parallel collision search with cryptanalytic applications. *Journal of cryptology*, 12(1):1–28, 1999.
35. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris, Séries A*, 273:305–347, 1971.

36. Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In *International Conference on Financial Cryptography and Data Security*, pages 163–181. Springer, 2017.

Supplementary material

A Isogeny sampling in SIKE and the CSSI problem

In their paper introducing SIDH [13], Jao and De Feo specify that the kernel generator is selected as $[m]P + [n]Q$, with $m, n \xleftarrow{\$} \mathbb{Z}_{\ell^e}$ not both divisible by ℓ (taking $d = \ell^e$ in this case). This ensures that every one of the $(\ell+1)\ell^{e-1}$ degree-d isogenies can be selected as the challenge. The CSSI problem defined in that paper therefore naturally assumes that the isogeny are sampled uniformly at random.

However for increased efficiency, it is proposed in [11, Section 4] to sample the generator points as $P + [\ell \cdot m]Q$ for $m \in [\ell^{e-1}]$. This has the consequence of only sampling from 1/3, resp. 1/4, of the possible isogenies, for $\ell = 2$ and $\ell = 3$ respectively. A similar method is included in the SIKE specification [2, Section 1.3.5] which furthermore samples m only in the set $[2^{\lfloor \log 3^{e_3} \rfloor}]$, therefore not reaching the full range of possible values. It is not expected that such imperfect sampling makes the CSSI problem easier, especially since such sampling methods still yield distributions of isogenies with min-entropy of the order of $O(\lambda)$. Nonetheless, we have included this difference into Problem 3 to make this sampling discrepancy more explicit.

B Sampling in Problem 4

We prove that sampling elements of the second distribution is efficient. Indeed, let R, S be a basis of $E_2[N]$. We can identify torsion points $xR+yS$ with elements $(x, y) \in \mathbb{Z}_N \times \mathbb{Z}_N$. Then we are looking for pairs $(a, b), (c, d)$ that verify the pairing equation. We can sample them in the following way. We write $N = \prod_{i=1}^{k} \ell_i^{e_i}$, where the ℓ_i are the prime factors of N. We denote the order of x by $|x|$.

1. Choose $a, b \xleftarrow{\$} \mathbb{Z}_N$ such that $|(a, b)| = N$.
2. If $\exists a^{-1} \in \mathbb{Z}_{\ell_i^{e_i}}$, choose $c_i \xleftarrow{\$} \mathbb{Z}_{\ell_i^{e_i}}$, else if $\exists b^{-1} \in \mathbb{Z}_{\ell_i^{e_i}}$, choose $d_i \xleftarrow{\$} \mathbb{Z}_{\ell_i^{e_i}}$.
3. Solve $ad_i - bc_i = t \bmod \ell_i^{e_i}$ for all $i = 1, \ldots, k$.
4. Recover $a, b, c, d \bmod N$ via Chinese remainder theorem.

We now show why this algorithm works and produces uniformly random pairs verifying the condition above. We first note that in Step 2, we will always have that either a or b has multiplicative inverse. We note that $|(a, b)| = N$ in $\mathbb{Z}_N$ implies $|(a, b)| = \ell_i^{e_i}$ in $\mathbb{Z}_{\ell_i^{e_i}}$, and since

$$|(a, b)| = \mathrm{lcm}(|a|, |b|),$$

this in turn implies that either a or b is of maximal order in $\mathbb{Z}_{\ell_i^{e_i}}$.

We have that

$$e(aR + bS, cR + dS) = e(R, S)^{ad-bc},$$

using that the pairing is bilinear and alternating. Then we want to impose condition (3),

$$e(R, S)^{ad-bc} = e(P, Q)^{\deg \varphi},$$

which is equivalent to

$$x(ad - bc) = \deg \varphi,$$

where x is the discrete logarithm of $e(R, S)$ with respect to $e(P, Q)$ (this can be efficiently computed as long as N is smooth). Therefore, the pairs satisfying condition (3) above are the solutions of the equation

$$ad - bc = t,$$

where $t = x^{-1} \deg \varphi$ (note that x is invertible because $\{R, S\}$ is a basis of $E_2[N]$). Finally, the equation modulo prime powers can be solved as

$$d_i = a^{-1}(t + bc_i) \mod \ell_i^{e_i}, \qquad \text{or} \qquad c_i = b^{-1}(ad_i - t) \mod \ell_i^{e_i},$$

depending on whether a or b is invertible.