Séta: Supersingular Encryption from Torsion Attacks

Luca de Feo, Cyprien Delpech de Saint Guilhem, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, Benjamin Wesolowski

Séta: Supersingular Encryption from Torsion Attacks

Luca De Feo[1], Cyprien Delpech de Saint Guilhem[2], Tako Boris Fouotsa[3], Péter Kutas[4,5], Antonin Leroux[6,7], Christophe Petit[4,8], Javier Silva[9], and Benjamin Wesolowski[10]

[1] IBM Research Europe, Zürich, Switzerland
[2] imec-COSIC, KU Leuven, Belgium
[3] Università Degli Studi Roma Tre, Italy
[4] University of Birmingham, UK
[5] Eötvös Loránd University, Budapest, Hungary
[6] DGA
[7] LIX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris
[8] Université Libre de Bruxelles
[9] Universitat Pompeu Fabra
[10] Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400, Talence, France INRIA, IMB, UMR 5251, F-33400, Talence, France
asiacrypt21@defeo.lu, cyprien.delpechdesaintguilhem@kuleuven.be
takoboris.fouotsa@uniroma3.it, kutasp@gmail.com
antonin.leroux@polytechnique.org, christophe.f.petit@gmail.com
javiersilvavelon@gmail.com

Abstract. We present *Séta*,[11] a new family of public-key encryption schemes with post-quantum security based on isogenies of supersingular elliptic curves. It is constructed from a new family of trapdoor one-way functions, where the inversion algorithm uses Petit's so called *torsion attacks* on SIDH to compute an isogeny between supersingular elliptic curves given an endomorphism of the starting curve and images of torsion points. We prove the OW-CPA security of Séta and present an IND-CCA variant using the post-quantum OAEP transformation. Several variants for key generation are explored together with their impact on the selection of parameters, such as the base prime of the scheme. We furthermore formalise an "uber" isogeny assumption framework which aims to generalize computational isogeny problems encountered in schemes including SIDH, CSDIH, OSIDH and ours. Finally, we carefully select parameters to achieve a balance between security and run-times and present experimental results from our implementation.

1 Introduction

Isogeny-based cryptography. Recent years have seen an increasing interest in cryptosystems based on supersingular isogeny problems as appropriate candi-

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

dates for post-quantum cryptography. The latter has received greater focus due to the recent standardization process initiated by NIST.[12]

More precisely, the central problem of 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 primitives such as hash functions [10], encryption schemes [2, 23], key encapsulation mechanism (KEM)s [2] and signatures [16, 21].

Encryption schemes. The first key agreement and public-key encryption (PKE) scheme based on isogenies of ordinary elliptic curves was independently discovered by Couveignes [15] and Rostovtsev and Stolbunov [34, 38]. It 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. They then send the curves to each other and finally use their respective secrets to arrive at a common curve E_{AB}. It is then immediate to transform the key agreement into a CPA-secure PKE by following El Gamal's blueprint.

In 2011, Jao and De Feo [23] introduced SIDH, a key agreement protocol based on isogenies of supersingular curves, inspired both by the Couveignes–Rostovtsev–Stolbunov scheme and by the hash function of Charles, Goren and Lauter [10]. In the supersingular case, however, isogenies do not have a natural commutative property, meaning that, for example, the result of applying Bob's isogeny φ_B to Alice's curve E_A cannot be meaningfully defined without some extra constraints. 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.

However, the isogeny problem upon which the security of the scheme is based now differs from the original problem in certain ways. Most importantly, the adversary has access to the image of certain torsion points under a secret isogeny. Galbraith, Petit, Shani and Ti [20] were the first to exploit this extra information in an active attack showing that one cannot use static keys in SIDH. Then, two further works studied the generic problem of finding isogenies if the action of the isogeny on some torsion is known [17, 33]. These look at two different scenarios:

1. The starting curve is $E_0 : y^2 = x^3 + x$;
2. The starting curve is chosen by the adversary;

Let p be a prime number; for simplicity we restrict to supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Let A be the degree of some secret isogeny φ and let B be the order of a torsion group on which the action of φ is known. In the first case [17] gives a polynomial-time algorithm to compute φ whenever $B > \sqrt{p}A^2$. In the second case it shows how to construct special starting curves

[12] 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.

(called *backdoor curves*) for which backdoor information is known, in the form of an endomorphism of the curve, which enables a polynomial-time algorithm to compute φ whenever $B > A^2$.

In SIDH one has $A \approx B \approx \sqrt{p}$ so these algorithms do not lead to an attack. However [17] also shows that, if an adversary is allowed to choose the starting curve, then even in the SIDH setting it is possible to mount key-recovery attacks which take exponential time, yet are faster than known algorithms [17, Corollary 32]. In anticipation of potential further cryptanalysis progress, it is desirable to design alternative cryptographic protocols that rely on different isogeny problems. An example of this is the CSIDH scheme [9] (and its variants [19, 31]), a key agreement protocol that relies on the original isogeny problem, but is restricted to supersingular elliptic curves over $\mathbb{F}_p$, and can be solved in quantum subexponential time.

These results show that 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. Our main contribution is to turn the attack described in [17] into a PKE by using the special starting curves mentioned above as public keys. The associated secret key can be derived from an endomorphism of the curve with a specific minimal polynomial. More precisely, one can use any special curve whose endomorphism ring has a particular quadratic order embedded into it. Using such a starting curve, one can design a PKE where a message corresponds to an isogeny and a ciphertext contains the codomain of the isogeny together with images of the torsion points under the isogeny. Decryption is then performed using the algorithm which recovers the secret isogeny using the techniques developed in [33] and [17].

Choosing parameters for our scheme is not obvious due to the following reason. Even though trapdoor curves can be constructed in polynomial time, in practice this can be very costly. This is acceptable for a backdoor, but not for a PKE for which key generation should be routine computation. The expensive step is to generate a supsersingular elliptic curve with a prescribed endomorphism ring. We utilize techniques from SQISign [16] where one uses special primes to substantially speed up the procedure of generating starting curves. Furthermore, the worst-case complexity of torsion-point attacks is dependent on the number of prime factors of the isogeny degree. We therefore impose extra conditions on the quadratic order to avoid timing attacks that this could imply.

We also present variants for constructing backdoor curves which allow for slightly different decryption mechanisms. Namely one can either construct the starting curve directly and then compute a backdoor, or instead choose a secret backdoor curve first and then apply a secret walk to it. We discuss trade-offs between security, key size and speed in this context.

We emphasize that just knowing the equation of the starting curve and a description of the quadratic order embedded in it does not seem to be helpful without the concrete knowledge of an endomorphims realizing this embedding.

We formalize this idea in what we call the *uber isogeny problem* or $\mathfrak{O}$-UIP (Problem 5.1): suppose that one knows that a certain quadratic order $\mathfrak{O}$ is embedded in the endomorphism ring of two curves E_0, E_s, and that and that a concrete embedding of E_0 is also given in input, the problem is to find an isogeny between E_0 and E_S corresponding to a $\mathfrak{O}$-ideal. The formulation of this $\mathfrak{O}$-UIP is inspired from the key recovery problem in CSIDH [9, Problem 10]. We show that SIDH, OSIDH [12] and our PKE scheme also rely implicitly on various instances of this assumption. We also provide an analysis on the difficulty of this problem.

Finally, we present an implementation of our scheme which includes searching for an appropriate base prime and measuring key generation and encryption/decryption speeds. Written in C, our implementation reuses some of the codebase of SQISign and improves the efficiency of several steps crucial for Séta computations.

In Section 2 we recall basic properties of supersingular elliptic curves and the SIDH protocol. Furthermore, we discuss backdoor curves (which in this context we rename as trapdoor curves) in more detail. In Section 3 we introduce our one-way function and PKE Séta. In Section 4 we show how one can generate keys efficiently for Séta. In Section 5 we introduce the uber isogeny assumption, discuss its relation to other studied isogeny problems and provide some analysis of its hardness. In Section 6 we provide details of our implementation.

2 Preliminaries

We denote the computational 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. 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 $x \xleftarrow{\$} \mathcal{S}$. A probability distribution X has min-entropy $H_\infty(X) = b$ if any event occurs with probability at most 2^{-b}. 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.

2.1 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). A maximal order is an order that 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 $\mathcal{O}_0$.

2.2 Class group action on the set of supersingular curves

We briefly recall the main definitions and properties related to the class group of quadratic imaginary orders and their link with supersingular elliptic curves. We say that a curve E admits an embedding of a quadratic imaginary order $\mathfrak{O}$, if there exists a subring of $\mathrm{End}(E)$ that is isomorphic to $\mathfrak{O}$. We say this embedding is *primitive* or *optimal* if this isomorphism cannot be extended to a super-order of $\mathfrak{O}$. We write $\mathcal{E}_\mathfrak{O}$ for the set of supersingular elliptic curves admitting a primitive embedding of $\mathfrak{O}$ (up to isomorphisms). Following [12], we also call a primitive embedding of $\mathfrak{O}$ in $\mathrm{End}(E)$ an $\mathfrak{O}$-*orientation* on E. Through the usual Deuring correspondence, $\mathfrak{O}$-ideals can be identified with isogenies. For any such ideal $\mathfrak{a}$, we write $\varphi_\mathfrak{a} : E \to \mathfrak{a} \star E$ for the corresponding isogeny. The property that $\mathfrak{a} \star E \cong \mathfrak{b} \star E$ when $\mathfrak{a}$ and $\mathfrak{b}$ are in the same ideal class proves that $\star$ defines a group action of the class group $\mathrm{Cl}(\mathfrak{O})$ on $\mathcal{E}_\mathfrak{O}$. The class number $h(\mathfrak{O})$ is the cardinality of $\mathrm{Cl}(\mathfrak{O})$. In full generality, we cannot say much more on $\#\mathcal{E}_\mathfrak{O}$ than the classical Proposition 2.1.

Proposition 2.1. *Let K be a quadratic imaginary field and let $\mathfrak{O}$ be a quadratic order inside K. When p does not split in K, the number of distinct embeddings of $\mathfrak{O}$ inside maximal orders of the quaternion algebra $\mathcal{B}_{p,\infty}$ is exactly* $\mathrm{Cl}(\mathfrak{O})$. *In particular, $\#\mathcal{E}_\mathfrak{O} \leq h(\mathfrak{O})$.*

In general, Proposition 2.1 does not help in estimating $\#\mathcal{E}_\mathfrak{O}$ precisely because we do not know how to estimate the number of different embeddings of $\mathfrak{O}$ into the same maximal order in $\mathcal{B}_{p,\infty}$. We provide examples of cases where more precise properties can be stated in Sections 5.2 and 5.3.

When p splits in the field K, then $\mathcal{E}_\mathfrak{O}$ is empty (the curves admitting an $\mathfrak{O}$-orientation are ordinary). In the remaining of this article, we consider that we are never in this case to simplify the notations and statements.

Any quadratic order $\mathfrak{O}$ can be written as $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$ where $\mathfrak{O}_0$ is another quadratic order (not necessarily distinct from $\mathfrak{O}$) and f is often called the conductor of $\mathfrak{O}$. When the conductor is one, we say that the quadratic order is *maximal*. In [29], it was shown that these conductors can be tied to isogenies.

Proposition 2.2. *Let $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$ be a quadratic order and let E be a supersingular curve defined over $\mathbb{F}_{p^2}$. If E is in $\mathcal{E}_{\mathfrak{O}}$, then there exists an isogeny of degree f between E and a supersingular curve $E_0 \in \mathcal{E}_{\mathfrak{O}_0}$. Conversely, when there exists an isogeny of degree f between E and a supersingular curve $E_0 \in \mathcal{E}_{\mathfrak{O}_0}$, then E is in $\mathcal{E}_{\mathbb{Z}+f'\mathfrak{O}_0}$ for some f' dividing f.*

In Proposition 2.2, we say that the isogeny $\varphi : E_0 \to E$ of degree f is *descending* when $f' = f$. Let $\varphi : E_0 \to E$ be a descending isogeny of degree f, the embedding of $\mathfrak{O}$ in $\mathrm{End}(E)$ in Proposition 2.2 is obtained with endomorphisms of the form $[d] + \varphi \circ \alpha_0 \circ \hat{\varphi}$ with $d \in \mathbb{Z}$ and α_0 in the embedding of $\mathfrak{O}_0$ inside $\mathrm{End}(E_0)$. Similar endomorphisms are constructed in torsion point attacks against SIDH variants [27, 33], and they underlie the decryption mechanism of the Séta encryption scheme.

2.3 SIDH and SIKE

Here we give a high level description of SIDH and SIKE. We start with the original SIDH protocol of Jao and De Feo [23]. In the setup one chooses two small primes ℓ_A, ℓ_B and a prime p 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 a fixed supersingular curve, for example, assuming $p = 3 \bmod 4$, the elliptic curve with j-invariant 1728.[13] 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:

[13] Jao and De Feo do not specify a particular curve, and recommend to pick one using Bröker's algorithm [8], however there appears to be no advantage in doing so, and thus SIKE opts for $j = 1728$ for simplicity.

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 [2, 23]. In the SIKE submission [2], it is transformed using the constructions in [22, Section 3] to produce an IND-CCA secure KEM in the random oracle model (ROM).

2.4 Trapdoor curves

Let E_1, E_2 be supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let $\phi : E_1 \to E_2$ be an isogeny of degree D. First we recall the following algorithmic problem:

Problem 2.3 (SSI-T). Let D and N be smooth coprime integers. Let $\phi : E_1 \to E_2$ be a secret isogeny of degree D. Assume that we know the action of ϕ on $E_1[N]$. Compute ϕ.

Remark 2.4. The SSI-T problem is a generalization of the CSSI introduced in [23] (Problem 5.6) where D and N are prime powers of the same size.

The SSI-T problem makes sense for any D, N which are coprime and sufficiently smooth. However, in many cases the size of the input is superlinear in p thus has no practical relevance. Thus from now on we restrict to instances where the D and N-torsion are efficiently representable:

Definition 2.5. *Let N be an integer and let p be a prime number. Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. We call $E[N]$ efficiently representable if representing points in $E[N]$ requires polynomial space in $\log p = O(\lambda)$.*

Remark 2.6. In particular $E[N]$ is efficiently representable whenever N is powersmooth or N divides $p^c - 1$ for some small c. In this paper we will mainly consider instances where N is smooth and divides $p^2 - 1$.

We recall (slightly modified version of)[17, Theorem 3] how finding a certain endomorphism of E_2 relates to finding the secret isogeny ϕ:

Theorem 2.7. *Let $\phi : E_1 \to E_2$ be a secret isogeny of degree D. Assume that $E[N]$ and $E[D]$ are efficiently representable for any supersingular curve E and that the action of ϕ on $E_1[N]$ is given. Suppose furthermore, that we know $\theta \in \mathrm{End}(E_1)$ and $d, e \in \mathbb{Z}$ such that the trace of θ is 0 and $\deg(\phi \circ \theta \circ \hat{\phi} + [d]) = N^2 e$. Let M be the largest divisor of D such that $E_2[M] \subset \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_2[D]$. Let k be the number of distinct prime divisors of M. Then we can compute ϕ in time $O^*(2^k \sqrt{e})$.*

Proof. We sketch the proof of the theorem. Let $\tau = \phi \circ \theta \circ \hat{\phi} + [d]$. Then if $\ker(\tau)$ is cyclic, then $\tau = \psi' \circ \eta \circ \psi$ where $\deg(\psi) = \deg(\psi') = N$ and $\deg(\eta) = e$ and the kernels of ψ and ψ' are cyclic. In [17, Theorem 3] it is shown that $\ker(\tau)$ is always cyclic if N is odd and if N is even then $\tau = \psi' \circ \eta \circ \psi \circ [K]$ where $\deg(\psi) = \deg(\psi') = N/K$, $\deg(\eta) = e$ and $K = 1$ or $K = 2$.

Then one can compute ψ and K using the torsion point information and ψ' using the observation that $\ker(\hat{\psi'}) = \tau(E_2[B])$. The isogeny η can be computed by a meet-in-the-middle algorithm. Once τ is computed, one can compute ϕ by looking at $G = \ker(\phi \circ \theta \circ \hat{\phi}) \cap E_2[D]$. If $M = 1$ then G is cyclic and can be recomputed easily. If not, then one can use [Section 4.3][33] to recover τ. The cost of this step is $O^*(2^k)$ where k is the number of prime factors of M.

Remark 2.8. Theorem 2.7 in particular implies that one can recover ϕ in $O^*(\sqrt{e})$ whenever the number of distinct prime divisors of D (and hence M) is smaller than $\log \log p$. In Section 3.3, we introduce a condition on the quadratic order $\mathbb{Z}[\theta]$ to ensure that M is always equal to 1.

The key ingredient to Theorem 2.7 is the knowledge of θ. When $M = 1$ (which will be the case for the concrete inversion procedure in Algorithm 1), all we really need is the action of θ on $E_1[N]$. Indeed, from the sketch of proof of Theorem 2.7, we see that in that case θ is only used to compute the kernel of the two isogenies ψ and ψ' of degree N. These kernels are computed by evaluating the N-torsion $\tau = \phi \circ \theta \circ \hat{\phi} + [d]$ which can be done with the action of θ and ϕ on $E_1[N]$.

Note the action of θ on $E_1[N]$ is hard to recover from E_1 only. This motivates a notion of (D, N)-trapdoor T to encompass any kind of information that enables the computation described in the proof of Theorem 2.7.

Definition 2.9. *Let p be a prime number and let D and N be coprime smooth integers. Then a tuple (E, T) is called a (D, N)-trapdoor curve if one can use T to solve any instance of the SSI-T problem (with parameters D, N, p) with starting curve E in polynomial time. We sometimes call T the trapdoor.*

In [17] the authors introduces a polynomial-time algorithm for constructing (D, N)-trapdoor curves whenever $N > D^2$ and the number of prime divisors of $D < \log \log p$. The main idea is to reproduce the set-up of Theorem 2.7. Thus, if one can construct a supersingular elliptic curve E together with an endomorphism $\theta \in \mathrm{End}(E)$ verifying the requirements of Theorem 2.7, and compute the action of this endomorphism θ on $E[N]$, then one can solve SSI-T in polynomial time (by finding an e which is sufficiently small).

The conditions put on θ in Theorem 2.7 are essentially conditions on the minimal polynomial of θ, meaning that every trace zero element in the quaternion algebra whose norm is $\frac{B^2 e - d^2}{A^2}$ can be used as a suitable θ. This implies that potential (D, N)-trapdoor curves are obtained from curves in $\mathcal{E}_{\mathfrak{O}}$ for quadratic order $\mathfrak{O}$ of the form $\mathbb{Z}\left(\sqrt{\frac{N^2 e - d^2}{D^2}}\right)$.

We briefly sketch how θ can be generated. Since $\mathrm{Tr}(\theta) = 0$, it can be written as $ci + bj + aij$ over $\mathcal{B}_{p,\infty}$. Then the degree of τ is $D^2(p^2a + p^2b + c^2) + d^2$. Observe that a, b, c can be rational numbers but since θ is an integral element its norm $p^2a^2 + p^2b^2 + c^2$ must be an integer. So one has to find d, e such that $N^2e - d^2$ is divisible by D^2 and is positive.

This can be achieved when $N > D^2$. Let $\Delta = N^2e - d^2$. Then one has to find a rational solution to the equation $p^2a^2 + p^2b^2 + c^2 = \Delta$, which exists whenever Δ is a quadratic residue modulo p (if that is not the case one chooses a different d and e). A solution can be found using Denis Simon's algorithm [37]. From there, we can find a maximal order $\mathcal{O}$ containing θ and then compute a supersingular elliptic curve whose endomorphism ring is isomorphic to $\mathcal{O}$ (see Algorithm 3 in Section 4.2). After that, the action of θ on the N-torsion can be found using an explicit representation of $\mathcal{O}$. All these operations can be done in polynomial time (see Algorithms 2 and 3 for more details), leading to the following theorem:

Theorem 2.10. *Let p be a prime number and let D and N be smooth coprime integers such that $N > D^2$ and the number of distinct prime divisors of D is smaller than $\log\log p$. Then there exists a polynomial-time algorithm which outputs a (D, N)-trapdoor curve E with the following information:*

- *The j-invariant of E.*
- *Integers d, e with $e = O(\log(p))$.*
- *A basis P, Q of $E[N]$ and the points $\theta(P), \theta(Q)$ for a trace 0 endomorphism θ such that $\deg([D]\theta + [d]) = N^2e$.*

3 Séta trapdoor one way function and public key encryption scheme

In this section we describe a general trapdoor one-way function where the main idea is to turn the attacks from [17] into a trapdoor mechanism.

We first generalise the CGL hash function and we describe a trapdoor subfamily of this generalization. We then provide more details on key generation, evaluation and inversion. We finally describe the Séta public key encryption scheme and its CCA version.

3.1 Generalised Charles-Goren-Lauter hash function

We generalise the CGL hash function family introduced in [10]. To select a hash function from this family, one selects a j-invariant $j \in \mathcal{J}_p$ which canonically fixes a curve $E/\mathbb{F}_{p^2}$ with $j(E) = j$. There are $\ell + 1$ isogenies of degree ℓ connecting E to other vertices. These $\ell + 1$ vertices can be ordered in a canonical way and a canonical one of them can be ignored. Then, given a message $m = b_1b_2 \dots b_n$, 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. The output of the CGL hash function h_j is then the j-invariant of the final curve in the path, i.e. $h_j(m) \coloneqq j(E_n)$, where the walk starts at vertex j and is defined as above. We see that starting at a different vertex j' results in a different hash function $h_{j'}$.

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 inputs $m = b_1 b_2 \dots b_n$ where $b_1 \in [\ell+1]$ and $b_i \in [\ell]$ for $2 \leq i \leq n$; this introduces a one-to-one correspondence between inputs and cyclic isogenies of degree ℓ^n originating from E.

Secondly, we consider a generalisation where the walk takes place over multiple graphs G_{ℓ_i}. Given an integer $D = \prod_{i=1}^{n} \ell_i^{e_i}$ where the ℓ_i are prime factors, we introduce the notation $\mu(D) \coloneqq \prod_{i=1}^{n} (\ell_i+1) \cdot \ell_i^{e_i-1}$. We then take the message m to be an element of

$$[\mu(D)] = \left\{ (m_1, \dots, m_n) \;\middle|\; \begin{array}{l} m_i = b_{i1} b_{i2} \dots b_{ie_i}, b_{i1} \in [\ell_i+1], b_{ij} \in [\ell_i] \\ \text{for } 2 \leq j \leq e_i, \text{ for } 1 \leq i \leq n \end{array} \right\}$$

where each m_i is hashed along the graph G_{ℓ_i}. To ensure continuity, the j-invariants are chained along the hash functions, that is, we write $j_i = h_{j_{i-1}}(m_i)$, where j_{i-1} is the hash of m_{i-1}. Thus, only $j = j_0$ parameterizes the overall hash function. As before, this generalization returns the final j-invariant $j_n = h_{j_{n-1}}(m_n)$ as the hash of m.

Thirdly, we also modify the CGL hash function to return the images of two canonically defined torsion points P_j and Q_j of order N under the D-isogeny $\varphi_m : E_j \to E_{j_n}$.

We call the resulting hash function family *generalized CGL* or G-CGL, and we denote it by $\mathcal{H}^{p,D,N}$, namely

$$\mathcal{H}^{p,D,N} = \left\{ h_j^{D,N} : m \mapsto (j(E_n), \varphi_m(P_j), \varphi_m(Q_j)) \mid j \in \mathcal{J}_p \right\}.$$

3.2 A trapdoor function family from the G-CGL family

Given p, D and N, let $\mathcal{J}_{T,p} \subset \mathcal{J}_p$ be the set of j-invariants of (D,N)-trapdoor curves defined over $\mathbb{F}_{p^2}$ (see Definition 2.9). By definition of a trapdoor curve, for any $j_T \in \mathcal{J}_{T,p}$, the hash function $h_{j_T}^{D,N}$ can be inverted using the trapdoor information. We hence obtain the following family of trapdoor functions:

$$\mathcal{F}_T^{p,D,N} = \left\{ f_{j_T}^{D,N} : m \mapsto (j(E_n), \varphi_m(P_{j_T}), \varphi_m(Q_{j_T})) \mid j_T \in \mathcal{J}_{T,p} \right\},$$

where $f_{j_T}^{D,N} \coloneqq h_{j_T}^{D,N}$.

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

Lemma 3.1. *Let $N^2 > 4D$. Then for any $j_T \in \mathcal{J}_{T,p}$, $f_{j_T}^{D,N}$ is injective.*

Proof. Let $N^2 > 4D$ and $j_T \in \mathcal{J}_{T,p}$, suppose that a function $f_{j_T}^D$ is not injective, i.e. that there are two distinct isogenies φ and φ' of degree D from E_{j_T} to E_c, corresponding to two distinct messages, with the same action on $E_{j_T}[N]$, implied by the colliding images of P_{j_T} and Q_{j_T}. Then, following [30, Section 4], their difference is also an isogeny between the same curves whose kernel contains the entire N-torsion. This, together with [36, Lemma V.1.2], implies that $4D \geq \deg(\varphi - \varphi') \geq N^2$. Taking $N^2 > 4D$ ensures that in fact $\varphi = \varphi'$ and therefore that $f_{j_T}^{D,N}$ is injective. □

One-wayness. One-wayness of our function family relies on Problem 3.2 below. This problem is a variant of the CSSI problem introduced in [23], with the difference that the starting j-invariant is chosen at random from $\mathcal{J}_{T,p}$ (instead of being fixed) and only the min-entropy of the distribution is specified.

Problem 3.2 (Trapdoor computational supersingular isogeny (TCSSI) problem). Given p and integers D and N, let j_T be a uniformly random element of $\mathcal{J}_{T,p}$ and $\varphi_m : E_{j_T} \to E_m$ be a random isogeny of degree D sampled from a distribution X with min-entropy $H_\infty(X) = O(\lambda)$. Let $\{P_{j_T}, Q_{j_T}\}$ be a basis of the torsion group $E_{j_T}[N]$. Given $E_{j_T}, P_{j_T}, Q_{j_T}, E_m, \varphi_m(P_{j_T})$ and $\varphi_m(Q_{j_T})$, compute φ_m.

Lemma 3.3. *Let j_T be a uniformly random element of $\mathcal{J}_{T,p}$. Then the function $f_{j_T}^{D,N} \in \mathcal{F}_T^{p,D,N}$ is (quantum) one-way under the (quantum) hardness of Problem 3.2.*

Proof. It is easy to check that the distribution of isogenies resulting from hashing a uniform $m^* \xleftarrow{\$} [\mu(D)]$ has the required entropy; hence the reduction is immediate. □

3.3 Inversion

In this section, we concretely show how to use methods from [17] to invert a given function $f_{j_T}^{D,N} \in \mathcal{F}_T^{p,D,N}$ with trapdoor information T. We assume that D is odd and that $\gcd(D, N) = 1$. We take E_{j_T} a supersingular curve inside $\mathcal{E}_{\mathfrak{O}}$ where $\mathfrak{O}$ is the quadratic order $\mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}]$ for some integers d, e. We write θ for the endomorphism of $\mathrm{End}(E_{j_T})$ such that $\mathbb{Z}[\theta] \cong \mathfrak{O}$. Let us also take a basis P_{j_T}, Q_{j_T} of $E_{j_T}[N]$. If we define T as $e, d, P_{J_T}, Q_{j_T}, \theta(P_{j_T}), \theta(Q_{j_T})$, then E_{j_T}, T is a (D, N)-trapdoor curve as produced in Theorem 2.10.

To make the inversion mechanism efficient on all inputs, we require the additional condition that the discriminant Δ of $\mathfrak{O}$ is a quadratic nonresidue modulo every prime divisor of D. The concrete statement can be found in Lemma 3.4. We explain how to generate $E_{j,T}$, $\mathfrak{O}$ and T in Sections 4.1 and 4.2. We are given (j_m, P_m, Q_m) as the output of $f_{j_T}^{D,N}$ for some input m, which we want to recover. Let the isogeny corresponding to m be denoted by ϕ_m. We assume that $P_m = \phi_m(P_{j_T})$ and $Q_m = \phi_m(Q_{j_T})$. Let $\tau := \phi_m \circ \theta \circ \hat{\phi}_m + [d]$ and let $G := \ker(\tau - [d]) \cap E_m[D]$.

Algorithm 1 Computing inverses

Require: $j_T \in \mathcal{J}_{T,p}$, a trapdoor T and c.
Ensure: $m \in [\mu(D)]$ such that $f_{j_T}^{D,N}(m) = \mathsf{c}$.
1: Parse c as $(j_m, P_m, Q_m) \in \mathbb{F}_{p^2} \times (\overline{\mathbb{F}_{p^2}})^2 \times (\overline{\mathbb{F}_{p^2}})^2$.
2: Parse T as $e, d, P_{J_T}, Q_{j_T}, \theta(P_{j_T}), \theta(Q_{j_T})$.
3: Compute the canonical curve E_m having j-invariant j_m.
4: Let $\tau = \phi_m \circ \theta \circ \hat{\phi}_m + [d] \in \mathrm{End}(E_m)$. ▷ Choices of θ and d ensure $\deg \tau = N^2 e$.
5: Compute τ as described in the proof of Theorem 2.7.
6: Compute $\ker(\phi_m \circ \theta \circ \hat{\phi}_m) \cap E_m[D] = \ker(\tau - [d]) \cap E_m[D] = \ker(\hat{\phi_m})$.
7: Compute $\ker(\phi_m)$ using $\ker(\hat{\phi}_m)$.
8: **return** $m \in [\mu(D)]$ that corresponds to $\ker(\phi_m)$.

Lemma 3.4. *If $\Delta = \mathrm{Disc}\ \mathfrak{O}$ is a non-quadratic residue, the group G is cyclic and equal to $\ker(\hat{\phi})$.*

Proof. It is clear that $\ker(\hat{\phi_m}) \subset G$ since it is contained in $\ker(\phi_m \circ \theta \circ \hat{\phi_m})$ 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_m can be decomposed as $\phi_{D/M} \circ \phi_M$. Then by [33, Lemma 5] the kernel of ϕ_M is fixed by θ. In the proof of [33, Lemma 6] it is shown that a subgroup of $E_{j_T}[M]$ can only be fixed by an endomorphism θ if $\mathrm{Tr}(\theta)^2 - 4\deg(\theta) = \mathrm{Disc}\ \mathbb{Z}[\theta] = \Delta$ is a square modulo M. Thus, the quadratic residuosity condition on Δ 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 $\ker(\hat{\phi_m})$ which is a group of order D. This implies that $G = \ker(\hat{\phi_m})$. □

The group $G = \ker(\hat{\phi})$ can be computed by solving a double discrete logarithm problem, which is efficient as D is smooth. We summarize the steps needed for inverting the one-way function in Algorithm 1.

In [17] it is shown that Algorithm 1 runs in polynomial time whenever $E_m[D]$ is efficiently representable and $\Delta = \mathrm{Disc}\ \mathbb{Z}[\theta]$ is as in Lemma 3.4.

3.4 Séta Public Key Encryption

We now build Séta, a Public Key Encryption scheme using the trapdoor one-way function family of Section 3.2, and we show that it is OW-CPA secure. Concretely, we define the Séta 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 be a prime such that $p^2 - 1 = DNf$ where D, N are smooth integers and f is a small co-factor such that $2^{2\lambda} < D$, $D^2 < N$. We let $\mathsf{params} = (\lambda, p, D, N)$.

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

1. Compute a uniformly random (D, N)-trapdoor supersingular elliptic curve (E_{j_T}, T) defined over $\mathbb{F}_{p^2}$ using Algorithms 2 and 3 (see Section 4).
2. Set $\mathsf{pk} \coloneqq (j_T)$ and $\mathsf{sk} \coloneqq T$.
3. 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 $n_m = \lfloor \log_2 \mu(D) \rfloor$, first cast m as an integer in the set $[\mu(D)]$ and then:

1. Parse $\mathsf{pk} = j_T \in \mathcal{J}_{T,p}$.
2. Compute $(j_m, P_m, Q_m) \leftarrow f_{j_T}^{D,N}(m)$.
3. Return $\mathsf{c} = (j_m, P_m, Q_m)$.

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

1. Given $\mathsf{params}, \mathsf{sk}$ and 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 $\tilde{m} \in [\mu(D)]$; if this fails, set $\tilde{m} = \perp$.
3. If $\perp$ was recovered, return $\perp$.
4. Otherwise, from $\tilde{m} \in [\mu(D)]$, recover $m \in \{0,1\}^{n_m}$ and return it.

Theorem 3.5. *Let p be a prime, let D and N be integers such that $D^2 < N$. Suppose that the output distribution of Algorithm 3 is statistically close to uniform. Let E_{j_T} be an output of Algorithm 3. If Problem 3.2 with p, D, N, E_{j_T} and X such that $H_\infty(X) = \lambda$ is hard for quantum PPT adversaries, then the PKE scheme above is quantum one-way chosen-plaintext attack (OW-CPA) secure.*

Proof. Let $\mathcal{M} = \{0,1\}^{n_m}$ denote the message space of the encryption scheme, with $n_m = O(\lambda)$. We see that a randomly sampled $m \xleftarrow{\$} \mathcal{M}$ directly embedded as an integer $m \in [\mu(D)]$ yields a distribution Y with min-entropy $H_\infty(Y) \geq \lambda$ on isogenies of degree D starting from E_{j_T}. The challenge of opening a given ciphertext c then reduces to recovering the secret isogeny of Problem 3.2 with $X = Y$. □

3.5 IND-CCA encryption scheme

We obtain an IND-CCA secure PKE scheme by applying the generic post-quantum OAEP transformation [39, Section 5] (see Appendix B) to Séta, for which we prove that our function $f_{j_T}^{D,N}$ is quantum partial-domain one-way.

Definition 3.6. *Let k_1, k_0 and n_c be integers. A family $\mathcal{F}$ of functions $f : \{0,1\}^{\lambda+k_1} \times \{0,1\}^{k_2} \to \{0,1\}^{n_c}$ is* partial domain one-way *if for any polynomial time adversary $\mathcal{A}$, the following advantage is negligible in λ:*

$$\mathrm{Adv}_\lambda(\mathcal{A}) = \Pr\left[s' = s; s' \leftarrow \mathcal{A}(1^\lambda, y), y \leftarrow f(s,t), (s,t) \xleftarrow{\$} A \times B, f \leftarrow \mathcal{F}\right]$$

Lemma 3.7. *Let j_T be a uniformly random element of $\mathcal{J}_{T,p}$. The function $f_{j_T}^{D,N}$ defined in Section 3.2 is a quantum partial-domain one-way function, under the hardness of Problem 3.2.*

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 as $D_1 \cdot D_2$ such that $\gcd(D_1, D_2) = 1$, $2^{\lambda+k_1} \leq \mu(D_1)$ and $2^{k_0} \leq \mu(D_2)$ (where $\lambda + k_0 + k_1$ is the bit-length of input strings) and then embed each of s and t into $\mu(D_1)$ and $\mu(D_2)$ respectively. Then we can set $f_{j_T}^{D,N}(s,t) := f_{j_1}^{D_2,N}(t)$ where $(j_1, P_1, Q_1) = f_{j_T}^{D_1,N}(s)$ and $f_{j_1}^{D_2,N}$ uses $\{P_1, Q_1\}$ as basis of $E_{j_1}[N]$. Since $2^{\lambda+k_1} \leq \mu(D_1)$, then recovering s from $y = f_{j_T}^{D,N}(s,t)$ is hard under the same assumption as Theorem 3.5 with D replaced by D_1. □

Theorem 3.8 ([39], Theorem 2). *If $f_{j_T}^{D,N}$ is a quantum partial-domain one-way function, then the OAEP-transformed scheme is IND-CCA secure in the quantum random oracle model (QROM).*

4 Key generation variants

In this section we describe various methods for generating keys for Séta. We first describe Algorithm 2, which can generate integers d, e so that $\Delta = \operatorname{Disc} \mathfrak{O}$, where $\mathfrak{O} = \mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}]$, satisfies the quadratic residuosity conditions imposed Section 3.3. Then, we present two options for generating a uniformly random supersingular elliptic curve inside $\mathcal{E}_{\mathfrak{O}}$ together with the remaining part of the trapdoor information T. Algorithm 3 treats the generic case, and Algorithm 4 focuses on computing a (DD_s, N)-trapdoor curve from a (D, N)-trapdoor curve and a random walk of degree D_s.

4.1 Computing the trapdoor information

We recall that the required condition is that $\Delta = \operatorname{Disc} \mathfrak{O} = -4\frac{N^2e-d^2}{D^2}$ must be negative and a quadratic non-residue modulo every prime dividing D and also modulo p. For simplicity, we fix $e = 1$ and look for d of a special form. This is described in Algorithm 2.

Lemma 4.1. *If d, e is the output of Algorithm 2, then $\frac{N^2e-d^2}{D^2}$ is a quadratic non-residue modulo all ℓ_i.*

Proof. Let r_i, s_{ℓ_i}, T 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 $\frac{-N^2e+(D^2r+u)^2}{D^2}$ is not a quadratic residue modulo ℓ_i which implies that $-\frac{N^2e-d^2}{D^2}$ 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

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

Algorithm 2 Computing the integers d, e

Require: D, N, p as above. Let S be the product of primes dividing D.

Ensure: (d, e) such that $-\frac{N^2e-d^2}{D^2} < 0$ is a quadratic non-residue modulo every prime dividing D and is a quadratic non-residue modulo p.

1: Set $e = 1$.
2: Find u such that $u^2 \equiv N^2 e \pmod{D^2}$.
3: **for** every prime ℓ_i dividing D **do**
4: Let s_{ℓ_i} be a quadratic non-residue modulo ℓ_i.
5: $r_i \leftarrow (s_{\ell_i} - \frac{-N^2e+u^2}{D^2})(2u)^{-1} \pmod{\ell_i}$.
6: Compute a residue r modulo S with the property that $r \equiv r_i \pmod{\ell_i}$.
7: $\ell \leftarrow 0$.
8: $d \leftarrow D^2(S\ell + r) + u$.
9: $A \leftarrow \frac{N^2e-d^2}{D^2}$.
10: **if** $A < 0$ **then**
11: **return** $\perp$
12: **if** A is not a square modulo p **then**
13: $\ell \leftarrow \ell + 1$.
14: **go to** Step 8.
15: **return** (d, e)

By our choice of r we have that

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

which is a quadratic nonresidue by the choice of s_{ℓ_i}. □

Lemma 4.2. *Let S be the product of all primes dividing D. If $N > D^2S$, then Algorithm 2 returns a correct pair (d, e) with probability higher than $1-2^{-\frac{N}{SD^2}+1}$ under plausible heuristic assumption.*

Proof. Since u is found by solving an equation modulo D^2, we obtain $u < D^2$. Similarly we have $r < S$. Under plausible heuristic assumptions, we can estimate to $1/2$ the probability that the quadratic reduosity condition on A is satisfied. Thus, we obtain a bound on the failure probability by counting how many values ℓ can be tried before A becomes negative. With the conservative bound that $D^2r + u \approx D^2S$, we obtain that we can try $\frac{N-D^2S}{DS^2}$ different values for small d, which gives the result.

Correctness of the result follows from Lemma 4.1.

4.2 Trapdoor curve generation

Now we focus on generating a random supersingular elliptic curve whose endomorphism ring contains an embedding of $\mathfrak{O} = \mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}$ for d, e outputs of Algorithm 2. In [17, Section 5.1] it is discussed how one can generate

a specific curve inside $\mathcal{E}_{\mathfrak{O}}$. Essentially, this is achieved by computing a maximal order $\mathcal{O}$ containing the suborder $\mathfrak{O}$ (with [42, Algorithm 7.9]) and then computing a supersingular elliptic curve whose endomorphism ring is isomorphic to $\mathfrak{O}$ (with [18, Algorithm 12]). This procedure can be made concretely efficient with the algorithms from [16] under some conditions on the prime p that partly underlie the choice of prime described in Section 6.2. However, this procedure is essentially deterministic, so an adversary knowing the quadratic order $\mathfrak{O}$ can just recompute the same trapdoor curve. The point of this subsection is to show how to randomize the procedure.

We obtain randomization by first generating a curve with the deterministic procedure and then applying the action of a random class group element to derive another random curve with the same embedding. This operation would be costly if it required to compute a lot of isogenies. However, we can do it over the quaternions at a negligible cost before applying the translation algorithm from maximal orders to elliptic curves.

For concrete randomization, we use the fact (see [24]) that there exists a bound B (polynomial in p) for which the graph whose vertices are curves in $\mathcal{E}_{\mathfrak{O}}$ and edges are isogenies of prime degree smaller than B is an expander graph. The fast mixing property of expander graphs implies that the distribution of curves obtained after a random walk of fixed length quickly converges to the uniform distribution as the length of the walk grows. More precisely, for any δ we can find a length ε (logarithmic in the size of the graph and δ) for which the statistical distance between the random walk distribution and the uniform distribution is less than δ. So once the length ε (corresponding to a sufficiently small δ) has been set, for any starting curve E_0 in $\mathcal{E}_{\mathfrak{O}}$ the curve $\prod_{i=1}^{n} \mathfrak{l}_i^{\varepsilon_i} \star E_0$ where $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ are prime ideals above the n prime $\ell_1, \ldots, \ell_n$ smaller than B that are split in $\mathfrak{O}$ and $(\varepsilon_1, \ldots, \varepsilon_n)$ is uniformly random among the vectors in $\mathbb{Z}^n$ such that $\sum_{i=1}^{n} |\varepsilon_i| = \varepsilon$, is statistically close to a uniformly random element in $\mathcal{E}_{\mathfrak{O}}$. This result underlies Algorithm 3.

Proposition 4.3. *Algorithm 3 is correct and terminates in polynomial time.*

Proof. All the sub-algorithms run in polynomial-time and by choice of B and ε, the number of iterations in the loop is also polynomial.

It is easy to verify that the ideal I corresponds through the Deuring correspondence to the isogeny $\varphi_{\mathfrak{l}_i}$. Thus, our method simulates a random walk over the graph that we described at the beginning of this section. For the reasons explained there, the curve E_{j_T} obtained in the end is statistically close to a random element in $\mathcal{E}_{\mathfrak{O}}$. □

4.3 Constraints on the prime

In Séta, we compute and evaluate isogenies of degree D and N. Hence we always require that D and N are smooth and that the DN-torsion groups are efficiently representable, i.e., that they are defined on extensions of $\mathbb{F}_{p^2}$ of small degree. For example, if we require that $E[DN] \subset E(\mathbb{F}_{p^4})$, then DN must divide $p^2 - 1$.

Algorithm 3 Generating the trapdoor curve from a quadratic order $\mathfrak{O}$

Require: A prime p, an integer N, a quadratic order $\mathfrak{O}$, a bound B, a length ε.

Ensure: A uniformly random curve $E_{j_T} \in \mathcal{E}_{\mathfrak{O}}$, a basis P_{j_T}, Q_{j_T} of $E_{j_T}[N]$, and $\theta(P_{j_T}), \theta(Q_{j_T})$ with $\theta \in \mathrm{End}(E_{j_T})$ such that $\mathbb{Z}[\theta] \cong \mathfrak{O}$.

1: Find a max. order $\mathcal{O} \subset \mathcal{B}_{p,\infty}$ with $\mathfrak{O}$ embedded in $\mathcal{O}$ with the alg. from [17].
2: Compute $\ell_1, \ldots, \ell_n$ the n primes split in $\mathfrak{O}$ smaller than B.
3: Select a random vector $(\varepsilon_1, \ldots, \varepsilon_n)$ in $\mathbb{Z}^n$ with L_1 norm equal to ε.
4: Set $\mathcal{O}_{j_T} = \mathcal{O}$.
5: **for** $1 \leq i \leq n$ **do**
6: Compute $\alpha_i \in \mathfrak{O}$ such that $\mathfrak{l}_i = \mathfrak{O}\langle \alpha_i, \ell_i \rangle$ is a prime ideal above ℓ_i.
7: **for** $1 \leq j \leq |\varepsilon_i|$ **do**
8: Compute the ideal $I = \mathcal{O}_{j_T}\langle \alpha_i, \ell_i \rangle$.
9: Set $\mathcal{O}_{j_T}$ as the right order of I.
10: Compute the curve E_{j_T} from $\mathfrak{O}_{j_T}$ with [18, Algorithm 12].
11: Compute a canonical basis P_{j_T}, Q_{j_T} of $E_{j_T}[N]$.
12: Select the correct element $\theta \in \mathcal{O}_{j_T}$ such that $\mathfrak{O} \cong \mathbb{Z}[\theta]$.
13: Use the representation of $\mathcal{O}_{j_T}$ obtained from the execution of [18, Algorithm 12] to compute $\theta(P_{j_T}), \theta(Q_{j_T})$.
14: **return** $E_{j_T}, P_{j_T}, Q_{j_T}$ of $E_{j_T}[N], \theta(P_{j_T}), \theta(Q_{j_T})$.

The smoothness bound B_1 of D impacts the efficiency of encryption and the smoothness bound B_2 of N impacts the efficiency of decryption. For a given security level λ, we require $2^{2\lambda} < D$ in order to protect the scheme against the meet-in-middle attack.

Since we have the range $D^2 < D^2S < D^3$ depending on the value of S (product of primes dividing D), and that Lemma 4.2 implies that $N > D^2S$ then we can estimate that the value DN will be between $2^{6\lambda}$ and $2^{8\lambda}$. If we want DN dividing $p^2 - 1$, we can estimate that the minimum size for the prime p will be between 3λ and 4λ bits. The actual size will depend on the size of $(p^2 - 1)/DN$.

Besides encryption and decryption, key generation also restricts the types of primes to be used in Séta. Indeed, Step 10 and Step 13 of Algorithm 3 use [18, Algorithm 12], which in turn uses the KLTP Algorithm [26]. Although this algorithm runs in polynomial time, it is not practical in general; the variant introduced in [16] achieves much greater efficiency, provided that $p^2 - 1$ is of the form $p^2 - 1 = l^f N_2 f_2$, where ℓ is a small prime, $N_2 > p^{3/2}$ is a smooth integer co-prime to ℓ and f_2 is a cofactor. We refer to [16, §8] for more details; a concrete method to select Séta-friendly primes is described in Section 6.2.

4.4 Alternative key generation

We describe an alternative method for computing trapdoor curves and suggest a variant of the key generation algorithm for Séta. The main idea is to perform a

Algorithm 4 Computing a (D, N)-trapdoor curve from a (D_sD, N)-trapdoor curve where $D_s \approx 2^{2\lambda}$ is a smooth integer

Require: a (D_sD, N)-trapdoor curve (E_{j_T}, T) where $T = (\theta(P_{j_T}), \theta(Q_{j_T}), d, e)$.
Ensure: a (D, N)-trapdoor curve (E_s, T').
1: Sample a uniformly random isogeny $\phi_s : E_{\theta,j} \to E_s$ of degree D_s .
2: Compute $T' = (\theta'(P_s), \theta'(Q_s), d, e)$ where $\theta' = \phi_s \circ \theta \circ \widehat{\phi_s}$ and $\{P_s, Q_s\}$ is a canonical basis of $E_s[N]$..
3: **return** (E_s, T')

random secret walk from a publicly available trapdoor curve. The method relies on the following proposition.

Proposition 4.4. *Let p be a prime, let D_s, D and N be three smooth integers. Let (E_{j_T}, T) where $T = (\theta(P_{j_T}), \theta(Q_{j_T}), d, e)$ be a (D_sD, N)-trapdoor curve. Let $\phi_s : E_{j_T} \to E_s$ be an isogeny of degree D_s. Set $T' = (\theta'(P_s), \theta'(Q_s), d, e)$ where $\theta' = \phi_s \circ \theta \circ \widehat{\phi_s}$ and $\{P_s, Q_s\}$ is a canonical basis of $E_s[N]$. Then (E_s, T') is a (D, N)-trapdoor curve.*

Proof. Since we know the action of θ on the torsion group $E_{j_T}[N]$ and ϕ_s, then we can efficiently evaluate $\theta' = \phi_s \circ \theta \circ \widehat{\phi_s}$ on $E_s[N]$. Since (E_{j_T}, T) is a (D_sD, N)-trapdoor curve, then $\mathrm{Tr}(\theta) = 0$ and $\widehat{\theta} = -\theta$. Hence

$$\mathrm{Tr}(\theta') = \phi_s \circ \theta \circ \widehat{\phi_s} + \widehat{\phi_s \circ \theta \circ \widehat{\phi_s}} = \phi_s \circ \theta \circ \widehat{\phi_s} - \phi_s \circ \theta \circ \widehat{\phi_s} = 0.$$

It follows that

$$\deg([D]\theta' + [d]) = D^2 \deg(\theta') + d^2 = D^2 D_s^2 \deg(\theta) + d^2 = N^2 e.$$

By Theorem 2.10, (E_s, T') is a (D, N)-trapdoor curve. □

Relying on Proposition 4.4, Algorithm 4 computes (D, N)-trapdoor curves when given a (D_sD, N)-trapdoor curve.

Lemma 4.5. *Algorithm 4 is correct and runs in polynomial time.*

Proof. The correctness of Algorithm 4 follows from Proposition 4.4. Step 1 of Algorithm 4 consists of a degree D_s isogeny computation. Since D_s is smooth, then Step 1 runs in polynomial time. Step 2 consists of an evaluation of $\phi_s \circ \theta \circ \widehat{\phi_s}$ on P_s and Q_s. One evaluate $\widehat{\phi_s}(P_s)$ and express it as a linear combination of P_{j_T} and Q_j to recover $\theta\left(\widehat{\phi_s}(P_s)\right)$, then on evaluates $\phi_s\left(\theta\left(\widehat{\phi_s}(P_s)\right)\right)$. Similarly, one evaluates $\phi_s\left(\theta\left(\widehat{\phi_s}(Q_s)\right)\right)$. All these steps run in polynomial time since D_s and N are smooth integers.

A variant of the Séta setup and key generation is described as follows.

Parameters. Let λ denote the security parameter. Let p be a prime such that $p^2-1 = D_sDNf$ where D_s, D, N are smooth integers and f is a small co-factor such that $2^{2\lambda} < D \approx D_s$, $D_s^2D^2 < N$. Compute a (D_sD, N)-trapdoor curve (E_{j_T}, T) using Algorithm 3. We let $\mathsf{params} = (\lambda, p, D_s, D, N, E_{j_T}, T)$.

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

1. Compute a random (D, N)-trapdoor curve (E_s, T') using Algorithm 4 with (E_{j_T}, T) as input.
2. Set $\mathsf{pk} \coloneqq (j_s)$ and $\mathsf{sk} \coloneqq T'$.
3. Return $(\mathsf{pk}, \mathsf{sk})$.

The advantage of this variant is the fact the key generation algorithm does not use Algorithm 3, hence most of the requirements on p enumerated in Section 4.3 can be relaxed. This implies having more freedom in the choice of D and N, for which we could opt for powers of very small primes. Mostly, less good SQISign primes would be admissible for this variant, which is not the case in the original Séta described in Section 3.4, since its key generation uses Algorithm 3 which requires good Séta primes in order to be practically efficient. This variant is hence a good alternative to the Séta key generation, given the fruitless search of good cryptographic size SQI-Sign primes.

On the other hand, using less good SQISign primes implies that generating the (D_sD, N)-trapdoor curve (E_{j_T}, T) in the parameters generation is less efficient. But since this parameter generation is run once and for all, then this does not constitute a considerable drawback.

The main drawback of this key generation method is the considerably large size of the base prime p. In fact, p needs to satisfy $p^2 - 1 = D_sDNf$ where f is a small co-factor, and $D_s \approx D \approx 2^{2\lambda}$ such that attacking the isogeny $\phi_s : E_{j_T} \to E_s$ or $\phi_m : E_s \to E_m$ are equivalent with respect to the meet in the middle attack. Considering the fact that $N > (D_sD)^2$, then $N > 2^{8\lambda}$ and $2^{12\lambda} < D_sDN \leq p^2 - 1$, as opposed to $2^{6\lambda} < ND < p^2 - 1$ in Séta (see Section 4.3). It follows that the bit size of $p^2 - 1$ practically doubles when we use Algorithm 4 for key generation.

5 "Uber" isogeny assumption

In this section, we introduce a generic framework, which we label *Uber Isogeny assumption* in analogy to [7], aiming at generalizing isogeny computation problems encountered in the main families of isogeny-based schemes such as SIDH [23], CSIDH [9], OSIDH [12] and Séta (presented in this work).

The uber isogeny problem does not directly underlie the security of these various schemes (in the sense that no formal reduction is yet known). However, for each of these protocols there exists a set of parameters for which if one can solve the uber isogeny problem, then one can break the scheme. At a higher-level, our new problem can be seen as a generic key recovery problem.

By introducing this new assumption our goal is twofold. First, we highlight the proximity between the various isogeny schemes and we provide a common

target for cryptanalysis. Second, the generic attack that we describe in Section 5.3 gives a lower-bound on the security of any future scheme whose security may be related to our uber assumption in a similar manner as SIDH, CSIDH, OSIDH and Séta.

5.1 The new generic problem

The principal mathematical structure behind the uber isogeny problem is the group action at the heart of the CSIDH protocol and all the following works. In the isogeny setting, these group actions emerge through class groups of quadratic orders. The main definitions and properties were introduced in Section 2.2.

Problem 5.1 ($\mathfrak{O}$-Uber Isogeny Problem ($\mathfrak{O} - UIP$)). Let $p > 3$ be a prime and let $\mathfrak{O}$ be a quadratic order of discriminant Δ. Given $E_0, E_s \in \mathcal{E}_{\mathfrak{O}}$ and an explicit embedding of $\mathfrak{O}$ into $\mathrm{End}(E_0)$ (i.e the knowledge of $\alpha_0 \in \mathrm{End}(E_0)$ such that $\mathbb{Z}[\alpha_0] \cong \mathfrak{O}$), find a powersmooth ideal $\mathfrak{a}$ of norm coprime with Δ such that $[\mathfrak{a}] \in \mathrm{Cl}(\mathfrak{O})$ is such that $E_s \cong \mathfrak{a} * E_0$.

Remark 5.2. In Problem 5.1, the powersmoothness condition on the norm is to ensure that the resulting isogeny can always be computed in polynomial time. In some special cases where the form of the prime p enables to compute some smooth isogenies in polynomial time, this condition might be relaxed a little bit.

5.2 Relation with various isogeny-based constructions

We start with the link with CSIDH [9] which is quite obvious. We state the CSIDH key recovery problem below [9, Problem 10].

Problem 5.3. Given two supersingular elliptic curves E, E_0 defined over F_p with the same F_p-rational endomorphism ring $\mathfrak{O}$, find an ideal $\mathfrak{a}$ of $\mathfrak{O}$ such that $[\mathfrak{a}] \star E = E_0$. This ideal must be represented in such a way that the action of $\mathfrak{a}$ on any curve can be evaluated efficiently, for instance a could be given as a product of ideals of small norm.

Proposition 5.4. *When $p = 3 \bmod 4$ and $\Delta = -4p$, Problem 5.1 is equivalent to the CSIDH key recovery Problem 5.3.*

Proof. In the case of CSIDH, the curves admitting an embedding of $\mathbb{Z}[\sqrt{-p}] \cong \mathbb{Z}[\pi]$ in their endomorphism rings are the curves defined over $\mathbb{F}_p$ (i.e left stable by π the Frobenius morphism). Then, it is quite clear that Problem 5.1 is equivalent to Problem 5.3.

The OSIDH protocol [12] is a generalization of CSIDH where $\mathbb{Z}[\pi]$ is replaced by a larger class of quadratic orders. The link between OSIDH and Problem 5.1 is also straightforward. Let us fix some notations[14] for this protocol and briefly

[14] These notations do not exactly agree with the ones introduced in [12] because we want to hightlight the link with our $\mathfrak{O}$-IOP.

recall the principle. The OSIDH key exchange protocol starts from a descending chain of ℓ-isogenies of size n that we write $\varphi_0 : F_0 \to E_0$ where F_0 admits a $\mathfrak{O}_0$-orientation (i.e an embedding of $\mathfrak{O}_0$ inside $\mathrm{End}(E_0)$). From there, φ_0 induces an $\mathfrak{O}$-orientation on E_0. The secret keys of Alice and Bob are $\mathfrak{O}$-ideals $\mathfrak{a}, \mathfrak{b}$ whose action on E_0 will lead to curves $E_A = \mathfrak{a} * E_0$ and $E_B = \mathfrak{b} * E_0$. These curves have also a $\mathfrak{O}$-orientation which implies the existence of ℓ^n-isogenies $\varphi_A : F_0 \to E_A$ and $\varphi_B : F_0 \to E_B$ as in Proposition 2.2. Alice public key will be E_A together with some torsion points (which will allow Bob to compute $\mathfrak{b} \star E_A$).

Proposition 5.5. *When $\mathfrak{O}_0$ is a quadratic order of class number 1 and $\mathfrak{O} = \mathbb{Z} + \ell^n \mathfrak{O}_0$, then if there exists a PPT algorithm that can break Problem 5.1, there is a PPT algorithm that can recover the keys of the OSIDH protocol presented in [12].*

Proof. From the definition of the group action of $\mathrm{Cl}(\mathfrak{O})$ on the curves having an $\mathfrak{O}$-orientation (see [12]), finding a smooth ideal $\mathfrak{c}$ such that $E_A = \mathfrak{c} * E_0$ is enough to recover the secret key.

Note that we do not have equivalence in Proposition 5.5 because the OSIDH public keys include more information than just curves. This will be the same for SIDH and Proposition 5.7.

For SIDH, we write[15] F_0 for the common starting curve. In SIDH, recovering the secret key from the public key is equivalent to the computational supersingular isogeny problem (CSSI), see [23] that we state in Problem 5.6.

Problem 5.6. Let ℓ_A be a small prime number and $A = \ell_A^{e_A}$ for some exponent e_A. Let $\varphi_A : F_0 \to E_A$ be an isogeny whose kernel is $\langle [m_A]P_A + [n_A]Q_A \rangle$, where m_A and n_A are chosen at random from $\mathbb{Z}/A\mathbb{Z}$ (where at least one is in $\mathbb{Z}/A\mathbb{Z}^\times$. Given E_A and the values $\varphi_A(P_B), \varphi_A(Q_B)$ for P, B, Q_B a basis of $F_0[B]$ find a generator R_A of $\ker \varphi_A$.

The proposition below requires a bit more work as the link between SIDH and group actions is less obvious.

Proposition 5.7. *Assume that F_0 admits an $\mathfrak{O}_0$-orientation with $\mathfrak{O}_0$ a maximal quadratic order of class number 1. If there exists a PPT algorithm solving Problem 5.1 for $\mathfrak{O} = \mathbb{Z} + A'\mathfrak{O}_0$ where A' divides A, then there exists a PPT algorithm that breaks the CSSI problem with overwhelming probability.*

Proof. First, note that A is chosen so that the kernel points of A-isogenies have a polynomial-size representation. Then, since A is also smooth, the discrete logarithms can be solved in polynomial time in the A-torsion and isogenies of degree A can be computed in polynomial time.

For the rest of this proof, let us write α the endomorphism of F_0 such that $\mathbb{Z}[\alpha]$ realizes the embedding of $\mathfrak{O}_0$ inside $\mathrm{End}(F_0)$.

[15] Once again, we highlight that these notations are unusual and were chosen to emphasize the link with Problem 5.1.

If the curve E_A is A-isogenous to F_0, then E_A admits an embedding of $\mathbb{Z} + A\mathfrak{O}_0$. This embedding is not necessarily primitive but we know there exists A' dividing A such that $\mathfrak{O} = \mathbb{Z}+A'\mathfrak{O}_0$ admits a primitive embedding in $\mathrm{End}(E_A)$ (see Proposition 2.2). Conversely, since the class number of $\mathfrak{O}_0$ is 1, then any $\mathbb{Z} + A'\mathfrak{O}_0$-orientation on E_A implies the existence of an A'-isogeny between E_A and F_0. Let us write $\varphi_{A'} : F_0 \to E_A$ this isogeny of degree A'. Then φ_A, the secret isogeny in Problem 5.6 is the composition of φ_A with an endomorphism θ_A of $\mathfrak{O}_0$ of degree A/A'. Since A/A' is a power of ℓ_A, there are two possibilities for θ_A. Thus, the difficulty lies in recovering $\varphi_{A'}$.

We can generate a curve E_0 in $\mathcal{E}_{\mathbb{Z}+A'\mathfrak{O}_0}$ by generating $\varphi_0 : F_0 \to E_0$ a descending isogeny of degree A'. Any ideal $\mathfrak{a}$ such that $E_A = \mathfrak{a} * E_0$ can be interpreted as an isogeny $\varphi_\mathfrak{a} : E_0 \to E_A$ of degree $n(\mathfrak{a})$. The proof is concluded by the fact that $\ker \hat{\varphi}_{A'} = \varphi_\mathfrak{a}(\ker \hat{\varphi}_0)$, which we prove below. Once $\ker \hat{\varphi}_{A'}$ has been computed, is easy to recover $\ker \varphi_{A'} = \hat{\varphi}_{A'}(E_A[A'])$ and find a solution to the CSSI as we explained above.

To prove $\ker \hat{\varphi}_{A'} = \varphi_\mathfrak{a}(\ker \hat{\varphi}_0)$, we need to understand how the fact that $\mathfrak{a}$ is an $\mathfrak{O}$-ideal translates on the action of $\varphi_\mathfrak{a}$ on $\hat{\varphi}_0$. As explained in Proposition 2.2 and the following paragraph, the embedding of $\mathfrak{O}$ in E_0 (resp. E_A) is obtained as $\mathbb{Z}[\varphi_0 \circ \alpha \circ \hat{\varphi}_0] = \mathbb{Z}[\theta_0]$ (resp. $\mathbb{Z}[\varphi_{A'} \circ \alpha \circ \hat{\varphi}_{A'}] = \mathbb{Z}[\theta_{A'}]$). By definition of $\mathfrak{a}$ being an $\mathfrak{O}$-ideal, we have that $\varphi_\mathfrak{a}(\ker \theta_0) = \ker \theta_A$. Thus, we need to prove that $\ker \theta_0 \cap E_0[A'] = \ker \hat{\varphi}_0$ and $\ker \theta_{A'} \cap E_A[A'] = \ker \hat{\varphi}_A$ (note that this property is exactly what underlies the inversion mechanism in Section 3.3). We will do it for θ_0, the property for $\theta_{A'}$ holds for the exact same reasons. It is clear from the definition of $\theta_0 = \varphi_0 \circ \alpha \circ \hat{\varphi}_0$ that we have $\ker \hat{\varphi}_0 \subset \ker \theta_0$. Let us take $P \in E_A[A'] \setminus \ker \hat{\varphi}_0$, then $Q = \hat{\varphi}_0(P) \in \ker \varphi_0 \setminus \langle 0 \rangle$. If we assume that $P \in \ker \theta_0$, it implies that $\alpha(Q) \in \ker \varphi_0$. Since $\ker \varphi_0$ is cyclic, we have that $\alpha(Q) = \lambda Q$ for some $\lambda \in \mathbb{Z}$. This contradicts the fact that φ_0 is descending. Indeed, if we write φ_Q, the isogeny of kernel generated by Q, we have $\varphi_0 = \psi_0 \circ \varphi_Q$ for some isogeny φ_Q and the condition $\alpha(Q) = \lambda Q$ implies that φ_Q is not descending and so φ_0 would not be descending, which is a contradiction. Thus, we have proven that $\ker \theta_0 \cap E_0[A'] = \ker \hat{\varphi}_0$ and this concludes the proof asexplained above.

We refer to Section 3 for the full details and notations about Séta. We write $\mathfrak{O} \cong \mathbb{Z}[\sqrt{(N^2e - d^2)/D^2}] \cong \mathbb{Z}[\theta]$ and assume that $e, d, \mathfrak{O}$ are public. This assumption is plausible as the procedure described in Algorithm 2 is essentially deterministic.

Proposition 5.8. *If there exists a PPT algorithm solving Problem 5.1 for $\mathfrak{O}$, then there exists a PPT algorithm that takes a Séta public key E_s and recovers a trapdoor T such that E_{j_T}, T is a (D, N)-trapdoor curve.*

Proof. Let E_{j_T} be a Séta public key. By applying Algorithm 3 in $\mathfrak{O}$ and adding the integers e, d a (D, N)-trapdoor curve E_0, T_0 can be found in polynomial time with $E_0 \in \mathcal{E}_\mathfrak{O}$. Thus, we can apply the PPT solver for Problem 5.1 on E_0 and E_{j_T} to compute an isogeny $\varphi_\mathfrak{a} : E_0 \to E_{j_T}$ corresponding to a $\mathfrak{O}$ ideal $\mathfrak{a}$. If we write $\theta_0 \in \mathrm{End}(E_0)$ and $\theta \in \mathrm{End}(E_{j_T})$ the endomorphisms such that $\mathfrak{O} \cong \mathbb{Z}[\theta_0] \cong \mathbb{Z}[\theta]$. Then, by definition of $\mathfrak{O}$-ideals, we have that $\theta \circ \varphi_\mathfrak{a} = \varphi_\mathfrak{a} \circ$. So if $T_0 =$

$e, d, P_0, Q_0, \theta_0(P_0), \theta_0(Q_0)$, then $T = e, d\varphi_{\mathfrak{a}}(P_0), \varphi_{\mathfrak{a}}(Q_0), \varphi_{\mathfrak{a}}(\theta_0(P_0)), \varphi_{\mathfrak{a}}(\theta_0(Q_0))$ is such that E_{j_T}, T is a (D, N)-trapdoor curve.

We finish this section by proving that some instances of Problem 5.1 are related to the more generic isogeny problem of finding a smooth isogeny between any two supersingular curves (Problem 5.9 below). For that it suffices to show that there exists some quadratic order that is embedded inside the endomorphism ring of any supersingular curve.

Problem 5.9. Let $p > 3$, be a prime number. Given E_1,E_2 two distinct supersingular curves over $\mathbb{F}_{p^2}$. Find $\varphi : E_1 \to E_2$, an isogeny of powersmooth degree.

Proposition 5.10. *There is an absolute constant $c > 0$ such that the following holds. Let $\mathfrak{O}$ be a quadratic order of conductor ℓ^e inside $\mathfrak{O}_0$ a maximal quadratic order, such that ℓ is inert in $\mathfrak{O}_0$, and $e \geq c \log_\ell(p)$. If there exists a PPT algorithm that can break Problem 5.1, then there is a PPT algorithm that breaks Problem 5.9.*

Proof. From the fact that the ℓ-isogeny graph is Ramanujan, and the rapid mixing of non-backtracking random walks in expander graphs [1], we deduce that for $e = \Omega(\log_\ell(p))$, there exists a non-backtracking path of degree ℓ^e between any two supersingular curves in the graph.

In particular, if E_0 is any $\mathfrak{O}_0$-orientable curve, there exists a cyclic isogeny of degree ℓ^e from E_0 to any other E, and since ℓ is inert in $\mathfrak{O}_0$, this isogeny must be a sequence of descending isogenies. This implies that any E is $\mathfrak{O}$-orientable. Thus, if we write E_1 and E_2, the two curves in the generic isogeny problem, then we can construct a middle curve E_0 with an explicit embedding of $\mathfrak{O}$, then use the PPT algorithm to find paths between E_0, E_1 and E_0, E_2, and finally concatenate the two paths to obtain a path between E_1 and E_2 of powersmooth degree.

5.3 Analysis of the uber isogeny assumption

In this section we investigate the complexity of solving Problem 5.1. We are going to see that there are various special cases leading to various complexities.

We start by giving a generic estimate which can be seen as the worst case complexity.

A first upper bound: exhaustive search The simplest method to solve Problem 5.1 is to apply an exhaustive search, for instance by selecting a set of small primes ℓ_i all split in $\mathfrak{O}$ and trying all combinations of $\prod \mathfrak{l}_i^{e_i} \star E_0$ until one is isomorphic to E_s, where each $\mathfrak{l}_i$ is a prime ideal above ℓ_i. The expected running time of this algorithm is in $O(\#\mathcal{E}_{\mathfrak{O}})$. The best generic bound on the size of this set is given in Proposition 2.1.

The classical estimate $h(\mathfrak{O}) = \Theta(\sqrt{\Delta})$ gives a first upper-bound on the complexity to solve Problem 5.1. In particular, it shows that solving Problem 5.1 is

easy when the discriminant Δ is small. However, when Δ grows, it is harder to estimate how this bound reflects on the actual complexity of the problem.

There are some special cases for which we can be a bit more precise than Proposition 2.1. For instance, when the discriminant are short, the following Theorem from Kaneko [25] can be applied to derive a precise statement.

Theorem 5.11. *Take two distinct quadratic orders $\mathfrak{O}_1, \mathfrak{O}_2$ of discriminants Δ_1, Δ_2 embedded optimally in the same maximal order inside the quaternion algebra ramified exactly at p and ∞. If we have $\mathbb{Q}(\sqrt{\Delta_1}) \cong \mathbb{Q}(\sqrt{\Delta_2})$, then $\Delta_1\Delta_2 \geq p^2$.*

Applying Theorem 5.11 to the discriminants $\Delta \leq p$, we see that there cannot be two distinct embeddings of $\mathfrak{O}$ inside the same maximal order, thus proving that $\#\mathcal{E}_{\mathfrak{O}} = h(\mathfrak{O})$. Thus, in that case, we know that the exhaustive search method described above has asymptotic complexity $\Theta(\sqrt{\Delta})$.

Another example is given in the proof of Proposition 5.10, where we saw that there are some values of Δ for which we know that $\mathcal{E}_{\mathfrak{O}}$ is exactly the set of supersingular curves. More generally, the link between the conductor of $\mathfrak{O}$ and isogenies (Proposition 2.2) allows us to obtain some better estimates on the size of $\mathcal{E}_{\mathfrak{O}}$ by using the expander properties of isogeny graphs.

The case of CSIDH (Proposition 5.4) has received a lot of attention from the community ([6, 9, 11, 32] since it was the first scheme that naturally fits into this framework. In fact, there are improvements over the exhaustive search strategy in both the classical and quantum settings. The main ingredient behind these speed-ups is the ability for anyone to obtain a concrete embedding (through the Frobenius morphism) of $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$ inside $\mathrm{End}(E)$ for any $E \in \mathcal{E}_{\mathfrak{O}}$. In particular, computing $\mathfrak{a} \star E$ becomes easy for any $E \in \mathcal{E}_{\mathfrak{O}}$ when $\mathfrak{a}$ has smooth norm. In the classical setting, this implies a quadratic speed-up over the generic exhaustive search by using a meet-in-the-middle technique (see [9]). In the quantum setting, the speed-up is even more radical, as it creates a malleability oracle (see [28]) that reduces CSIDH's security to an instance of the hidden shift problem which can be solved in quantum sub-exponential time as described in [6, 32] for instance.

Note that neither of these attacks can be used in the generic case as it seems hard to obtain this malleability oracle for other group actions. For instance, in OSIDH [12] the public keys are made of a curve E and some torsion points to make possible the computation of $\mathfrak{a} \star E$ for some secret ideal $\mathfrak{a}$. These additional torsion points are not needed in CSIDH because they can be easily computed.

Smooth conductor inside a maximal quadratic order A better algorithm also exists when the conductor f of $\mathfrak{O}$ is smooth. By Proposition 2.2, there exists an isogeny of degree f between any curve $E \in \mathcal{E}_{\mathfrak{O}}$ and any curve in $\mathcal{E}_{\mathfrak{O}_0}$, where $\mathfrak{O}_0$ is the quadratic maximal order containing $\mathfrak{O}$. Let E_0, E_s given by in an instance of Problem 5.1, and let us write $\varphi_0 : F_0 \to E_0$ and $\varphi_s : F_s \to E_s$ the two isogenies of degree f.

The alternative resolution method enumerates through all possible $F_s = \mathfrak{a}_0 \star F_0$ in $\mathcal{E}_{\mathfrak{O}_0}$ then tries to find φ_s of degree f. Since f is smooth, we can apply a meet-in-the-middle technique to reduce this part to $O(\sqrt{f})$. Once $\varphi_s : F_s \to E_s$ and a $\mathfrak{O}_0$-ideal $\mathfrak{a}_0$ such that $F_s = \mathfrak{a}_0 \star F_0$ has been found, we can compute a $\mathfrak{O}$-ideal such that $E_s = \mathfrak{a} \star E_0$ as described in [12, Section 5.1].

If we write $\Delta = f^2\Delta_0$ where Δ_0 is the fundamental discriminant of $\mathfrak{O}_0$. The complexity of this algorithm is $\Theta(\sqrt{f}\sqrt{\Delta_0})$ which is better than $\Theta(\sqrt{\Delta}) = \Theta(f\sqrt{\Delta_0})$.

Other cases When we are not in one of the above cases, there is no known improvement over the exhaustive search (classically or quantumly). Thus, the presumed security entirely relies on the size of $\mathcal{E}_{\mathfrak{O}}$. In that regard, the cases where the conductor of $\mathfrak{O}$ is big might give more confidence in the difficulty of Problem 5.1 as the size of $\mathcal{E}_{\mathfrak{O}}$ is tied to the number of isogenies of a given degree between distinct pair of curves. In comparison, the distribution of embeddings of a maximal quadratic order of big discriminant (i.e above the bound in Theorem 5.11) have been less studied. As of yet, there are no reason to believe that there exists such quadratic orders that would be embedded in only a small portion of all the supersingular curves but not enough work has been done on the question to reach a definitive conclusion.

6 Implementation

We implemented the version of Séta where the starting curve (E_{j_T}, T) is a (D, N)-trapdoor curve, i.e, the secret key does not contain a random walk, as described in Section 4.2. Our implementation is written in pure C, reusing large parts of the codebase of SQISign[16]; in particular we depend on GMP 6.2.1 for integer arithmetic, Pari 2.13 for quaternion arithmetic [40], and we adapt the so called `velusqrt` code for isogeny evaluation [4][17]. Our code is avaible at `https://github.com/seta-isogeny-encryption/seta`.

6.1 Main building blocks

Key generation consists of two parts. Finding a suitable θ in its quaternion form and then finding a supersingular elliptic curve whose endomorphism ring contains θ. The difficult part of this procedure in practice is a subroutine for finding a supersingular elliptic curve whose endomorphism ring is isomorphic to a particular maximal order $\mathcal{O}$. For this step we reused a substantial amount of the code used for SQISign [16].

Encryption consists in the evaluation of an isogeny of degree D at points of order N. In order to make this efficient we choose parameters where D has small prime factors and both D and N divide $p^2 - 1$ to avoid using extension fields.

[16] `https://github.com/SQISign/sqisign`
[17] `https://velusqrt.isogeny.org/software.html`

Decryption also uses evaluations of isogenies, but here isogenies of degree N are evaluated. Furthermore, decryption requires some linear algebra modulo D (when computing the intersection $\ker(\tau - [d]) \cap E_m[D]$) and modulo N (when computing the isogenies ψ and ψ'). In these steps one uses subroutines for solving discrete logarithms but due to N and D being smooth, this step is negligible compared to other computations.

6.2 Prime search

To efficiently implement Séta, it is necessary to select a prime satisfying the many constraints mentioned in Section 4.3. To maximise efficiency of encryption and decryption, while maintaining reasonably efficient key generation, we opted to search for a prime satisfying the following constraints: (1) $p^2 - 1 = DN$, with both D and N smooth; (2) $D \approx 2^{2\lambda}$ and $N \approx 2^{4\lambda}$; and (3) D has as few prime factors as possible.

There are currently three known techniques to search for primes such that $p^2 - 1$ is smooth, all discussed in [14]. Of these, the most apt to satisfy the constraint that D has few prime factors was introduced by Costello in [13]: fix an exponent $n > 1$, and sieve the space of integers $p = 2x^n - 1$ until one is found such that both $p + 1 = 2x^n$ and $p - 1 = 2(x^n - 1)$ are smooth.

Thanks to this technique, D can be taken as a factor of $p + 1$, and has thus much fewer prime factors than a generic smooth prime of the same size. The drawback of the technique is that, as n increases, the search space decreases, to the point where no smooth integers may be found.

Concretely, for $\lambda = 128$, we fixed $n = 12$ and we sieved within the space $2^{32} < x < 2^{33}$, i.e., $2^{385} < p < 2^{397}$. This yielded four primes with largest factor bounded by 2^{25}, and three with bound 2^{26}, corresponding to $x = 4679747572$, 4845958752, 4966654633, 5114946480, 6334792777, 8176556533, 8426067021. Unfortunately, the search space was fully explored, meaning that no better primes exist for $n = 12$.

The relatively large smoothness bounds negatively affect performance of all algorithms in Séta. Unfortunately, it appears to be difficult to find better primes given current knowledge. Even dropping the constraint on the number of prime factors of D, the best algorithms known today can hardly beat a 2^{20} smoothness bound for a prime of 384 bits [14, Table 3].

6.3 Experimental results

We ran experiments on a 4.00GHz Quad-Core Intel Core i7, using a single core. We used the prime $p = 2 \cdot 8426067021^{12} - 1$, and the smooth factors

$$
\begin{aligned}
D &= 43^{12} \cdot 84719^{11}, \\
N &= 3^{21} \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 73 \cdot 257^{12} \cdot 313 \cdot 1009 \cdot 2857 \cdot 3733 \cdot 5519 \cdot 6961 \\
&\quad \cdot 53113 \cdot 499957 \cdot 763369 \cdot 2101657 \cdot 2616791 \cdot 7045009 \cdot 11959093 \\
&\quad \cdot 17499277 \cdot 20157451 \cdot 33475999 \cdot 39617833 \cdot 45932333.
\end{aligned}
$$

The key generation was ran only once, and took 10.43 hours. The encryption procedure took 4.63 seconds, and the decryption took 10.66 minutes, averaged over six runs. The decryption time is almost entirely devoted to the evaluation of isogenies of degrees the largest factors of N.

7 Further work and conclusion

The efficiency of the scheme essentially depends on the prime factorization of D. We have managed to keep all computations within $\mathbb{F}_{p^2}$ but D still has large prime factors. In principle, one can construct trapdoor curves whenever $N > D^2$ so in particular when ND divides $p - 1$ and $N = 2^k, D = 3^l$. The bottleneck here is the generation of the trapdoor curve which is rather inefficient, despite its polynomial complexity. Note that generating the curve does not affect the speed of encryption and decryption, it only affects the speed of key generation. Thus if one devised a more efficient version of the KLPT algorithm which speeds up the maximal order to elliptic curve mapping algorithm, then one could derive a much more efficient scheme. We estimate that in the best case, one could get a scheme which is only 5 times slower than SIDH. Another interesting research direction is whether one could build upon our Séta scheme and derive more advanced primitives. The framework of Séta has certain advantages in this context when compared to SIDH. First, Séta is based on a trapdoor one-way function which could be useful in building signature schemes. Second, SIDH-based constructions are more likely to need a trusted setup to avoid backdoor curve attacks such as the one described in [3, Section 6]. Finally, public key validation is easy in the context of Séta which could be used to build non-interactive key exchange or counteract fault attacks.

This work presents the OW-CPA PKE scheme Séta, built upon a generalized version of the isogeny-based CGL hash function family. To do so, we made use of a "torsion-point attack" against SIDH-like schemes [33] and transformed this into a decryption mechanism which recovers a message encrypted as a secret isogeny between a trapdoor starting curve and a final ciphertext curve. An IND-CCA variant is constructed using the post-quantum OAEP transform and both security properties are proven to reduce to the TCSSI problem, derived from the CSSI problem introduced in [23]. We then discussed the key generation in terms of computing trapdoor information, the corresponding curve generation, and of the constraints that this does or does not place on the base prime of the scheme; we also proposed an alternative method for these computations. Of independent interest, we formalized the "uber isogeny asumption" and discussed its relation with existing isogeny-based schemes, such as CSIDH, OSDIH and SIDH, before analyzing its complexity. Finally, we presented implementation results for both the search of a well-suited base prime and for key-generation, encryption and decryption experiments.

Acknowledgments. We would like to thank the anonymous reviewers for their remarks and suggestions. Péter Kutas and Christophe Petit's work was supported by EPSRC grant EP/S01361X/1. Péter Kutas was also supported by

the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary. Cyprien Delpech de Saint Guilhem's work was supported by by ERC Advanced Grant ERC-2015-AdG-IMPaCT, by DARPA under contract No. HR001120C0085, and by CyberSecurity Research Flanders with reference number VR20192203.

Bibliography

[1] Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 9(04):585–603, 2007.

[2] Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Aaron Hutchinson, Amir Jalali, David Jao, Koray Karabina, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular isogeny key encapsulation, 2020.

[3] Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Antonio Sanso. Cryptanalysis of an oblivious PRF from supersingular isogenies. Cryptology ePrint Archive, Report 2021/706, 2021.

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

[5] Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *International Conference on Cryptology in India*, pages 428–442. Springer, 2014.

[6] Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In *Advances in Cryptology - EUROCRYPT 2020*, pages 493–522, 2020.

[7] Xavier Boyen. The uber-assumption family. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, volume 5209 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2008.

[8] Reinier Bröker. Constructing supersingular elliptic curves. *Journal of Combinatorics and Number Theory*, 1(3), 2009.

[9] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology - ASIACRYPT 2018*, pages 395–427, 2018.

[10] Denis X Charles, Kristin E Lauter, and Eyal Z Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009.

[11] Jorge Chávez-Saab, Jesús-Javier Chi-Domınguez, Samuel Jaques, and Francisco Rodrıguez-Henrıquez. The SQALE of CSIDH: square-root vélu quantum-resistant isogeny action with low exponents. Technical report,

Cryptology ePrint Archive, Report 2020/1520, 2020. https://eprint. iacr. org . . . , 2020.
[12] Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, 14(1):414–437, 2020.
[13] Craig Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. Technical report, Cryptology ePrint Archive, Report 2019/1145, 2019. https://eprint.iacr.org/2019/1145, 2019.
[14] Craig Costello, Michael Meyer, and Michael Naehrig. Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem. Cryptology ePrint Archive, Report 2020/1283, 2020.
[15] Jean-Marc Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006:291, 1999.
[16] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 64–93. Springer, 2020.
[17] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Improved torsion point attacks on SIDH variants. *arXiv e-prints*, page arXiv:2005.14681, May 2020.
[18] Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 329–368, Cham, 2018. Springer International Publishing.
[19] Tako Boris Fouotsa and Christophe Petit. InSIDH: a Simplification of SiGamal. Cryptology ePrint Archive, Report 2021/218, 2021. `https://eprint.iacr.org/2021/218`.
[20] 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, 2016.
[21] Steven D Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. *Journal of Cryptology*, 33(1):130–175, 2020.
[22] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. *IACR Cryptology ePrint Archive*, 2017:604, 2017.
[23] 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.
[24] David Jao, Stephen D Miller, and Ramarathnam Venkatesan. Expander graphs based on grh with an application to elliptic curve cryptography. *Journal of Number Theory*, 129(6):1491–1504, 2009.
[25] Masanobu Kaneko. Supersingular j-invariants as singular moduli mod p. *Osaka Journal of Mathematics*, 26(4):849–855, 1989.
[26] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.

[27] Péter Kutas, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Weak instances of SIDH variants under improved torsion-point attacks. *IACR Cryptology ePrint Archive*, 2020:633, 2020.

[28] Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Charlotte Weitkämper. One-way functions and malleability oracles: Hidden shift attacks on isogeny-based protocols. *IACR Cryptology ePrint Archive*, 2021:282, 2021.

[29] Jonathan Love and Dan Boneh. Supersingular curves with small noninteger endomorphisms. *Open Book Series*, 4(1):7–22, 2020.

[30] Chloe Martindale and Lorenz Panny. How to not break SIDH. Cryptology ePrint Archive, Report 2019/558, 2019. `https://eprint.iacr.org/2019/558`.

[31] Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. Sigamal: A supersingular isogeny-based pke and its application to a prf. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 551–580. Springer, 2020.

[32] Chris Peikert. He gives C-sieves on the CSIDH. In *Advances in Cryptology - EUROCRYPT 2020*, pages 463–492, 2020.

[33] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *Advances in Cryptology - ASIACRYPT 2017*, pages 330–353, 2017.

[34] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006:145, 2006.

[35] Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151 of *Graduate Texts in Mathematics*. Springer-Verlag, New-York, 1994.

[36] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.

[37] Denis Simon. Quadratic equations in dimensions 4, 5 and more. *Preprint*, 2005.

[38] 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.

[39] 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.

[40] The PARI Group, Université de Bordeaux. *PARI/GP version `2.12.0`*, 2021. available from `http://pari.math.u-bordeaux.fr/`.

[41] Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris, Séries A*, 273:305–347, 1971.

[42] John Voight. Identifying the matrix ring: algorithms for quaternion algebras and quadratic forms. In *Quadratic and higher degree forms*, pages 255–298. Springer, 2013.

A Supersingular elliptic curves

We recall definitions and results related to 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 at infinity of E_1 to the point of infinity at 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 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, together with the zero map, 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. Isogenous curves have isomorphic endomorphism rings thus supersingularity is preserved under an isogeny. 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 [41].

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 $\mathcal{J}_p$.

Twists of elliptic curves. As presented in [**?** , Section 2.4], two curves $E_1 \sim (A_1, B_1)$ and $E_2 \sim (A_2, B_2)$ are isomorphic over $\overline{\mathbb{F}_{p^2}}$ if and only if there is some $u \in \overline{\mathbb{F}_{p^2}} \setminus \{0\}$ such that $A_1 = u^4 A_2$ and $B_1 = u^6 B_2$. It happens that two curves defined over $\mathbb{F}_{p^2}$ are isomorphic over $\overline{\mathbb{F}_{p^2}}$ but not over $\mathbb{F}_{p^2}$; such curves are *twists* of one another.n For $p \neq 2, 3$, a quadratic twist of $E \sim (A, B)$ is any curve of the form $E^t \sim (t^2 A, t^3 B)$ for $t \in \mathbb{F}_{p^2} \setminus \mathbb{F}_{p^2}^2$ (i.e. t is not a square in $\mathbb{F}_{p^2}$). Curves with j-invariant equal to 0 or 1728 are treated separately and we refer to [**?** , Section 2.4].

Canonical curves. We take the same approach as [21, Appendix A] to fix a canonical choice of curve for each j-invariant. 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 $\Phi_\ell(x, y) \in \mathbb{Z}[x, y]$ [35]. 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.

B Post-quantum OAEP transformation

We present here the post-quantum OAEP generic transformation we used in Section 3.5.

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 $r \xleftarrow{\$} \{0,1\}^{k_0}$ and set

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

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 $\perp$.