POKÉ: A Compact and Efficient PKE from Higher-Dimensional Isogenies

Andrea Basso[1,2(✉)] and Luciano Maino[1]

[1] IBM Research Europe, Zürich, Switzerland
andrea.basso@ibm.com, luciano.maino@bristol.ac.uk
[2] University of Bristol, Bristol, UK

Abstract. We introduce a new PKE protocol, POKÉ, based on isogenies of unknown degree. The protocol relies on two new techniques: the first constructs an SIDH square while also working with higher-dimensional representations, whereas the second allows us to obtain a shared secret even when all curves in the commutative diagram are known.

The resulting protocol is compact and extremely efficient. We provide a proof-of-concept implementation in SageMath of POKÉ that shows encryption and decryption taking about a hundred milliseconds at security level I: POKÉ is thus the most efficient encryption protocol from isogenies, and it outperforms existing protocols by more than an order of magnitude.

1 Introduction

Since its inception nearly two decades ago, isogeny-based cryptography has focused on isogenies between elliptic curves. Although some protocols based on isogenies between abelian varieties (a generalization of elliptic curves to higher dimensions) had been proposed (see, for instance, [25,33,34]), these constructions were often less studied and less efficient than their one-dimensional counterparts. This trend drastically changed in 2022: SIDH [20,30], possibly the most efficient and well-known isogeny-based PKE, was broken in a series of three breakthrough papers by Castryck and Decru [7], Maino, Martindale, Panny, Pope and Wesolowski [35], and Robert [45].

These attacks, which rely on computing isogenies between abelian varieties, led to a complete key-recovery attack on SIDH. But, more fundamentally, they drastically changed the landscape in isogeny-based cryptography: in particular, the attacks showed that, given two curves, the degree of the connecting isogeny, and its action on a sufficiently large torsion basis, it is always possible to recover the connecting isogeny. If its kernel generators are accessible, recovering the isogeny means obtaining one kernel generator; this is not possible if the degree is not smooth, but the tools that power the SIDH attacks still enable evaluating the connecting isogeny at any point. Since *knowing* an isogeny really means being able to evaluate it, the techniques used to attack SIDH showed that two sets of curves and torsion points (together with the degree) can provide an alternative

S. Fehr and P.-A. Fouque (Eds.): EUROCRYPT 2025, LNCS 15602, pp. 94–123, 2025.
https://doi.org/10.1007/978-3-031-91124-8_4

representation of an isogeny. This is the first representation that can efficiently describe isogenies of non-smooth degree between any two curves.

These new techniques to represent isogenies, especially those of non-smooth degree, as well as the development of better algorithms to efficiently work with such representations [12,17,42,50] led to a renewed interest in developing cryptographic protocols based on isogenies between abelian surfaces. Existing protocols in dimension one, such as SQIsign [21] and SCALLOP [19], obtained considerable improvements by relying on higher-dimensional representations [2,11,16,24,41]. New constructions, which could only be built from higher-dimensional representations, were also proposed soon after, such as FESTA [4] and its improved version QFESTA [40].

At the same time, new protocols that rely on an SIDH-like structure (i.e. a commutative diagram of isogenies of coprime degree) but are secure were also developed: initially, Fouotsa, Moriya, and Petit [26] proposed countermeasures against the SIDH attacks that relied on masked degrees and masked torsion points. They obtained MD-SIDH and M-SIDH, which are larger and slower than SIDH, but avoid the existing attacks. More recently, Basso and Fouotsa developed binSIDH and terSIDH [3], which also avoid the attacks on SIDH while being significantly more efficient than M-SIDH and MD-SIDH. The authors also suggested a mixed approach, where one party computes binSIDH or terSIDH isogenies, while the other relies on SIDH-like isogenies.

However, the two approaches, developing SIDH-like protocols in dimension one on one side, and using higher-dimensional isogenies on the other, have not mixed. Although the higher-dimensional protocols mentioned above still compute one-dimensional isogenies, their constructions are fundamentally different from those of SIDH-like protocols. To date, hardly any protocol that builds a commutative diagram à la SIDH works with higher-dimensional isogenies. A possible reason behind this is that SIDH-like protocols rely on isogenies with a kernel generator defined over $\mathbb{F}_{p^2}$: such isogenies can be represented with a single curve (its domain, together with a kernel generator); higher-dimensional isogenies, on the other hand, often represent isogenies whose kernel has generators defined over large extension fields of $\mathbb{F}_{p^2}$: these isogenies require a two-curve representation, where both the domain and the codomain are provided. The only partial exception is IS-CUBE [36], which relies on long isogenies (much longer than the order of the torsion images revealed) to build an SIDH diagram while also working with a two-dimensional representation: while the general framework is similar to this work, the resulting protocol has large ciphertext and particularly inefficient decryption. In independent and concurrent work, the author of IS-CUBE also proposed a new and more efficient protocol, LIT-SiGamal [37], that shares some similarities with this work.

Contributions. In this work, we introduce a new PKE, POKÉ, that combines the two approaches: it constructs a commutative diagram where two isogenies are represented using higher-dimensional representations. In other words, it allows two parties, one computing isogenies of unknown and non-smooth degree (with kernel generators defined over large extension fields) and the other computing

smooth-degree isogenies (with kernel generators defined over $\mathbb{F}_{p^2}$), to establish a commutative diagram and obtain a shared secret.

Since one isogeny in the exchange needs both its domain and codomain to be recomputed, all four curves in the commutative diagram need to be public: to obtain a shared secret, both parties reveal the (scaled) action of their secret isogeny on a basis and eventually obtain a shared basis on a public curve. Since the two parties end up agreeing on two points, we call the protocol POKÉ[1] for POint-based Key Exchange. We describe our general approach in Sect. 3, while we detail the PKE components in Sect. 4.

The security of the protocol relies on a novel assumption: it is hard to compute an isogeny ϕ from its scaled action $[\alpha]\phi(P), [\alpha]\phi(Q)$ if the isogeny has *unknown degree*. Despite its novelty, the assumption has many connections with more established problems. In Sect. 5, we formalize the security of the protocol, and in Sect. 6 we study the hardness of the new assumption, providing a detailed analysis of its connections and potential attacks.

Lastly, we developed a proof-of-concept implementation in SageMath. We report our experimental results in Sect. 7, which show POKÉ is extremely compact and efficient: it works with a prime of size $\approx 10\lambda/3$ bits. As a consequence, the public keys and ciphertexts are small and compare favourably with most protocols in the literature. Beyond its compactness, POKÉ is also extremely efficient: our unoptimized proof-of-concept implementation in SageMath runs in about a hundred milliseconds for both encryption and decryption (at security level I), suggesting that POKÉ is the most efficient encryption protocol (either a PKE or a key exchange) based on isogenies.

2 Preliminaries

For a thorough treatment of elliptic curves and isogenies, we refer the reader to [47], and for their applications to cryptography, we refer the reader to [18]. We recall some key aspects that are used throughout the paper.

Let $\phi : E_0 \to E_A$ and $\psi : E_0 \to E_B$ be two isogenies with the same domain and coprime degree. We write $\phi_*\psi$ for the pushforward of ψ under ϕ, i.e. the isogeny $\psi' : E_A \to E_{AB}$ such that $\ker \psi' = \phi(\ker \psi)$. In that case, we call ψ and ψ' *parallel* with respect to the isogeny ϕ (which may be omitted, if explicit from the context). Two sets of parallel isogenies, i.e. isogenies $\phi : E_0 \to E_A$, $\psi : E_0 \to E_B$, $\phi' : E_B \to E_{AB}$, and $\psi' : E_B \to E_{AB}$ such that $\phi' = \psi_*\phi$ and $\psi' = \phi_*\psi$, form a *commutative diagram* (of isogenies), which is also known in the literature as an *SIDH square*.

Isogenies can be represented in different ways. We thus recall the definition of an efficient representation of isogenies.

Definition 1 (based on [2, Def. 1]). *Let $\phi : E_0 \to E_1$ be an isogeny defined over a finite field $\mathbb{F}_q$. An* efficient representation *of ϕ is some data $D \in \{0,1\}^*$ of polynomial size in $\log(\deg \phi)$ and $\log(q)$ such that there exist:*

[1] Pronounced ['poʊ.keɪ] (as two syllables), named after the Hawaiian dish (keeping alive the tradition of fishy names in isogeny-based cryptography).

- *an algorithm that, on input D, produces the domain and codomain E_0, E_1;*
- *an algorithm that, on input D, produces the degree* $\deg \phi$*;*
- *an algorithm that, on input D and a point $P \in E_0(\mathbb{F}_{q^k})$, returns $\phi(P)$ in polynomial time in $k \log(q)$ and* $\log(\deg \phi)$*;*

In the literature, several efficient representations are reported. For a complete treatment of efficient isogeny representations, we refer the reader to [46]. In this paper, we focus on two specific types of isogenies.

- SIDH isogenies [30]: they are prime-power degree isogenies (in our case, the degree will always be a power of three) with kernel generators defined over $\mathbb{F}_{p^2}$. Thus, they can be represented with a *kernel representation*, which consists of a curve, its domain, and a single point that generates its kernel.
  To evaluate an SIDH isogeny $\phi : E_0 \to E_1$ on a point P, Vélu's formulas [49] give an efficient way to obtain the codomain E_1 and its evaluation $\phi(P)$.
  If the isogeny degree is ℓ^e, a kernel generator can be expressed as a linear combination of any two linearly independent points of order ℓ^e; thus, computing the pushforward of an SIDH isogeny under another secret isogeny requires revealing the action of the secret isogeny on two linearly independent points of order ℓ^e, possibly both scaled by the same random scalar in $\mathbb{Z}_{\ell^e}^\times$.
- (Q)FESTA isogenies [4,40]: they are isogenies whose degree can be written as $q(2^a - q)$, for some positive value a and any $q < 2^a$; their kernel does not have generators defined over $\mathbb{F}_{p^2}$ but over a large extension field. Hence, we need to use a higher-dimensional representation, which includes their domain and codomain, together with their degree and action on a large torsion basis. Since its degree has the form $q(2^a - q)$, we can use the computationally more efficient representation in dimension two.
  Two-dimensional representations rely on Kani's lemma:

Theorem 2 (Kani's Lemma [31]). *Consider the following commutative diagram of isogenies*

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_1} & E_1 \\ {\scriptstyle \phi_2}\downarrow & & \downarrow{\scriptstyle \phi_2'} \\ E_2 & \xrightarrow[\phi_1']{} & E_{12} \end{array}$$

where $\deg(\phi_i) = \deg(\phi_i') = d_i$ *and* $\gcd(d_1, d_2) = 1$*. Then, the isogeny*

$$\Phi = \begin{pmatrix} \phi_1 & -\hat{\phi}_2' \\ \phi_2 & \hat{\phi}_1' \end{pmatrix} : E_0 \times E_{12} \to E_1 \times E_2$$

is a $(d_1 + d_2)$-isogeny whose kernel is given by

$$\operatorname{Ker} \Phi = \{([-d_1]P, \phi_2' \circ \phi_1(P)) \mid P \in E_0[d_1 + d_2]\}.$$

Thus, a two-dimensional representation of ϕ consists of a description of the kernel of an isogeny Φ, which contains information about ϕ_1 and ϕ_2, which are isogenies of degree q and $2^a - q$ such that $\phi = \phi_1 \circ \phi_2$.

We describe how to evaluate an isogeny ϕ from its two-dimensional representation in Sect. 4.3, while we propose a method to compute its pushforward under a secret isogeny in Sect. 3.

Notation. We write $\mathbb{Z}_n$ to denote the ring $\mathbb{Z}/n\mathbb{Z}$ and $\mathbb{Z}_n^\times$ to denote the group of invertible elements in $\mathbb{Z}_n$; λ denotes the security parameter. The group $\mathrm{SL}_2(\mathbb{Z}_n)$ is the special linear group, the group of determinant-one 2×2 matrices with entries in $\mathbb{Z}_n$. We write $a \leftarrow\!\$\ A$ to denote that a is sampled uniformly at random in the set A.

Throughout the paper, we work with a prime of the form $p = 2^a 3^b 5^c f - 1$, for some choice of $a, b, c, f > 0$, and four curves E_0, E_A, E_B, E_{AB} defined over $\mathbb{F}_{p^2}$. We also use several torsion points: the points P_i, Q_i form a basis of $E_i[2^a]$, the points R_i, S_i form a basis of $E_i[3^b]$, and the points X_i, Y_i form a basis of $E_i[5^c]$. These points are scaled by random values, which are denoted by greek letters and with a subscript expressing the order of the points scaled, e.g. the scalars $\alpha_2, \beta_2, \omega_2$ always scale points of order 2^a, while γ_3 scales points of order 3^b and δ_5 scales points of order 5^c. In some cases, we consider a scaling matrix $\mathbf{D}_5$ that scales points of order 5^c: if $\mathbf{D}_5 = \left[\begin{smallmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{smallmatrix}\right]$, then $\mathbf{D}_5[X_i, Y_i]^T$ represents the points $[d_{11}]X_i + [d_{12}]Y_i$ and $[d_{21}]X_i + [d_{22}]Y_i$.

3 The Building Blocks

In this section, we present the two main building blocks that make up POKÉ: a technique to combine one- and higher-dimensional isogenies to construct a commutative diagram, and a strategy to obtain a shared secret even when all curves in the diagram are publicly known.

3.1 Constructing a Commutative Diagram

A commutative diagram of isogenies, similar to the one at the core of SIDH [30] or CSIDH [8], consists of four isogenies ϕ, ψ, ϕ', ψ' such that ϕ and ψ have coprime degree, and ϕ' and ψ' are the pushforwards of ϕ under ψ and ψ under ϕ, i.e. we have $\ker \phi' = \psi(\ker \phi)$ and $\ker \psi' = \phi(\ker \psi)$. In such a diagram, isogenies *commute*, i.e. $\phi' \circ \psi = \psi' \circ \phi$, which is an important property that can be used to build cryptographic protocols (Fig. 1).

However, isogenies do not generally commute: given the domain and codomain E_0, E_A of ϕ, it is hard to compute ψ and ψ' such that $\psi' = \phi_*(\psi)$. To sidestep the problem, it is thus necessary to reveal additional information about ϕ, while ensuring that ϕ remains secret: in the case of SIDH, this additional information consisted of the degree of ϕ and its images on a large torsion basis; but the SIDH attacks [7,35,45] showed that such information leads to a complete

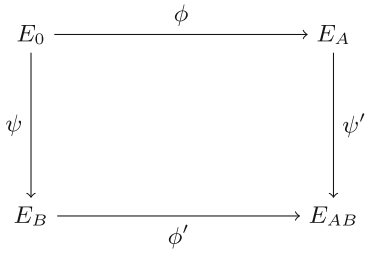

Fig. 1. A commutative diagram of isogenies.

recovery of the isogeny ϕ. In the case of CSIDH, the additional information consists of an orientation, i.e. one specific endomorphism of both E_0 and E_A. While CSIDH is secure, the orientation framework enables a quantum sub-exponential attack [5,10,44] that leads to very large parameters.

We propose a new approach to avoid such issues by relying on isogenies *of unknown degree* and using a higher-dimensional representation. Consider two parties, Alice and Bob, who want to jointly construct a commutative diagram, without revealing their secret isogeny. Let $\phi : E_0 \to E_A$ be an isogeny of unknown degree known to Alice, and consider the basis $R_0, S_0 \in E_0[3^b]$, for some $b > 0$. Then, it is possible for Alice to reveal E_A, together with $R_A, S_A \coloneqq [\gamma_3]\phi(R_0), [\gamma_3]\phi(S_0)$ (where γ_3 is a random scalar[2] in $\mathbb{Z}_{3^b}$), without revealing ϕ since its degree is unknown (we thoroughly analyze this assumption in Sect. 5). At the same time, this information is sufficient to compute parallel isogenies $\psi : E_0 \to E_B$ and $\psi' : E_A \to E_{AB}$: Bob can compute an isogeny ψ with $\ker \psi = \langle R_0 + [r]S_0 \rangle$ and their pushforwards ψ' as the isogeny with kernel $\langle R_A + [r]S_A \rangle$. This is similar to the approach taken in SIDH, but the isogeny ϕ having unknown degree crucially prevents all known isogeny-recovery algorithms.

Analogously to the case of ψ, knowing E_B and E_{AB} (i.e. the codomains of ψ and ψ') is not sufficient for Alice to construct the pushforward $\phi' = \psi_* \phi : E_B \to E_{AB}$. It is thus necessary for Alice to break the efficient representation of ϕ into two parts: one component that can be public and that is enough to be pushed through under some other isogeny, and another component that remains secret. In this case, Bob can then push this representation through ψ and obtain enough information that —together with Alice's secret information— forms an efficient representation of ϕ'. This justifies introducing the following definition.

Definition 3 (Public/private representation). *Let $\phi : E_0 \to E_A$ be an isogeny defined over a finite field $\mathbb{F}_q$. A* public/private efficient representation *of ϕ is some data $D_{\mathsf{pub}} \in \{0,1\}^*, D_{\mathsf{sec}} \in \{0,1\}^*$ of polynomial size in* $\log(\deg \phi)$ *and* $\log(q)$ *such that there exist:*

- *$D_{\mathsf{pub}}, D_{\mathsf{sec}}$ jointly provide an efficient representation, and*

[2] The points $\phi(R_0), \phi(S_0)$ need to be scaled by a random scalar γ_3 to avoid revealing the degree through a simple pairing computation.

– *there exists an algorithm that, on input* D_{pub} *and two isogenies* $\psi : E_0 \to E_B$ *and* $\psi' := \phi_*\psi : E_B \to E_{AB}$*, returns some data* D'_{pub} *such that* $D'_{\mathsf{pub}}, D_{\mathsf{sec}}$ *is an efficient representation of* $\psi_*\phi$ *in polynomial time in* T*, where* T *is the maximum of the costs of evaluating* ψ *and* ψ'*;*

and such that given only D_{pub} *and a point* $P \in E_0(\mathbb{F}_{q^k})$*, it is computationally hard to compute* $\phi(P)$*.*

We show in Sect. 6.1 that a higher-dimensional representation of an isogeny of unknown degree with scaled torsion is indeed a public/private representation. More concretely, let P_0, Q_0 be a basis of $E_0[2^a]$, for some value a. Alice may reveal $P_A, Q_A := [\alpha_2]\phi(P_0), [\beta_2]\phi(Q_0)$, where α_2, β_2 are random scalars in $\mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{2^a}^\times$, (together with E_A, R_A, S_A that she is already revealing for Bob to compute parallel isogenies): Bob can then compute

$$\begin{bmatrix} P_B \\ Q_B \end{bmatrix} := \begin{bmatrix} \omega_2 & 0 \\ 0 & 1/\omega_2 \end{bmatrix} \begin{bmatrix} \psi(P_0) \\ \psi(Q_0) \end{bmatrix}, \qquad \begin{bmatrix} P_{AB} \\ Q_{AB} \end{bmatrix} := \begin{bmatrix} \omega_2 & 0 \\ 0 & 1/\omega_2 \end{bmatrix} \begin{bmatrix} \psi'(P_A) \\ \psi'(Q_A) \end{bmatrix},$$

where $\omega_2 \leftarrow\$ \mathbb{Z}_{2^a}^\times$, and send P_B, Q_B, P_{AB}, Q_{AB} to Alice. Eventually, she can recover ϕ' by scaling the 2^a-points on E_{AB}: the points $[1/\alpha_2]P_{AB}, [1/\beta_2]Q_{AB}$ are the exact images of P_B, Q_B under ϕ', which means Alice can thus obtain an efficient representation of ϕ'.

Hence, both Alice and Bob can compute the pushforward of their secret isogeny under the other's secret isogeny (without revealing their own secret), thus jointly constructing a commutative diagram.

The proposed approach is similar to the one proposed in IS-CUBE [36], but crucially it relies on isogenies of unknown degree rather than isogenies of very large degree. Both approaches avoid the SIDH attacks, but our approach achieves considerably smaller parameters that lead to a much more efficient implementation.

3.2 Obtaining a Shared Secret

The technique presented in the previous section shows how both parties can compute parallel isogenies, and thus jointly construct a commutative diagram, without revealing their secret isogenies. This is the fundamental building block that is necessary to develop encryption protocols. However, most protocols in the literature [3,8,26,30] have used the final curve E_{AB} as a shared secret: this is not possible in our case since we are using two-dimensional representations that include both domain and codomain; indeed, Bob needs to send E_B and E_{AB} to Alice, which makes E_{AB} publicly known. In this section, we propose a method that Alice and Bob can employ to obtain a shared secret, even if all curves are public.

Despite all curves being public, Alice's and Bob's isogenies remain secret. Moreover, they make up a commutative diagram, which means that $\phi' \circ \psi = \psi' \circ \phi$. Since the isogenies (and thus their action) remain secret and commute, we can use this property to obtain a shared secret. Let X_0, Y_0 form a basis of $E_0[5^c]$,

for some exponent c. Alice can reveal the scaled action of ϕ on X_0 and Y_0, i.e. $X_A, Y_A = [\delta_5]\phi(X_0), [\delta_5]\phi(Y_0)$. Similarly, Bob can also reveal the action of ψ on X_0 and Y_0, scaled by a random determinant-one matrix, i.e.

$$\begin{bmatrix} X_B \\ Y_B \end{bmatrix} \coloneqq \mathbf{D}_5 \begin{bmatrix} \psi(X_0) \\ \psi(Y_0) \end{bmatrix},$$

where $\mathbf{D}_5 \leftarrow\!\$\ \mathrm{SL}_2(\mathbb{Z}_{5^c})$. Then, both parties can compute the same points X_{AB}, Y_{AB}, which are defined as

$$\underbrace{\begin{bmatrix} [\delta_5]\phi'(X_B) \\ [\delta_5]\phi'(Y_B) \end{bmatrix}}_{\text{computed by Alice}} =: \begin{bmatrix} X_{AB} \\ Y_{AB} \end{bmatrix} \coloneqq \underbrace{\mathbf{D} \begin{bmatrix} \psi'(X_A) \\ \psi'(Y_A) \end{bmatrix}}_{\text{computed by Bob}},$$

and which can be used as a shared secret.

The proposed approach shares some similarities with SiGamal [39], but it also has significant differences: firstly, our approach does not rely on group actions, which avoids subexponential quantum attacks. Moreover, we rely on two points (rather than one) to obtain smaller parameters. POKÉ also uses the shared secret differently: rather than embedding the message into a scaling factor, the ciphertext includes an extra term that is derived from the message and the shared secret: this leads to a simpler security assumption and a more compact ciphertext.

Remark 4. Note the different scalars employed so far: on the one hand, Alice scales the 2^a-torsion with two independent scalars α_2, β_2, Bob scales the 2^a-torsion with two scalars $\omega_2, 1/\omega_2$; on the other hand, Alice scales the 3^b- and 5^c-order points with the same scalar γ_3 and δ_5 (one for the 3^b-torsion and one for the 5^c-torsion), while Bob scales the 5^c-order points with a full 2×2 matrix of determinant one.

There are multiple reasons for this approach:

- The scalars, viewed as matrices, used by Alice and Bob on the torsion points of the same order need to commute: thus, when Alice uses a diagonal scaling ($\left[\begin{smallmatrix} \alpha_2 & 0 \\ 0 & \beta_2 \end{smallmatrix}\right]$), so does Bob ($\left[\begin{smallmatrix} \omega & 0 \\ 0 & 1/\omega_2 \end{smallmatrix}\right]$); and when she uses a scalar matrix ($\left[\begin{smallmatrix} \delta_5 & 0 \\ 0 & \delta_5 \end{smallmatrix}\right]$), Bob can pick any matrix ($\mathbf{D}_5$).
- Alice uses an isogeny ϕ of degree $q(2^a - q)$, for some only-known-to-Alice q, and thus cannot reveal the action of ϕ on the 2^a-torsion only scaled by a scalar, as shown in the last paragraph of Sect. 6.1.
- At the same time, Alice needs to reveal the action on the 3^b-torsion only up to scalar ($\left[\begin{smallmatrix} \gamma_3 & 0 \\ 0 & \gamma_3 \end{smallmatrix}\right]$), so that Bob can correctly compute pushforwards.
- Bob reveals the action of ψ and ψ' on the 5^c-torsion only up to a full matrix ($\mathbf{D}_5$) because that leads to a tighter reduction (see Sect. 6.2).

4 The POKÉ Protocol

We combine the approaches described in the previous section to obtain a novel PKE, which we call POKÉ. The resulting protocol is one of the most compact post-quantum PKEs and the most efficient isogeny-based PKE to date.

Let p be a prime of the form $p = 2^a 3^b 5^c f - 1$, where f is a small cofactor.[3] Let E_0 be the supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ with j-invariant 1728. Let (P_0, Q_0), (R_0, S_0), (X_0, Y_0) denote bases of $E_0[2^a]$, $E_0[3^b]$, and $E_0[5^c]$, respectively.

4.1 Key Generation

The goal of key generation is to sample a random value $q \in [1, 2^a - 1]$, coprime with $2, 3, 5$, and compute a uniformly random isogeny $\phi : E_0 \to E_A$ of degree $q(2^a - q)$. The secret key consists of a representation of ϕ, while the public key consists of the curve E_A, together with the scaled action of ϕ on the torsion bases on E_0. Namely, the public key consists of:

1. $P_A, Q_A \coloneqq [\alpha_2]\phi(P_0)$, $[\beta_2]\phi(Q_0)$, where $\alpha_2, \beta_2 \leftarrow \mathbb{Z}_{2^a}^\times$;
2. $R_A, S_A \coloneqq [\gamma_3]\phi(R_0)$, $[\gamma_3]\phi(S_0)$, where $\gamma_3 \leftarrow \mathbb{Z}_{3^b}^\times$;
3. $X_A, Y_A \coloneqq [\delta_5]\phi(X_0)$, $[\delta_5]\phi(Y_0)$, where $\delta_5 \leftarrow \mathbb{Z}_{5^c}^\times$.

Note that the 2^a-torsion is scaled diagonally, i.e. with different coefficients for P_0 and Q_0, while the same coefficient is used for both R_0 and S_0 and for X_0 and Y_0. Looking ahead, scaling the 3^b-torsion with the same coefficient is necessary to compute parallel isogenies during encryption, while the diagonal scaling of the 2^a-torsion is needed to guarantee the security of the scheme.

The main computational step during key generation is the sampling of the isogeny $\phi : E_0 \to E_A$ of degree $q(2^a - q)$. We identify two approaches to do so: a rigorous one, which is computationally more intensive but yields uniformly distributed public keys, and a heuristic one, which is more efficient but whose output is only computationally indistinguishable from uniform. The key generation procedure, independent of the choice of isogeny sampling, is represented in Algorithm 1.

Algorithm 1. Key Generation

Input: E_0, $P_0, Q_0 \in E_0[2^a]$, $R_0, S_0 \in E_0[3^b]$, $X_0, Y_0 \in E_0[5^c]$.
Output: A secret/public key pair.

1: Sample a random $q \in [0, 2^a]$ such that $q(2^a - q)$ is coprime with $2, 3$ and 5.
2: Generate a $q(2^a - q)$-isogeny $\phi : E_0 \to E_A$. ▷ Algorithm 2 or Algorithm 3
3: Sample random $\alpha_2, \beta_2 \leftarrow\$ \mathbb{Z}_{2^a}^\times$.
4: Sample a random $\gamma_3 \leftarrow\$ \mathbb{Z}_{3^b}^\times$.
5: Sample a random $\delta_5 \leftarrow\$ \mathbb{Z}_{5^c}^\times$.
6: Compute $P_A, Q_A = [\alpha_2]\phi(P_0), [\beta_2]\phi(Q_0)$.
7: Compute $R_A, S_A = [\gamma_3]\phi(S_0), [\gamma_3]\phi(R_0)$.
8: Compute $X_A, Y_A = [\delta_5]\phi(X_0), [\delta_5]\phi(Y_0)$.
9: **return** $\mathsf{sk} = (q, \alpha_2, \beta_2, \delta_2)$, $\mathsf{pk} = \big(E_A,\ P_A, Q_A,\ R_A, S_A,\ X_A, Y_A\big)$

[3] We remark that, in practice, $p = 3 \pmod 4$ since $a > 2$.

Rigorous Key Generation. The rigorous approach to key generation works with elements of a quaternion algebra due to the Deuring correspondence [23]: similar to what is done in SQIsign2D-West [2], we first sample a random ideal of norm $q(2^a - q)$ and then use an algorithm, based on Clapoti [43], to translate it into an isogeny. This strategy is summarized in Algorithm 2.

More precisely, we start key generation by sampling an integer $q \in [0, 2^a]$ such that $q(2^a - q)$ is coprime with 2, 3, and 5. Then, we generate a random left $\mathcal{O}_0$-ideal I of norm $q(2^a - q)$, where $\mathcal{O}_0$ is the maximal order associated with the endomorphism ring of the starting curve E_0. To do so, we use the RandomFixedNormIdeal algorithm from SQIsign2D-West [2, Alg. 3]: we first find a quaternion μ with norm a multiple of $q(2^a - q)$ and a randomizing quaternion ν of norm coprime with $q(2^a - q)$, and then we produce $I = \mathcal{O}_0\mu\nu + q(2^a - q)\mathcal{O}_0$. Given an ideal I, we can translate it into its corresponding isogeny ϕ with a Clapoti-based approach [2, Alg. 2]:

1. We first sample two ideals I_1, I_2 equivalent to I of norm d_1, d_2, where d_1 and d_2 are pairwise coprime and coprime with $2, 3, 5$. Write ϕ_1, ϕ_2 for the isogenies associated with I_1, I_2.
2. We find integers u, v such that $ud_1 + vd_2 = 2^{a'}3^{b'}5^{c'}$ and $\gcd(ud_1, vd_2) = 1$, for some $a' \leq a$, $b' \leq b$ and $c' \leq c$.
3. Using knowledge of the endomorphism ring of E_0, we compute two isogenies $\phi_u : E_0 \to E_u$ and $\phi_v : E_0 \to E_v$ of degree u and v, respectively. To do so, we use FixedDegreeIsogeny [2, Alg. 7] (based on [40, Alg. 2], and strictly related to our Algorithm 3).
4. Write $P = [2^{a-a'}3^{b-b'}5^{c-c'}](P_0 + R_0 + X_0)$ and $Q = [2^{a-a'}3^{b-b'}5^{c-c'}](Q_0 + S_0 + Y_0)$.
5. We compute the isogeny $\Phi : E_u \times E_v \to E_A \times F$, with kernel
$$\big\langle \big([-d_1]\phi_u(P), \phi_v \circ \widehat{\phi}_2 \circ \phi_1(P)\big), \big([-d_1]\phi_u(Q), \phi_v \circ \widehat{\phi}_2 \circ \phi_1(Q)\big)\big\rangle.$$
6. The isogeny Φ gives us a representation of $\widehat{\phi}_u \circ \phi_1$, from which we can extract ϕ_1 since we know ϕ_u. From ϕ_1, we can obtain the public key E_A as its codomain and evaluate ϕ through [2, Lemma 1].

The isogeny-to-ideal translation in SQIsign2D is highly efficient, but it exploits the fact that the accessible power-of-two torsion is as large as the prime p. Unfortunately, this is not the case in POKÉ: the adapted version of isogeny-to-ideal translation thus requires to compute a chain of $(2, 2)$-isogenies, which is efficient [17], as well as a chain of $(3, 3)$- and $(5, 5)$-isogenies. This comes at a higher computational cost, especially since they are less studied, although very recent work [12,50] shows the feasibility of such computations.

Heuristic Key Generation. The faster key generation is based on an adaptation of the RandIsogImages algorithm, originally proposed in QFESTA [40, Algorithm 2]. The original algorithm generates isogenies starting from E_0 of degree u, as long as the 2^a-torsion is accessible (with $2^a > u$) and $u(2^a - u)$ is

Algorithm 2. Generating a $q(2^a - q)$-isogeny (rigorous) based on [2, Sec. 3.2]

Input: A degree q and a prime p of the form $p = 2^a 3^b 5^c f - 1$, where f is a small cofactor.
Output: A representation of an isogeny $\phi : E_0 \to E_A$ of degree $q(2^a - q)$.
1: Sample a random left $\mathcal{O}_0$-ideal I of norm $q(2^a - q)$. ▷ Using [2, Alg. 3]
2: Translate I into $\phi : E_0 \to E_A$. ▷ Using [2, Alg. 2]
3: **return** ϕ

larger than p. In our case, this is not directly applicable since the value 2^a is not large enough.

To avoid the problem, we use the entire torsion available: we first generate an endomorphism θ of E_0 of degree $q(2^a - q)3^b 5^c$ by using the FullRepresentInteger algorithm [22, Alg. 1]. Then, we compute the $3^b 5^c$-isogeny $\eta : E_0 \to E_A$ that backtracks θ, i.e. η is such that $\eta \circ \theta = [3^b 5^c]\phi$, for some isogeny $\phi : E_0 \to E_A$ of degree $q(2^a - q)$.

The codomain E_A of η is the public key curve since it is also the codomain of ϕ. Lastly, we need to evaluate points under ϕ to obtain the torsion points in the public key. Since $\eta \circ \theta = [3^b 5^c]\phi$, we have $\phi = [1/(3^b 5^c)]\eta \circ \theta$, which allows us to evaluate ϕ on any point of order coprime with $3^b 5^c$. Hence, we can obtain the images $\phi(P_0), \phi(Q_0)$ on the 2^a-torsion and thus a higher-dimensional representation of ϕ. From such a representation, using Theorem 2 we can evaluate ϕ on points of any order, including R_0, S_0 and X_0, Y_0. The isogeny sampling is summarized in Algorithm 3.

Algorithm 3. Generating a $q(2^a - q)$-isogeny (heuristic) based on [40, Alg. 2]

Input: A degree q, a prime p of the form $p = 2^a 3^b 5^c f - 1$.
Output: A representation of an isogeny $\phi : E_0 \to E_A$ of degree $q(2^a - q)$.
1: Generate an endomorphism θ of E_0 of degree $q(2^a - q)3^b 5^c$.
2: Compute $P', Q' = \theta(P_0), \theta(Q_0)$.
3: Let $K = \ker(\hat{\theta}) \cap E_0[3^b 5^c]$.
4: Compute $\eta : E_0 \to E_A$ with kernel K.
5: Compute $P_A, Q_A = [1/(3^b 5^c)]\eta(P'), [1/(3^b 5^c)]\eta(Q')$.
6: Compute Φ with kernel $\langle ([-q]P_0, P_A), ([-q]Q_0, Q_A) \rangle$.
7: **return** Φ

4.2 Encryption

Encryption follows an ElGamal approach: the sender first generates a commutative diagram and obtains a shared secret X_{AB}, Y_{AB}, as shown in Sect. 3, and then the ciphertext includes $ct' := m \oplus \mathrm{KDF}(X_{AB}, Y_{AB})$, for some fixed key-derivation function KDF.

More precisely, the sender samples a random integer r_3 in $\mathbb{Z}_{3^b}$ and computes the parallel isogenies $\psi : E_0 \to E_B$ and $\psi' : E_A \to E_{AB}$, respectively with kernel

$\langle R_0 + [r_3]S_0 \rangle$ and with kernel $\langle R_A + [r_3]S_A \rangle$. They also sample a random integer $\omega_2 \leftarrow \mathbb{Z}_{2^a}^{\times}$ as well as a random determinant-one matrix $\mathbf{D}_5 \leftarrow\$ \mathrm{SL}_2(\mathbb{Z}_{5^c})$.[4] Then, the sender computes the points

$$\begin{bmatrix} P_B \\ Q_B \end{bmatrix} = \begin{bmatrix} \omega_2 & 0 \\ 0 & 1/\omega_2 \end{bmatrix} \begin{bmatrix} \psi(P_0) \\ \psi(Q_0) \end{bmatrix}, \qquad \begin{bmatrix} X_B \\ Y_B \end{bmatrix} = \mathbf{D}_5 \begin{bmatrix} \psi(X_0) \\ \psi(Y_0) \end{bmatrix}$$

on the curve E_B and similarly obtains

$$\begin{bmatrix} P_{AB} \\ Q_{AB} \end{bmatrix} = \begin{bmatrix} \omega_2 & 0 \\ 0 & 1/\omega_2 \end{bmatrix} \begin{bmatrix} \psi'(P_A) \\ \psi'(Q_A) \end{bmatrix}, \qquad \begin{bmatrix} X_{AB} \\ Y_{AB} \end{bmatrix} = \mathbf{D}_5 \begin{bmatrix} \psi'(X_A) \\ \psi'(Y_A) \end{bmatrix}$$

on the curve E_{AB}.

The ciphertext consists of the curve E_B, together with the points (P_B, Q_B), (X_B, Y_B), the curve E_{AB}, together with the points P_{AB}, Q_{AB}, and the message-encoding component $ct' = m \oplus \mathrm{KDF}(X_{AB}, Y_{AB})$. The encryption procedure is summarized in Algorithm 4.

Algorithm 4. Encryption

Input: A message m, a public key E_A, P_A, Q_A, R_A, S_A, X_A, Y_A.
Output: A ciphertext ct.

1: Sample a random $r_3 \leftarrow\$ \mathbb{Z}_{3^b}$
2: Compute the isogeny $\psi : E_0 \to E_B$ with kernel $\langle R_0 + [r_3]S_0 \rangle$.
3: Compute the isogeny $\psi' : E_A \to E_{AB}$ with kernel $\langle R_A + [r_3]S_A \rangle$.
4: Sample a random $\omega_2 \leftarrow\$ \mathbb{Z}_{2^a}^{\times}$.
5: Compute $P_B = [\omega_2]\psi(P_0)$, $Q_B = [1/\omega_2]\psi(Q_0)$.
6: Compute $P_{AB} = [\omega_2]\psi'(P_A)$, $Q_{AB} = [1/\omega_2]\psi'(Q_A)$.
7: Sample a random $\mathbf{D}_5 \leftarrow\$ \mathrm{SL}_2(\mathbb{Z}_{5^c})$.
8: Compute $[X_B, Y_B]^T = \mathbf{D}_5\,[\psi(X_0), \psi(Y_0)]^T$.
9: Compute $[X_{AB}, Y_{AB}]^T = \mathbf{D}_5\,[\psi'(X_B), \psi'(Y_B)]^T$.
10: Compute $\mathsf{ct}' = \mathrm{KDF}(X_{AB}, Y_{AB}) \oplus m$.
11: **return** $\mathsf{ct} = \big((E_B, P_B, Q_B, X_B, Y_B), (E_{AB}, P_{AB}, Q_{AB}), ct'\big)$

Remark 5. During encryption, the diagonal scaling of the 2^a-torsion uses scalars $\omega_2, 1/\omega_2$ rather than two independent scalars ω_2, ω_2' because from the Weil pairing $e_{2^a}(P_B, P_{AB})$ an attacker may recover $\omega_2\omega_2' 3^b$. Since the degree 3^b is known, the attacker may always rescale points so that they are scaled by ω_2 and $1/\omega_2$. During key generation, the degree is secret, and thus the scaling uses α_2, β_2: if the scaling were done with $\alpha_2, 1/\alpha_2$, the pairing would leak the secret degree.

[4] Here, we recall that $\mathrm{SL}_2(\mathbb{Z}_{5^c})$ denotes the special linear group, the group of 2×2 matrices with determinant one and entries in $\mathbb{Z}_{5^c}$.

4.3 Decryption

At a high level, decryption consists of recomputing the shared secret X_{AB}, Y_{AB}, as described in Sect. 3, and extracting the message from $\mathsf{ct}' = \mathrm{KDF}(X_{AB}, Y_{AB}) \oplus m$.

To compute the pushforward isogeny $\phi' = \psi_* \phi$ and obtain the shared secret from a ciphertext $\big((E_B, P_B, Q_B, X_B, Y_B), (E_{AB}, P_{AB}, Q_{AB}), ct'\big)$, the receiver first computes the points $P'_{AB}, Q'_{AB} = [1/\alpha_2]P_{AB}, [1/\beta_2]Q_{AB}$. By the commutativity of the diagram, i.e. $\phi'\psi = \psi'\phi$, it follows that P'_{AB}, Q'_{AB} are the exact images of P_B, Q_B under ϕ'. The sender thus computes the two-dimensional isogeny $\Phi' : E_B \times E_{AB} \to F \times F'$ with kernel

$$\big\langle \big([-q]P_B, P'_{AB}\big), \big([-q]Q_B, Q'_{AB}\big) \big\rangle,$$

which by Kani's theorem (Theorem 2) can be expressed in matrix form as

$$\Phi' = \begin{pmatrix} \phi_1 & -\hat{\phi}_2 \\ * & * \end{pmatrix} : E_B \times E_{AB} \to F \times F',$$

where $\phi_1 : E_B \to F$ and $\phi_2 : F \to E_{AB}$ are a decomposition of ϕ', i.e. $\phi' = \phi_2 \circ \phi_1$, with $\deg \phi_1 = q$ and $\deg \phi_2 = 2^a - q$. Now, Φ' is a higher-dimensional representation of ϕ', which allows the receiver to evaluate ϕ' on X_B, Y_B. More precisely, the receiver first evaluates

$$(X'_B, \_) = \Phi'(X_B, \mathcal{O}_{E_{AB}}) \quad \text{and} \quad (Y'_B, \_) = \Phi'(Y_B, \mathcal{O}_{E_{AB}})$$

to obtain the images of X_B, Y_B on the middle curve F; then, they generate a basis U, V of $E_{AB}[5^c]$ and similarly map it to the middle curve by computing

$$(U', \_) = \Phi'(\mathcal{O}_{E_B}, U) \quad \text{and} \quad (V', \_) = \Phi'(\mathcal{O}_{E_B}, V).$$

The receiver then finds a change-of-basis matrix[5] that maps X'_B, Y'_B to U', V', i.e. they find coefficients x, y, w, z such that

$$X'_B = [x]U' + [y]V', \qquad Y'_B = [w]U' + [z]V'.$$

Lastly, the receiver obtains $\phi'(X_B)$ as $[2^a - q]([x]U + [y]V)$ and $\phi'(Y_B)$ as $[2^a - q]([w]U + [z]V)$ (the $[2^a - q]$ factor accounts for the dual of ϕ_2 that is implicitly computed), from which they can obtain $X_{AB} = [\delta_5]\phi'(X_B)$ and $Y_{AB} = [\delta_5]\phi'(Y_B)$. The decryption algorithm is summarized in Algorithm 5, while the entire POKÉ protocol is depicted in Fig. 2.

Remark 6. The proposed method to evaluate ϕ', given its higher-dimensional representation, requires solving discrete logarithms in $\mathbb{Z}_{5^c}$. Since 5^c is a smooth quantity, solving the discrete logarithm is efficient. However, we could relax the

[5] Finding a change-of-basis matrix amounts to solving four discrete logarithms modulo 5^c.

Algorithm 5. Decryption

Input: $\mathsf{sk} = (q, \alpha_2, \beta_2, \delta_5)$, $\mathsf{ct} = \big((E_B,\, P_B, Q_B,\, X_B, Y_B),\, (E_{AB},\, P_{AB}, Q_{AB}), ct'\big)$
Output: A message m'

Compute $P'_{AB} = [1/\alpha_2]P_{AB},\ Q'_{AB} = [1/\beta_2]Q_{AB}$
Compute Φ' with kernel $\langle([-q]P_B, P'_{AB}), ([-q]Q_B, Q'_{AB})\rangle$
Evaluate $(X'_B, \_) = \Phi'(X_B, \mathcal{O}_{E_{AB}})$
Evaluate $(Y'_B, \_) = \Phi'(Y_B, \mathcal{O}_{E_{AB}})$
Generate a basis U, V of $E_{AB}[5^c]$
Evaluate $(U', \_) = \Phi'(\mathcal{O}_{E_B}, U)$
Evaluate $(V', \_) = \Phi'(\mathcal{O}_{E_B}, V)$
Find x, y such that $X'_B = [x]U' + [y]V'$
Find w, z such that $Y'_B = [w]U' + [z]V'$
Compute $X_{AB} = [\delta_5][2^a - q]([x]U + [y]V)$
Compute $Y_{AB} = [\delta_5][2^a - q]([w]U + [z]V)$
Compute $m' = \mathrm{KDF}(X_{AB}, Y_{AB}) \oplus \mathsf{ct}'$
return m'

smoothness requirement by avoiding computations of discrete logarithms. We now explain this alternative approach.

It is possible to evaluate $\phi' = \phi_2 \circ \phi_1$ given its higher-dimensional representation $\Phi' : E_B \times E_{AB} \to F \times F'$ by evaluating the isogeny Φ' on the 2^a-torsion on E_B and E_{AB}. This gives a representation of a $q(2^a - q)$-isogeny from $F \to F'$, whose matricial form contains the isogeny ϕ_2. By combining the two higher-dimensional representations, any point —regardless of its order— can be mapped through. However, this approach is computationally expensive since it requires computing and evaluating two-dimensional isogenies; for this reason, we prefer the former approach, despite the disadvantages (as discussed in the last paragraph of Sect. 4.4).

Correctness. Correctness follows from the commutativity of the diagram, as argued in Sect. 3. The isogenies computed by two parties commute, i.e. we have $\phi'\psi = \psi'\phi$. Moreover, since isogenies are group homomorphisms, it remains true that

$$[d_1]\psi'([\delta_5]\phi(X_0)) + [d_2]\psi'([\delta_5]\phi(Y_0)) = X_{AB} = [\delta_5]\phi'([d_1]\psi(X_0) + [d_2]\psi(Y_0)),$$

where the left-hand side and the right-hand side are the points computed respectively by the receiver and by the sender if d_1 and d_2 are the first two entries in the matrix $\mathbf{D}_5$. A similar argument shows that both parties also compute the same point Y_{AB}, which guarantees the correctness of the protocol.

4.4 Optimizations and Trade-Offs

We briefly present some potential optimizations or trade-offs in the design of POKÉ that provide additional security, lead to smaller parameters at the cost of

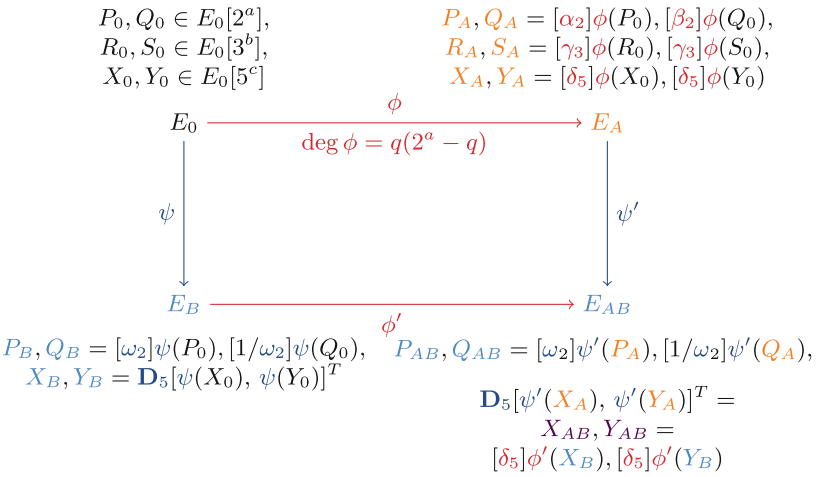

Fig. 2. The POKÉ protocol. Public parameters are in black; elements in red are part of the secret key or computed during decryption with the secret key, while elements in orange are part of the public key. Elements in blue are ephemeral secrets used during encryption and elements in light blue are part of the ciphertext. The shared secret points are in purple. (Color figure online)

slower computations, or introduce a speed-up in the computations of one party at the expense of the other (i.e. they obtain faster encryption at the cost of slower decryption, or vice versa).

Alternative Key Generation. The PKE protocol, as presented in Sect. 4, starts from a curve E_0 whose endomorphism ring is known, since knowledge of the endomorphism ring is used to compute an isogeny of non-smooth degree during key generation. While the public endomorphism ring of E_0 does not affect the security of the protocol, future cryptanalytic breakthroughs may change the situation. As a possible defense-in-depth measure, it may be prudent to replace E_0 with a curve of unknown endomorphism ring, even if this comes at the cost of a larger public key.

It is easy to adapt the protocol to generate and work with starting curves of unknown endomorphism ring: key generation can start as described from E_0 (the curve with j-invariant 1728) to obtain $\phi : E_0 \to E_A$ and the points R_A, S_A. Then, the receiver can construct a POKÉ diagram by themselves: they sample random ψ, compute ψ', and obtain $\phi' : E_B \to E_{AB}$. This is an isogeny of degree $q(2^a - q)$ between two curves of unknown endomorphism ring, which can thus act as a secret key. To complete the generation of the public key, the receiver needs to generate a basis of 3^b-torsion on E_B, map it to E_{AB}, and scale it accordingly. The corresponding public key is then the curves E_B and E_{AB}, together with the corresponding torsion points on them.

Hence, with this modified key generation, Alice can generate a public key where the starting curve has unknown (to anyone else) endomorphism ring; this, however, comes at the cost of a longer key generation procedure and a larger public key.

Higher-Dimensional Representations. Similarly to SQIsignHD [16], it is possible to use a four-dimensional representation. The isogenies ϕ and ϕ' would no longer need to have degree of the form $q(2^a - q)$; instead, the degree would only need to be chosen such that $2^a - \deg \phi$ can be expressed as a sum of squares. This has several benefits: since the degree does not have a prescribed shape, it is no longer necessary to scale the 2^a-torsion diagonally (see the *Exploiting the degree structure* subsection in Sect. 6), which means that all torsion points are scaled with the same scalar. This leads to a simpler protocol and a cleaner security assumption. Moreover, since scalar matrices commute with any another matrix, we conjecture that POKE4D would be resistant against adaptive attacks (certainly, the attack presented in [38] no longer applies).

A four-dimensional representation also leads to smaller parameters: Robert [45] showed that a four-dimensional representation of a n^2-isogeny only requires torsion points of order n. Thus, the quantity 2^a could chosen to be $\approx 2^{\lambda/2}$, compared to the current choice of $a \approx \lambda$. Such a change leads to a smaller prime ($\log p \approx 17\lambda/6$, compared to the current $\log p \approx 10\lambda/3$), which in turn translates to smaller public keys and ciphertexts, and faster arithmetic. Thus, encryption would be faster (since this change does not affect encryption), but decryption would be slower due to the increased computational cost of higher-dimensional representations [15].

B-SIDH Approach. Unlike many other protocols that rely on higher-dimensional representations, POKÉ is also well-suited for a B-SIDH approach [14]: if $\deg(\psi) = 3^b$ is chosen such that $3^b \mid p^2 - 1$ (rather than $3^b \mid p + 1$), the prime p could be considerably smaller. This requires that the kernel of ψ is defined over $\mathbb{F}_{p^4}$, but by working with x-only arithmetic and using elliptic curves and their quadratic twists, it is possible to keep all computations over $\mathbb{F}_{p^2}$. Nonetheless, such an approach would require choosing $\deg(\psi)$ to be significantly less smooth and finding smaller primes p such that $2^a 5^c \deg(\psi) \mid p^2 - 1$ is not straightforward, although recent work [13] proposed several possible primes for POKÉ based on an earlier draft of this work. Overall, such a change would lead to smaller public keys and ciphertexts, and both key generation and decryption could be significantly faster due to the smaller characteristic. Since the degree of ψ would not be as smooth, however, computing the encryption isogenies may be slower, but the slowdown would be partially compensated by the faster arithmetic.

Alternative Shared Secret Points. POKÉ uses two shared secret points X_{AB}, Y_{AB} of smooth order and defined over $\mathbb{F}_{p^2}$. It is also possible to choose a

single point of non-smooth order and defined over $\mathbb{F}_{p^4}$ (indeed, this was the choice made in an earlier draft of this work).[6] The advantages of such an approach is that the prime is smaller (5^c no longer needs to divide $p+1$), which leads to smaller (uncompressed) ciphertexts and faster arithmetic; despite their definition over $\mathbb{F}_{p^4}$, it is possible to keep all computations in $\mathbb{F}_{p^2}$ by working with quadratic twists and x-only arithmetic (although it would be necessary to adapt $(2,2)$-isogeny formulas to work with x-only arithmetic). However, if the shared secret points have non-smooth order, it is necessary to use a more complex procedure to evaluate ϕ and ϕ' (see Remark 6), which increases the computational time of decryption. Moreover, the shared secret points having smooth order is necessary to compress them (point compression requires solving discrete logarithms), which in turn leads to smaller compressed ciphertexts.

5 Security

The security of POKÉ, similarly to that of other PKEs with an ElGamal structure, hinges on the hardness of the computational problem that asks to find the shared secret given all public details. In our context, the computation problem is the following.

Problem 7 (C-POKE). Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, with known endomorphism ring. Let P_0, Q_0 be a basis of $E_0[2^a]$, R_0, S_0 a basis of $E_0[3^b]$, and X_0, Y_0 a basis of $E_0[5^c]$.

Let $\phi : E_0 \to E_A$ be an isogeny of degree $q(2^a - q)$, for some unknown value $q \in [0, 2^a]$. Write

- $P_A, Q_A = [\alpha_2]\phi(P_0), [\beta_2]\phi(Q_0)$,
- $R_A, S_A = [\gamma_3]\phi(R_0), [\gamma_3]\phi(S_0)$,
- $X_A, Y_A = [\delta_5]\phi(X_0), [\delta_5]\phi(Y_0)$,

where $\alpha_2, \beta_2, \gamma_3, \delta_5$ are uniformly random scalars from $\mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{3^b}^\times \times \mathbb{Z}_{5^c}^\times$.

Let $\psi : E_0 \to E_B$ be an isogeny of degree 3^b, and write $\psi' : E_A \to E_{AB}$ for its pushfoward $\phi_*\psi$. Write

- $P_B, Q_B = [\omega_2]\psi(P_0), [1/\omega_2]\psi(Q_0)$,
- $P_{AB}, Q_{AB} = [\omega_2]\psi'(P_A), [1/\omega_2]\psi'(Q_A)$
- $X_B, Y_B = \mathbf{D}_5\,[\psi(X_0), \psi(Y_0)]^T$,
- $X_{AB}, Y_{AB} = \mathbf{D}_5\,[\psi'(X_A), \psi'(Y_A)]^T$,

where ω_2 is a uniformly random scalar from $\mathbb{Z}_{2^a}^\times$ and $\mathbf{D}_5$ is a uniformly random matrix from $\mathrm{SL}_2(\mathbb{Z}_{5^c})$.

Given $\big(E_0, (P_0, Q_0), (R_0, S_0), (X_0, Y_0)\big)$, $\big(E_A, (P_A, Q_A), (R_A, S_A), (X_A, Y_A)\big)$, $\big(E_B, (P_B, Q_B), (X_B, Y_B)\big)$, and $\big(E_{AB}, (P_{AB}, Q_{AB})\big)$, compute the points X_{AB}, Y_{AB}.

[6] Three binary alternatives are possible: one vs two points, smooth vs non-smooth order, defined over $\mathbb{F}_{p^2}$ vs $\mathbb{F}_{p^4}$. While there are thus eight possible choices, the two presented above (two points, smooth order, defined over $\mathbb{F}_{p^2}$; and one point, non-smooth order, defined over $\mathbb{F}_{p^4}$) seem to be optimal.

The IND-CPA security of the protocol is formalized in the following theorem.

Theorem 8. *The POKÉ protocol, where the key-derivation function* KDF *is modeled as a random oracle, is IND-CPA secure in the random oracle model under the assumption that the C-POKE problem is hard.*

Proof. The proof is analogous to the proof of security of the hashed ElGamal PKE, see for instance [32, Theorem 1]. Given an adversary $\mathcal{A}$ that breaks the IND-CPA security of the proposed protocol, it is possible to construct an adversary $\mathcal{B}$ that solves an instance of the C-POKE problem. The adversary $\mathcal{B}$ simulates the random oracle model and sends a POKÉ ciphertext $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ to $\mathcal{A}$, where ct_0 is the data given by Problem 7 and ct_1 is randomly sampled: if $\mathcal{A}$ does not query the ROM with X_{AB}, Y_{AB} it cannot win the IND-CPA game with non-negligible advantage, and if it does, $\mathcal{B}$ can output a random query. With non-negligible probability (since X_{AB}, Y_{AB} are guaranteed to be among the queries), $\mathcal{B}$ outputs the solution to the C-POKE problem.

Similarly to the hashed ElGamal PKE and all analogous constructions, including SiGamal [39], it is also possible to construct an IND-CPA secure PKE in the standard model from the hardness of the decisional variant of the C-POKE problem, D-POKE, which is the following.

Problem 9 (D-POKE). Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, with known endomorphism ring. Let P_0, Q_0 be a basis of $E_0[2^a]$, R_0, S_0 a basis of $E_0[3^b]$, and X_0, Y_0 a basis of $E_0[5^c]$.

Let $\phi : E_0 \to E_A$ be an isogeny of degree $q(2^a - q)$, for some unknown value $q \in [0, 2^a]$. Write

- $P_A, Q_A = [\alpha_2]\phi(P_0), [\beta_2]\phi(Q_0)$,
- $R_A, S_A = [\gamma_3]\phi(R_0), [\gamma_3]\phi(S_0)$,
- $X_A, Y_A = [\delta_5]\phi(X_0), [\delta_5]\phi(Y_0)$,

where $\alpha_2, \beta_2, \gamma_3, \delta_5$ are uniformly random scalars from $\mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{3^b}^\times \times \mathbb{Z}_{5^c}^\times$.

Let $\psi : E_0 \to E_B$ be an isogeny of degree 3^b, and write $\psi' : E_A \to E_{AB}$ for its pushfoward $\phi_*\psi$. Write

- $P_B, Q_B = [\omega_2]\psi(P_0), [1/\omega_2]\psi(Q_0)$,
- $P_{AB}, Q_{AB} = [\omega_2]\psi'(P_A), [1/\omega_2]\psi'(Q_A)$
- $X_B, Y_B = \mathbf{D}_5\,[\psi(X_0), \psi(Y_0)]^T$,
- $X^0, Y^0 = \mathbf{D}_5\,[\psi'(X_A), \psi'(Y_A)]^T$,

where ω_2 is a uniformly random scalar from $\mathbb{Z}_{2^a}^\times$ and $\mathbf{D}_5$ is a uniformly random matrix from $\mathrm{SL}_2(\mathbb{Z}_{5^c})$.

Lastly, let X^1, Y^1 be a uniformly random basis of $E_{AB}[5^c]$ with pairing $e(X^1, Y^1) = e(X_B, Y_B)^{\gamma_3^2 q(2^a - q)}$, and let $b \leftarrow\$ \{0, 1\}$ be a random bit.

Given $\big(E_0, (P_0, Q_0), (R_0, S_0), (X_0, Y_0)\big)$, $\big(E_A, (P_A, Q_A), (R_A, S_A), (X_A, Y_A)\big)$, $\big(E_B, (P_B, Q_B), (X_B, Y_B)\big)$, and $\big(E_{AB}, (P_{AB}, Q_{AB})\big)$, and two points X^b, Y^b, compute b.

It is possible to obtain a PKE that is IND-CCA secure (in the ROM) by applying a standard transformation, such as the Fujisaki-Okamoto transform [27, 29].

Heuristic Key Generation. The analysis presented so far applies to the rigorous key generation case. When the key generation is computed with the heuristic variant, we need to rely on an additional assumption that states that the generated keys are indistinguishable from uniformly random ones, as those generated by the rigorous approach. In other words, it states the hardness of the following problem.

Problem 10. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, with known endomorphism ring. Let P_0, Q_0 be a basis of $E_0[2^a]$, R_0, S_0 a basis of $E_0[3^b]$, and X_0, Y_0 a basis of $E_0[5^c]$. Given a POKÉ public key $\mathsf{pk} = \big(E_A, (P_A, Q_A), (R_A, S_A), (X_A, Y_A)\big)$, sampled with probability 1/2 from either distribution:

- $\mathcal{D}_0$, where pk is computed by Algorithm 1 using Algorithm 2, i.e. the curve E_A is uniformly random among the supersingular elliptic curves that are $q(2^a - q)$-isogenous to E_0;
- $\mathcal{D}_1$, where pk is computed by Algorithm 1 using Algorithm 3, i.e. the curve E_A is simultaneously $q(2^a - q)$-isogenous and $3^b 5^c$-isogenous to E_0;

distinguish which distribution the public key has been sampled from.

Resistance Against Backdoor Attacks. In [9], Castryck and Vercauteren propose a backdoor attack against FESTA: if the points P_0, Q_0 on the starting curve E_0 are chosen to be eigenvectors of a small endomorphism, it is possible in some instances to recover an isogeny from its action on P_0, Q_0, even when scaled diagonally.

In POKÉ, these attacks do not apply. It is not possible to target the receiver's secret isogeny since it has unknown degree: the backdoor attack reduces the scaled-torsion problem to an unscaled one, and the SIDH attacks depend on knowledge of the degree. In the case of the sender's isogenies, its degree, 3^b, is known: however, the attacks would not be applicable anyway because the isogeny degree is too large compared to the order of the points. Even assuming the optimal case, where the basis consists of eigenvectors of Frobenius or the $\sqrt{-1}$ endomorphism (which would be easy to check), the attack requires that the degree of the attacked isogeny is smaller than the order of the revealed torsion images, which equals 2^a. As discussed in Sect. 7.1, we will choose $2^a \approx \sqrt{3^b}$: the parameters are thus very far from enabling such an attack.

6 Hardness Analysis of the C-POKE Assumption

The security of the protocol relies on a novel assumption (the hardness of Problem 7). In this section, we analyze its conjectured hardness and propose several arguments in its support.

The C-POKE problem (Problem 7) is a CDH-like problem: given the public data produced by two parties, compute their shared secret. While it seems hard to formally reduce C-POKE to finding the secrets associated with either party, we believe that any realistic attack against C-POKE would need to recover either the isogeny ϕ or the isogeny ψ. This is commonly the case for CDH-like assumptions: in many instances, such as the somewhat similar CDH problem for isogeny-based group actions, there exists a formal reduction [28]; in other cases, where the CDH assumption was eventually broken, such as in the case of SIDH, the attack relied on recovering either party's secret key [7,35,45]. In our case, it is similarly reasonable to expect that the only attacks against C-POKE would recover the secret isogeny of either party: Problem 7 asks to recover the scaled images (under unknown scalars) of certain points of order 5^c under secret isogenies. However, the problem does not reveal the action of either isogeny on any point of order 5^c on the target curve E_{AB}; thus, it seems hard to be able to obtain the shared secret points X_{AB}, Y_{AB} without being able to evaluate either ϕ or ψ (which amounts to fully recovering one of the secret isogenies). For these reasons, we focus our attention on the key-recovery problem for the sender and the receiver.

6.1 Receiver's Security

The security of the receiver against key-recovery attacks depends on the following problem:[7]

Problem 11. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, with known endomorphism ring, and write (P_0, Q_0) and (R_0, S_0) for bases of $E_0[2^a]$ and $E_0[3^b 5^c]$, respectively.

Let q be a random integer in $[0, 2^a]$ such that $q(2^a - q)$ is coprime with $3^b 5^c$, and let $\phi : E_0 \to E_1$ be a random isogeny of degree $q(2^a - q)$. Let $\alpha_2, \beta_2, \gamma_{3,5}$ be uniformly random values sampled in $\mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{3^b 5^c}^\times$.

Given $\big(E_0, (P_0, Q_0), (R_0, S_0)\big)$ and $\big(E_1, (P_1, Q_1), (R_1, S_1)\big)$, where

$$\begin{bmatrix} P_1 \\ Q_1 \end{bmatrix} = \begin{bmatrix} \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix} \begin{bmatrix} \phi(P_0) \\ \phi(Q_0) \end{bmatrix}, \qquad \begin{bmatrix} R_1 \\ S_1 \end{bmatrix} = \begin{bmatrix} \gamma_{3,5} & 0 \\ 0 & \gamma_{3,5} \end{bmatrix} \begin{bmatrix} \phi(R_0) \\ \phi(S_0) \end{bmatrix},$$

recover ϕ.

Problem 11 is also a novel problem, although it has connections to more established problems. On one end, it is similar to the CIST problem [4, Prob. 7], which reveals torsion information scaled diagonally (similarly to P_1 and Q_1); however, in the CIST problem there is no equivalent to R_1, S_1, although — crucially— the degree of the isogeny is known. On the other hand, Problem 11 shares similarities with the MD-SIDH problem [26, Prob. 5], which also uses an

[7] For simplicity, the computational problem compresses the points R, S and X, Y used in the protocol into a single pair of points. The points R_0, S_0 used in the problem thus refer to the points $R_0 + X_0, S_0 + Y_0$ in the protocol.

isogeny of unknown degree (but a multiple of the degree is known) and reveals information similarly to R_1, S_1, without any diagonal torsion information.

We identify and discuss several potential attacks, and we explain why they do not apply.

Exploiting the Torsion Information. Fundamentally, the hardness of the problem relies on the fact that the isogeny ϕ has unknown degree: all known attacks that exploit torsion point information to recover an isogeny [7,9,26, 35,45] rely extensively on the knowledge of the degree of the secret isogeny. Extending the known attacks to work with isogenies of unknown degree would require a substantially different approach.

Recovering the Secret Degree. The torsion images that are revealed in Problem 11 do not provide any help in recovering the degree $q(2^a - q)$: the only known technique that relates the degree and the torsion information relies on pairings. Indeed, we have that

$$e_{2^a}(P_1, Q_1) = e_{2^a}(P_0, Q_0)^{\alpha_2 \beta_2 q(2^a - q)} \quad \text{and}$$
$$e_{3^b 5^c}(R_1, S_1) = e_{3^b 5^c}(R_0, S_0)^{\gamma_{3,5}^2 q(2^a - q)}.$$

Since the pairings $e_{2^a}(P_0, Q_0)$ and $e_{3^b 5^c}(R_1, S_1)$ are easily computable, an attack may recover the quantities $\alpha_2 \beta_2 q(2^a - q) \bmod 2^a$ and $\gamma_{3,5}^2 q(2^a - q) \bmod 3^b 5^c$ through discrete logarithm computations. Nonetheless, α_2, β_2, and $\gamma_{3,5}$ are sampled uniformly at random: thus, the first quantity ($\alpha_2 \beta_2 q(2^a - q) \bmod 2^a$) does not contain, information theoretically, any information on the degree, while the second one ($\gamma_{3,5}^2 q(2^a - q) \bmod 3^b 5^c$) only reveals one bit of information on the degree: its quadratic residuosity (since $\gamma_{3,5}^2$ is clearly a square, $\gamma_{3,5}^2 q(2^a - q)$ is a square modulo $3^b 5^c$ only if $q(2^a - q)$ is itself a square).

An attacker may thus choose to brute-force the correct value of q: they could guess a degree of the form $q'(2^a - q')$ and obtain a higher-dimensional representation of ϕ from its action on the $3^b 5^c$-torsion (this would require using a four- or eight-dimensional representation). By choosing a sufficiently large value of 2^a, the cost of such an attack is exponentially large.

Exploiting the Degree Structure. While the degree is secret, an attacker may hope to exploit the structure in the degree, i.e. the fact that the degree is of the form $q(2^a - q)$ (which is used to obtain a two-dimensional representation). Indeed, this would be a valid attack direction if the action of ϕ on the 2^a-torsion were revealed *without diagonal scaling*. To see how the attack proceeds, assume we are given a curve E_0, with a basis P_0, Q_0 of $E_0[2^a]$, and E_1, with $P_1 = [\alpha_2]\phi(P_0)$ and $Q_1 = [\alpha_2]\phi(Q_0)$ (note that, unlike in POKÉ, the same scalar is used for both points). From the pairings of the two torsion bases, we can obtain $\alpha_2^2 q(2^a - q) \bmod 2^a$, which is equivalent to $-\alpha_2^2 q^2 \bmod 2^a$, from which we

can extract $\alpha_2 q \bmod 2^a$. This is sufficient to recover the isogeny ϕ, because its two-dimensional representation is given by an isogeny Φ with kernel

$$\begin{aligned}\ker \Phi &= \big\langle \big([-q]P_0, \phi(P_0)\big), \big([-q]Q_0, \phi(Q_0)\big)\big\rangle \\ &= \big\langle \big([-\alpha_2 q]P_0, \underbrace{[\alpha_2]\phi(P_0)}_{=P_1}\big), \big([-\alpha_2 q]Q_0, \underbrace{[\alpha_2]\phi(Q_0)}_{=Q_1}\big)\big\rangle.\end{aligned}$$

The same attack does not apply when the scaling of the 2^a-torsion is diagonal because, as mentioned above, the pairing leakage is $-\alpha_2\beta_2 q^2$, which information theoretically hides both q and $\alpha_2 q$.

Lastly, it is natural to wonder whether the same attack could be extended to the 3^b5^c torsion: is it possible that the degree $q(2^a - q)$ is also expressible in the form $q'(B' - q')$, for any B' dividing 3^b5^c? While the probability of this is conceivably negligible, our choice of concrete parameters always prevents such a coincidence.[8]

6.2 Sender's Security

We now focus on the security of key-recovery attacks against the sender's isogenies. This is formalized in the following problem.

Problem 12. With the same setup as Problem 7, given $\big(E_0, (P_0, Q_0), (X_0, Y_0)\big)$, $\big(E_A, (P_A, Q_A), (X_A, Y_A)\big)$, $\big(E_B, (P_B, Q_B), (X_B, Y_B)\big)$, and $\big(E_{AB}, (P_{AB}, Q_{AB})\big)$, compute the isogeny ψ.

This problem is very similar to the CIST^2 problem introduced in FESTA [4, Problem 8], with the main difference that in Problem 12 the isogenies ψ and ψ' are parallel (i.e. ψ' is the pushforward of ψ under ϕ), while they are not in the CIST^2 problem. We can formalize this similarity by showing there exists a reduction from Problem 12 to the following problem.

Problem 13. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, with known endomorphism ring. Let P_0, Q_0 be a basis of $E_0[2^a]$, Let $\phi : E_0 \to E_A$ be an isogeny of degree N. Let $\psi : E_0 \to E_B$ be an isogeny of degree 3^b, and write $P_B = [\omega_2]\psi(P_0)$, $Q_B = [1/\omega_2]\psi(Q_0)$, where ω_2 is a uniformly random scalar in $\mathbb{Z}_{2^a}^\times$.

Given $\big(E_0, P_0, Q_0, \phi\big)$ and $\big(E_B, P_B, Q_B, \psi_*\phi\big)$, compute ψ.

Theorem 14. *Given an adversary $\mathcal{A}$ that solves Problem 12, there exists an adversary $\mathcal{B}$ that solves an instance $\big((E_0, P_0, Q_0, \phi), \big(E_B, P_B, Q_B, \phi'\big)\big)$ of Problem 13, where $N = q(2^a - q)$ in about the same time and memory as $\mathcal{A}$.*

[8] Assume there exists a q' and $b' < b, c' < c$ such that $q(2^a - q) = q'(3^{b'}5^{c'} - q')$. If $b' = 0$, $5^{c'} \ll 2^a$, which prevents the collision. If $b' \neq 0$, we have $q(2^a - q) \equiv q2^a - q^2 \equiv q2^a + 2 \bmod 3$, while $q'(3^{b'}5^{c'} - q') \equiv -q'^2 \equiv 2 \bmod 3$. The equality is thus never satisfied since $q2^a$ is not divisible by 3.

Proof. Consider the following adversary $\mathcal{B}$: the adversary $\mathcal{B}$ samples a random basis X_0, Y_0 of $E_0[5^c]$ and a random basis X_B, Y_B of $E_B[5^c]$ (this is possible because the sender masks the 5^c-torsion with a full 2×2 matrix). Then, they extract the codomains E_A and E_{AB} of ϕ and ϕ' (any efficient representation of an isogeny comes with an algorithm to compute its codomain), and they sample uniformly random scalars $\alpha_2, \beta_2, \delta_5 \leftarrow\$ \mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{2^a}^\times \times \mathbb{Z}_{5^c}^\times$. Lastly, $\mathcal{B}$ computes

- $P_A, Q_A = [\alpha_2]\phi(P_0), [\beta_2]\phi(Q_0)$,
- $X_A, Y_A = [\delta_5]\phi(X_0), [\delta_5]\phi(Y_0)$,
- $P_{AB}, Q_{AB} = [\alpha_2]\phi'(P_B), [\beta_2]\phi'(Q_B)$,

and they query $\mathcal{A}$ with $\big(E_0, (P_0, Q_0), (X_0, Y_0)\big)$, $\big(E_A, (P_A, Q_A), (X_A, Y_A)\big)$, $\big(E_B, (P_B, Q_B), (X_B, Y_B)\big)$, and $\big(E_{AB}, (P_{AB}, Q_{AB})\big)$. The adversary $\mathcal{B}$ then returns the same output produced by $\mathcal{A}$. Note that the query prepared by $\mathcal{B}$ is identically distributed as the input of an instance of Problem 12, and the output produced by $\mathcal{A}$ is a valid solution to the input instance of Problem 13. □

Now, Problem 13 is very similar to the CIST problem (or equivalently the SSIP-A problem from binSIDH [3, Prob. 1]). In both cases, we are given two curves and two sets of points scaled diagonally. In Problem 13, we are also given an isogeny and its pushforward, which can be represented by a generator of the kernel and its image, up to a scalar. Thus, Problem 13 (and thus also Problem 12, by Theorem 14) can be reduced to an instance of the CIST problem where the order of torsion points is $q(2^a - q)5^c$. The reduction is polynomial time whenever $q(2^a - q)$ is polynomially smooth (the reduction requires evaluating an isogeny of degree $q(2^a - q)$) or if the reduction is given one-time access to an oracle that can compute unsmooth-degree isogenies (similarly to what is done in the security proof of SQIsignHD and its variants [2,16,24,41]).

7 Implementation and Results

The proposed protocol is one of the most compact post-quantum PKEs and possibly the most efficient isogeny-based PKE to date.

7.1 Parameters

To guarantee the security of the protocol, we select

- $a \approx \lambda$: choosing $a \approx \lambda$ is sufficient to make brute-forcing q hard. While there are slightly less than 2^λ possible choices for q (since we require q and $2^a - q$ to be coprime with $2, 3, 5$), the brute-force attack first guesses q and then computes a four-dimensional isogeny using the 3^b-torsion to verify the correctness of the guess: the cost of evaluating such an isogeny makes up for the reduction in search space, which gives us a good margin of security.
- $3^b \approx 2^{2\lambda}$: choosing 3^b the degree of the encryption isogenies leads to a fast encryption procedure. Also, the quantity b is chosen so that meet-in-the-middle attacks to recover ϕ are exponentially expensive.

– $5^c \approx 2^{\lambda/3}$: we choose the order of the points to used the shared secret to be a power of five to enable quick evaluation of isogenies with a higher-dimensional representation (see Remark 6) and large enough to avoid brute-force attacks. If U, V is any basis of $E_{AB}[5^c]$, guessing the shared secret points X_{AB}, Y_{AB} amounts to guessing the correct change-of-basis matrix. The pairing of X_{AB}, Y_{AB} is known, which removes one degree of freedom: thus, guessing X_{AB}, Y_{AB} requires guessing three coefficients in $\mathbb{Z}_{5^c}$, hence the choice of $5^c \approx 2^{\lambda/3}$.

This leads to a prime size with $\log p \approx 10\lambda/3$. Thus, even without using the B-SIDH optimization (Sect. 4.4), POKÉ has one of the smallest characteristics of all isogeny-based encryption protocols (both PKEs and key exchanges). If such an optimization were to be used, a recent work [13, Table 7] found suitable primes for POKÉ at NIST-I security level with fewer than 4λ bits, even when the smooth divisor (i.e. the degree of ϕ) is 2^{12}-smooth.

The list of proposed parameters is reported in Table 1, together with the size of public keys and ciphertexts. If uncompressed, a public key consists of a curve and two points (we can reduce multiple points of coprime order into a single point by summing them), which thus requires $6 \log p$ bits to be represented; similarly, an uncompressed ciphertext consists of two sets of curves and points, and thus it can be represented with $12 \log p$ bits. Compressing a public key does not significantly reduce its size since it contains points of order $2^a 3^b 5^c \approx p$: relying on the expected pairing value to represent the points with only three coefficients reduces the public key from $6 \log p$ to $5 \log p$. The results are different for the ciphertext: a ciphertext consists of two curves E_B, E_{AB}, the 2^a-points P_B, Q_B and P_{AB}, Q_{AB}, and the 5^c-points X_B, Y_B. Since the 5^c-points are scaled by a full 2×2 matrix, it is possible to omit them by taking a deterministic basis on E_B as X_B, Y_B (in this way, the scaling matrix is implicitly defined). Moreover, the 2^a-points have small order, so they can be represented in only $3a$ bits, if they are expressed as three coefficients with respect to a deterministic basis. With such a compression, a ciphertext can be represented in only $4 \log p + 3a$ bits.

Table 1. Parameters and public-key and ciphertext size for POKÉ.

	NIST I	NIST III	NIST V
Prime	$2^{129} \cdot 3^{164} \cdot 5^{18} - 1$	$2^{192} \cdot 3^{243} \cdot 5^{28} \cdot 7^2 - 1$	$2^{256} \cdot 3^{324} \cdot 5^{36} \cdot 547 - 1$
$\log_2 p$	431	648	863
pk size (B)	324	486	648
ct size (B)	648	972	1296
$\mathsf{pk}_{\mathsf{cmp}}$ size (B)	270	405	540
$\mathsf{ct}_{\mathsf{cmp}}$ size (B)	264	396	528

7.2 Performance

We implemented[9] the uncompressed version of POKÉ (with the heuristic key generation) in SageMath [48] by using the implementation of isogenies between abelian surfaces proposed in [17]. The implementation is in a high-level language and lacks several possible optimizations, but it gives an indication of the performance profile of the protocol, and –as shown in the comparison below– it is enough to justify the claim that POKÉ is the fastest encryption protocol from isogeny assumptions. We also expect to obtain a very significant improvement in performance when switching to a low-level language, similar to the improvements reported in [17]. While a full optimized implementation is required to validate such claims, preliminary results of a Rust implementation support this claim.

Table 2. Performance of our SageMath proof-of-concept implementation of POKÉ. Timings are the average of 100 executions on an Apple M3 CPU.

	Timings (ms)		
	KeyGen	Encrypt	Decrypt
NIST-I	428	105	97
NIST-III	1001	209	194
NIST-V	1882	350	325

Comparison with Other Protocols. We start by considering CSIDH [8]: while the original proposal relied on a 511-bit prime (for $\lambda = 128$), subsequent analyses [5,10,44] of quantum attacks showed that such a prime is not sufficient to guarantee the expected security. Instead, [5] proposed to use a prime of size between 2260 or 5280 bits (depending on the conservativeness of the analysis), while [10] proposed to use a 4096-bit prime. In terms of performance, this comes with a significant cost. A recent work [6] reports optimized implementations of CSIDH: encrypting a message (i.e. computing two group action evaluations) takes more than a second with $p = 2048$ and more than several seconds with $p = 4096$, while decryption is only twice as fast.[10] This is significantly longer than the results of our proof-of-concept implementation in SageMath reported in Table 2, where encryption and decryption take about one hundred milliseconds.

If we compare our protocol with other isogeny-based constructions, the comparison is similarly positive. FESTA [4], as proposed, works with a 1292-bit prime for $\lambda = 128$ and requires much larger public keys and ciphertexts (between two and four times as large). FESTA, when implemented in SageMath with the

[9] The implementation is available at https://github.com/andreavico/POKE-PKE.

[10] The authors report their implementation results in [6, Table 5] only in clock cycles, but no CPU frequency is reported. Assuming a frequency of 3.2 GHz, one group action evaluation takes about 690 ms with $p = 2048$ and 3.5 s with $p = 4096$.

same library for isogeny computations [17] and running on the same hardware as Table 2, requires 3.1 s for encryption and 2.8 s for decryption, hence more than an order of magnitude slower than POKÉ. FESTA was later improved in QFESTA [40], which is currently the most efficient encryption protocol from isogenies. QFESTA uses a prime of roughly 3λ bits, similarly to POKÉ, but its compressed ciphertexts[11] are larger: in both protocols, the ciphertext consists of two curves and several points, but in QFESTA the order of the torsion points is much larger. Indeed, the QFESTA compressed ciphertexts are almost twice as large as the compressed ciphertexts in POKÉ, and the QFESTA running times (also implemented in SageMath, using the same library for higher-dimensional isogenies) are also an order of magnitude slower. In independent work to POKÉ, Moriya proposed LIT-SiGamal, which shares a similar structure with POKÉ: benchmarking both SageMath implementations on the same machine shows that LIT-SiGamal has a comparable encryption time (albeit slightly slower), but decryption is more than $5\times$ slower. Lastly, we compare our protocol with terSIDH-hyb, the most efficient isogeny-based key exchange [3]. In terSIDH-hyb, the prime characteristic is $\approx 4\times$ larger, which leads to larger public keys and ciphertexts; if terSIDH-hyb is used as PKE through a standard KE-to-PKE transform, encryption takes twice as long, while decryption is more than $15\times$ slower.

Hence, we expect POKÉ to be the most efficient isogeny-based encryption protocol, outperforming existing protocols in overall computation times (encryption and decryption) by at least an order of magnitude. Moreover, such an increase in computational efficiency does not come at the cost of public key or ciphertext sizes: POKÉ is also one of the most compact post-quantum encryption protocols, with the option of becoming even more compact by relying on compression techniques or using a B-SIDH-like approach.

8 Conclusion

We proposed a new PKE protocol, POKÉ, that relies on isogenies of unknown degree to build a commutative diagram where one party works with FESTA-like isogenies, represented by a two-dimensional representation, and the other party computes simple SIDH-like isogenies. The combination of these techniques, together with the use of unknown-degree isogenies, allows us to obtain the smallest prime characteristic of any isogeny-based encryption protocol (either public-key encryption or key exchange): in turn, the small characteristic and precise design choices that optimized performance led to the most efficient isogeny-based encryption protocol to date, improving on existing protocols by more than an order of magnitude.

In future work, we aim to further improve POKÉ and to develop an optimized implementation of it. Moreover, we are interested in exploring the applications of POKÉ in the design of new cryptosystems: POKÉ is a PKE, but its underlying

[11] The QFESTA paper only reports compressed ciphertext sizes. We expect that uncompressed ciphertexts are comparable in size between the two protocols.

structure is similar to a key-exchange protocol, which makes it well suited as a replacement for SIDH in advanced constructions (for instance, [1]) or new isogeny-based protocols with advanced functionalities.

References

1. Basso, A.: A post-quantum round-optimal oblivious PRF from isogenies. In: Carlet, C., Mandal, K., Rijmen, V. (eds.) SAC 2023. LNCS, vol. 14201, pp. 147–168. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-53368-6_8
2. Basso, A., et al.: SQIsign2D-west: the fast, the small, and the safer. In: ASIACRYPT 2024. Springer, Cham (2024). https://doi.org/10.1007/978-981-96-0891-1_11
3. Basso, A., Fouotsa, T.B.: New SIDH countermeasures for a more efficient key exchange. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VIII. LNCS, vol. 14445, pp. 208–233. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8742-9_7
4. Basso, A., Maino, L., Pope, G.: FESTA: fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 98–126. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8739-9_4
5. Bonnetain, X., Schrottenloher, A.: Quantum security analysis of CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 493–522. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_17
6. Campos, F., et al.: Optimizations and Practicality of High-Security CSIDH. Cryptology ePrint Archive, Paper 2023/793 (2023). https://eprint.iacr.org/2023/793
7. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 423–447. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_15
8. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 395–427. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_15
9. Castryck, W., Vercauteren, F.: A polynomial time attack on instances of M-SIDH and FESTA. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 127–156. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8739-9_5
10. Chávez-Saab, J., Chi-Domínguez, J.J., Jaques, S., Rodríguez-Henríquez, F.: The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. J. Cryptogr. Eng. **12**(3), 349–368 (2022). https://doi.org/10.1007/s13389-021-00271-w
11. Chen, M., Leroux, A., Panny, L.: SCALLOP-HD: group action from 2-dimensional isogenies. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part II. LNCS, vol. 14603, pp. 190–216. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-57725-3_7
12. Corte-Real Santos, M., Costello, C., Benjamin, S.: Efficient (3, 3)-isogenies on fast kummer surfaces (2025). https://doi.org/10.1007/s40993-024-00600-y
13. Corte-Real Santos, M., Eriksen, J.K., Meyer, M., Rodríguez-Henríquez, F.: Finding practical parameters for isogeny-based cryptography. Cryptology ePrint Archive, Report 2024/1150 (2024). https://eprint.iacr.org/2024/1150

14. Costello, C.: B-SIDH: Supersingular isogeny Diffie-Hellman using twisted torsion. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 440–463. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_15
15. Dartois, P.: Fast computation of 2-isogenies in dimension 4 and cryptographic applications. Cryptology ePrint Archive, Report 2024/1180 (2024). https://eprint.iacr.org/2024/1180
16. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: new dimensions in cryptography. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 3–32. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58716-0_1
17. Dartois, P., Maino, L., Pope, G., Robert, D.: An algorithmic approach to (2, 2)-isogenies in the theta model and applications to isogeny-based cryptography. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024. ASIACRYPT 2024. LNCS, vol 15486. Springer, Singapore (2025). https://doi.org/10.1007/978-981-96-0891-1_10
18. De Feo, L.: Mathematics of isogeny based cryptography (2017)
19. De Feo, L., et al.: SCALLOP: scaling the CSI-FiSh. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 345–375. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-31368-4_13
20. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Math. Cryptol. **8**(3), 209–247 (2014). https://doi.org/10.1515/jmc-2012-0015
21. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_3
22. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 659–690. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_23
23. Deuring, M.: Die typen der multiplikatorenringe elliptischer funktionenkörper. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 14, pp. 197–272. Springer, Heidelberg (1941)
24. Duparc, M., Fouotsa, T.B.: SQIPrime: a dimension 2 variant of SQISignHD with non-smooth challenge isogenies. In: ASIACRYPT 2024. Springer, Cham (2024). https://doi.org/10.1007/978-981-96-0891-1_13
25. Flynn, E.V., Ti, Y.B.: Genus two isogeny cryptography. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, pp. 286–306. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_16
26. Fouotsa, T.B., Moriya, T., Petit, C.: M-SIDH and MD-SIDH: countering SIDH attacks by masking information. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 282–309. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_10
27. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34
28. Galbraith, S.D., Lai, Y.F., Montgomery, H.: A simpler and more efficient reduction of DLog to CDH for abelian group actions. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part II. LNCS, vol. 14603, pp. 36–60. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-57725-3_2

29. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 341–371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_12
30. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
31. Kani, E.: The number of curves of genus two with elliptic differentials. Journal für die reine und angewandte Mathematik **485**, 93–122 (1997). https://doi.org/10.1515/crll.1997.485.93
32. Kiltz, E., Malone-Lee, J.: A general construction of IND-CCA2 secure public key encryption. In: Paterson, K.G. (ed.) Cryptography and Coding, pp. 152–166. Springer, Heidelberg (2003)
33. Kunzweiler, S., Ti, Y.B., Weitkämper, C.: Secret keys in genus-2 SIDH. In: AlTawy, R., Hülsing, A. (eds.) SAC 2021. LNCS, vol. 13203, pp. 483–507. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99277-4_23
34. LeGrow, J.T., Ti, Y.B., Zobernig, L.: Supersingular non-superspecial abelian surfaces in cryptography. Cryptology ePrint Archive, Report 2022/650 (2022). https://eprint.iacr.org/2022/650
35. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 448–471. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_16
36. Moriya, T.: IS-CUBE: an isogeny-based compact KEM using a boxed SIDH diagram. Cryptology ePrint Archive, Report 2023/1506 (2023). https://eprint.iacr.org/2023/1506
37. Moriya, T.: LIT-SiGamal: an efficient isogeny-based PKE based on a LIT diagram. Cryptology ePrint Archive, Report 2024/521 (2024). https://eprint.iacr.org/2024/521
38. Moriya, T., Onuki, H.: The wrong use of FESTA trapdoor functions leads to an adaptive attack. Cryptology ePrint Archive, Report 2023/1092 (2023). https://eprint.iacr.org/2023/1092
39. Moriya, T., Onuki, H., Takagi, T.: SiGamal: a supersingular isogeny-based PKE and its application to a PRF. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 551–580. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_19
40. Nakagawa, K., Onuki, H.: QFESTA: efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part V. LNCS, vol. 14924, pp. 75–106. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-68388-6_4
41. Nakagawa, K., et al.: SQIsign2D-east: a new signature scheme using 2-dimensional isogenies. In: ASIACRYPT 2024. Springer, Cham (2024). https://doi.org/10.1007/978-981-96-0891-1_9
42. Oudompheng, R., Pope, G.: A note on reimplementing the Castryck-Decru attack and lessons learned for SageMath. Cryptology ePrint Archive, Report 2022/1283 (2022). https://eprint.iacr.org/2022/1283
43. Page, A., Robert, D.: Introducing clapoti(s): evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766 (2023). https://eprint.iacr.org/2023/1766

44. Peikert, C.: He gives C-sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 463–492. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_16
45. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 472–503. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_17
46. Robert, D.: On the efficient representation of isogenies (a survey). Cryptology ePrint Archive, Report 2024/1071 (2024). https://eprint.iacr.org/2024/1071
47. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009)
48. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.1) (2024). https://www.sagemath.org
49. Vélu, J.: Isogénies entre courbes elliptiques. Comptes Rendus de l'Académie des Sciences de Paris **273**, 238–241 (1971). https://gallica.bnf.fr/ark:/12148/bpt6k56191248/f52.item
50. Yoshizumi, R., Onuki, H., Ohashi, R., Kudo, M., Nuida, K.: Efficient theta-based algorithms for computing (ℓ, ℓ)-isogenies on Kummer surfaces for arbitrary odd ℓ. Cryptology ePrint Archive, Paper 2024/1519 (2024). https://eprint.iacr.org/2024/1519