FESTA: Fast Encryption from Supersingular Torsion Attacks

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

[1] University of Bristol, Bristol, UK
{andrea.basso,luciano.maino}@bristol.ac.uk, giacomo.pope@nccgroup.com
[2] NCC Group, Cheltenham, UK

Abstract. We introduce FESTA, an efficient isogeny-based public-key encryption (PKE) protocol based on a constructive application of the SIDH attacks.

At its core, FESTA is based on a novel trapdoor function, which uses an improved version of the techniques proposed in the SIDH attacks to develop a trapdoor mechanism. Using standard transformations, we construct an efficient PKE that is IND-CCA secure in the QROM. Additionally, using a different transformation, we obtain the first isogeny-based PKE that is IND-CCA secure in the standard model.

Lastly, we propose a method to efficiently find parameters for FESTA, and we develop a proof-of-concept implementation of the protocol. We expect FESTA to offer practical performance that is competitive with existing isogeny-based constructions.

Keywords: Isogeny-based Cryptography · Public-key Encryption · Trapdoor Function

1 Introduction

Over the last decade, isogeny-based cryptography has become one of the major candidates to develop cryptographic protocols that are resistant against attacks from quantum computers. Isogeny-based solutions often offer practical execution times, and, despite being significantly slower than their lattice-based counterparts, they usually benefit from small bandwidth requirements.

The Supersingular Isogeny Diffie-Hellman (SIDH) protocol by De Feo, Jao, and Plût [22] has been the most well-known and efficient encryption protocol

Author list in alphabetical order; see https://www.ams.org//profession/leaders/CultureStatement04.pdf. The first author has been supported in part by EPSRC via grant EP/R012288/1, under the RISE (http://www.ukrise.org) programme. The second author has been supported by the UK Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training (CDT) in Trust, Identity, Privacy and Security in Large-scale Infrastructures (TIPS-at-Scale) at the Universities of Bristol and Bath.

J. Guo and R. Steinfeld (Eds.): ASIACRYPT 2023, LNCS 14444, pp. 98–126, 2023.
https://doi.org/10.1007/978-981-99-8739-9_4

based on isogenies. However, recent attacks [9,36,48] broke the security guarantees of the protocol. Fouotsa, Moriya, and Petit [26] proposed two countermeasures to these attacks, but the result requires significantly larger parameters which make the protocols impractically slow for most applications. Further countermeasures have been suggested by Basso and Fouotsa [3]; these countermeasures achieve better performance and smaller keys compared to [26].

The attacks on SIDH significantly altered the landscape of isogeny-based protocols: they similarly affected the security of other protocols that revealed torsion point information, such as SÉTA [21]. Other isogeny-based encryption schemes, such as CSIDH [10] and pSIDH [35], are unaffected; however, they are vulnerable to a quantum subexponential attack [42] and a quantum polynomial-time attack [12], respectively. This makes it hard to estimate the quantum security of a given parameter set; nonetheless, according to the conservative estimates in [42], CSIDH requires very large primes, which would increase the running time of a single key exchange to several seconds [14].

In this work, we aim to fill the gap by proposing a novel PKE protocol that is practical and efficient; we call it FESTA, for *Fast Encryption from Supersingular Torsion Attacks*. We first develop a trapdoor function, where the SIDH attacks are used to invert the one-way function. Then, we use the proposed trapdoor function to build a IND-CCA secure PKE.

In the trapdoor formulation, the trapdoor key is an isogeny $\varphi_A \colon E_0 \to E_A$ and a random special matrix $\mathbf{A}$; the public parameters are the codomain E_A, together with the image of a large torsion basis (P_b, Q_b) under φ_A. The image points, before being revealed, are scaled by the matrix $\mathbf{A}$, which protects the isogeny φ_A from the SIDH attacks. The one-way function receives as input two isogenies $\varphi_1 : E_0 \to E_1$, $\varphi_2 : E_A \to E_2$, and a random special matrix $\mathbf{B}$. Evaluating the function then consists in computing the images of the torsion basis on E_0 and E_A under φ_1 and φ_2, respectively, and scaling them both with the matrix $\mathbf{B}$; see Fig. 1. The matrices $\mathbf{A}$ and $\mathbf{B}$ are special in the sense that they commute; this is the case, for instance, for diagonal matrices. Commutativity of the matrices is what enables the trapdoor inversion: applying the inverse matrix $\mathbf{A}^{-1}$ to scale the points on E_2 yields the correct images of the torsion points on E_1 under the isogeny $\psi := \varphi_2 \circ \varphi_A \circ \widehat{\varphi}_1$. Hence, the SIDH attacks allow the trapdoor holder to recover the function input φ_1, φ_2, and the matrix $\mathbf{B}$, while the attacks are infeasible to anyone who does not know the secret matrix $\mathbf{A}$.

Related Work. FESTA can be considered a successor of SÉTA [21]: both protocols constructively use torsion-point attacks to develop a trapdoor function, which is then the foundation of a IND-CCA PKE. Despite the similarities, the two protocols rely on different techniques, and the efficiency of the SIDH attacks [41], compared to the torsion-point attacks used by SÉTA [44], allows us to obtain a practical encryption protocol.

In terms of techniques used, key generation and encryption in FESTA rely on similar computations as those in SIDH, with the key difference that the revealed torsion images are scaled to prevent the SIDH attacks. Unlike the scaling pro-

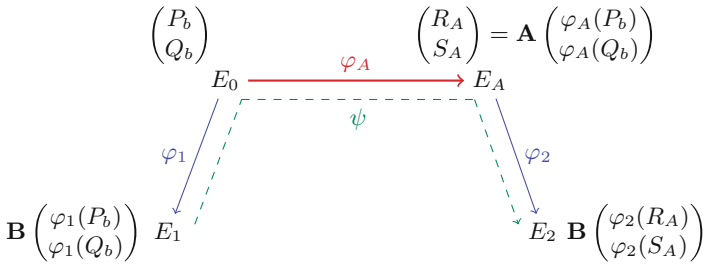


Fig. 1. The FESTA trapdoor function. The parameter generation computes the isogeny φ_A, while the trapdoor function evaluation consists of evaluating the isogenies φ_1 and φ_2. The inversion algorithm recovers the isogeny $\psi = \varphi_2 \circ \varphi_A \circ \widehat{\varphi}_1$.

posed in [26], the two points are scaled by different values, which provides higher security and allows us to use significantly smaller parameters. The decryption algorithm in FESTA recovers two secret isogenies at once by adapting techniques used for SIDH attacks: this is one of the first protocols to use this cryptanalytic tool constructively. A different application of similar techniques has been proposed in [19,24].

Contributions. In this work, we make the following contributions:

1. We propose the FESTA trapdoor function, which constructively uses the SIDH attacks to invert a one-way function.
2. We assess the security of the proposed trapdoor functions. The security proofs rely on novel security assumptions, for which we provide a comprehensive discussion on potential classical and quantum attacks.
3. Relying on the new trapdoors, we apply the OAEP transform [25] to obtain an efficient PKE that is IND-CCA secure in the QROM. We call this the FESTA PKE, or just FESTA. We also derive the first isogeny-based PKE to be IND-CCA secure in the standard model, using a generic transform by Hohenberger, Koppula, and Waters [30].
4. We describe a novel technique to find parameters that lead to a fast computation of the SIDH attacks. In particular, we leverage scalar endomorphisms to obtain an efficient SIDH attack in dimension two that recovers isogenies between supersingular elliptic curves whose endomorphism ring is unknown.
5. Lastly, we implement the proposed FESTA PKE in SageMath: while this is a proof of concept, it demonstrates the feasibility of our protocol. Given these preliminary results, we expect that an optimised implementation of FESTA can offer practical running times that are competitive with existing isogeny-based constructions.

Organisation. In Sect. 2, we cover the necessary background of cryptographic one-way functions and isogenies between abelian varieties. In Sect. 3, we introduce the FESTA family of trapdoor functions, and its security is analysed in

Sect. 4. Then, we build upon the proposed trapdoor functions to obtain a PKE that is IND-CCA secure in the QROM model in Sect. 5. Section 6 gives a precise and concrete description of the FESTA PKE, which is supported by the proof-of-concept implementation detailed in Sect. 7.

Notation. Throughout this paper, we denote the security parameter as λ, and we say a function $f(x)$ is negligible if, for all positive integers c, there exists an integer N such that $|f(x)| < x^{-c}$, for all $x > N$. We write $\mathsf{negl}(\cdot)$ to say a negligible function and $\mathbb{Z}_{>0}$ to represent the set of positive integers. Given a $t \in \mathbb{Z}_{>0}$, we denote its square-free part by t_{sf}. We also write $x \xleftarrow{\$} \mathcal{X}$ to denote that x is sampled uniformly at random among the elements of $\mathcal{X}$.

We also define TorGen to be a *deterministic* algorithm that, given a supersingular elliptic curve E and an integer n, outputs two generators of the n-torsion on E, denoted by $E[n]$. Given four isogenies $\varphi_{i,j} \colon E_i \to E_j$ and two points $P_i \in E_i$, for $i = 1, 2$ and $j = 3, 4$, evaluating the isogeny

$$\begin{pmatrix} \varphi_{1,3} & \varphi_{2,3} \\ \varphi_{1,4} & \varphi_{2,4} \end{pmatrix} : E_1 \times E_2 \to E_3 \times E_4$$

at $\begin{pmatrix} P_1 & P_2 \end{pmatrix}^T$ amounts to

$$\begin{pmatrix} \varphi_{1,3} & \varphi_{2,3} \\ \varphi_{1,4} & \varphi_{2,4} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} \varphi_{1,3}(P_1) + \varphi_{2,3}(P_2) \\ \varphi_{1,4}(P_1) + \varphi_{2,4}(P_2) \end{pmatrix}.$$

In particular, we can view the action of scaling points P_1, Q_1 by a matrix $\mathbf{A}$ just as above, by interpreting the matrix coefficients $\alpha, \beta, \gamma, \delta$ to be scalar endomorphisms:

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} [\alpha]P_1 + [\beta]P_2 \\ [\gamma]P_1 + [\delta]P_2 \end{pmatrix}.$$

2 Preliminaries

In this section, we summarise some background knowledge about public-key encryption schemes and isogenies.

2.1 Cryptographic Preliminaries

For the sake of being self-contained, we briefly recall some cryptographic notions we will use in the rest of the paper; we refer to [7] for background material. The main ingredient in FESTA is the notion of a *trapdoor function*. Roughly speaking, trapdoor functions can be seen as one-way functions with the property of being easily invertible if one has access to additional secret information. While definitions in the literature vary, throughout this paper we restrict ourselves to injective trapdoor functions. Formally, trapdoor functions form a family of functions indexed by the public parameters, but when the context allows it, we may refer to a trapdoor function to denote the entire family for ease of notation.

Definition 1 (Family of trapdoor functions). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets. A* family of trapdoor functions *is a triple of algorithms* $(\mathsf{KeyGen}, f, f^{-1})$ *such that:*

- $\mathsf{KeyGen}(\lambda) \xrightarrow{\$} (\mathsf{sk}, \mathsf{pk})$: KeyGen *is a probabilistic key generation algorithm that outputs a secret key* sk *and a public key* pk *for a given security parameter* λ*;*
- $f(\mathsf{pk}, x) \to y$*: f is a deterministic algorithm that, on input a public key* pk *and $x \in \mathcal{X}$, outputs $y \in \mathcal{Y}$;*
- $f^{-1}(\mathsf{sk}, y) \to x$*: f^{-1} is a deterministic algorithm that, on input the secret key* sk *and $y \in \mathcal{Y}$, outputs $x \in \mathcal{X}$.*

Trapdoor functions need to be correct*, i.e. for all possible outputs* $(\mathsf{pk}, \mathsf{sk})$ *of* KeyGen *and for all $x \in \mathcal{X}$, $f^{-1}(\mathsf{sk}, f(\mathsf{pk}, x)) = x$. Moreover, they should also be* one-way*, which means that given a valid output $y \in \mathcal{Y}$, computed using* $(\mathsf{sk}, \mathsf{pk})$*, and the public key* pk*, any probabilistic polynomial-time adversary cannot compute $x \in \mathcal{X}$ such that $f(\mathsf{pk}, x) = y$ with probability greater than* $\mathsf{negl}(\lambda)$*.*

In this work, we rely on a stronger version of trapdoor functions: *partial-domain trapdoor function* [25]. Informally, this means that not only recovering the entire input is hard, but also the same holds if one tries to recover a part of it.

Definition 2 (Quantum partial-domain one-way function). *Let $\mathcal{X}_0, \mathcal{X}_1$ and $\mathcal{Y}$ be three finite sets. A function $f : \mathcal{X}_0 \times \mathcal{X}_1 \to \mathcal{Y}$ is a* quantum partial-domain one-way function *if, for any polynomial-time quantum adversary $\mathcal{A}$, the following holds:*

$$P\left(s' = s \;\middle|\; s \xleftarrow{\$} \mathcal{X}_0, t \xleftarrow{\$} \mathcal{X}_1, s' \leftarrow \mathcal{A}(f(s,t))\right) < \mathsf{negl}(\lambda).$$

In Sect. 5.1, we show how to derive a Public-key Encryption (PKE) scheme from a quantum partial-domain one-way function.

Definition 3 (Public-key Encryption). *A* public-key encryption scheme *is a triple of efficient algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

- $\mathsf{KeyGen}(\lambda) \xrightarrow{\$} (\mathsf{sk}, \mathsf{pk})$*:* KeyGen *is a probabilistic key generation algorithm that outputs a secret key* sk *and a public key* pk *for a given security parameter* λ*;*
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$*:* Enc *is a probabilistic algorithm that, given a public key* pk *and a message m, returns a ciphertext* ct*;*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m$*:* Dec *is a deterministic algorithm that returns a message m having a ciphertext* ct *and a secret key* sk *as input.*

A public-key encryption scheme is correct if, for all possible outputs $(\mathsf{sk}, \mathsf{pk})$ *of* KeyGen *and for all messages m,* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$*.*

Mainly, there are two notions of indistinguishability security for PKEs: security against a chosen plaintext attack (CPA), and security against a chosen ciphertext attack (CCA). The FESTA PKE we will introduce in Sect. 5 verifies the strongest notion of IND-CCA: roughly speaking, given two messages, any probabilistic polynomial-time adversary cannot distinguish which message has been encrypted even if they can ask to decrypt some ciphertexts different from the challenge ciphertext at any point during the attack.

2.2 Isogenies

Most of the existing isogeny-based cryptosystems rely on the computation of isogenies between elliptic curves. For details on these, we refer the reader to [20,49]. We recall here some results about isogenies between abelian varieties, while keeping in mind our main application: recovering isogenies between elliptic curves from isogenies between abelian varieties.

Implicitly, elliptic curves come equipped with an additional structure: the *principal polarisation*. Principal polarisations can be seen as special isomorphisms between the curve itself and its Jacobian. Thus, the correct generalisation of elliptic curves to higher dimension is *principally polarised abelian varieties*; that is, abelian varieties endowed with a principal polarisation. An abelian variety of dimension two is called *abelian surface*. Up to isomorphisms over the algebraic closure of their field of definition, principally polarised abelian surfaces come into two flavors: either Jacobians of genus-2 hyperelliptic curves or products of two elliptic curves. This property allows us to compute certain polarised isogenies between abelian surfaces efficiently; polarised isogenies are isogenies that are compatible with the principal polarisations on the two abelian surfaces. We refer to [39] for more thorough background and to [36, Section 2] for an introduction to principally polarised abelian surfaces from a cryptographic perspective.

Products of supersingular elliptic curves are the main ingredient underlying the recent attacks on SIDH. Given the description of some torsion under a secret isogeny, it is possible to recover such an isogeny using the following result, which is proved in [36, Theorem 1] and is a corollary of Kani's criterion [32].

Theorem 4. *Let E_0, E_1 and E_2 be three elliptic curves defined over $\overline{\mathbb{F}}_p$ such that there exist two isogenies $\varphi_{N_1}\colon E_0 \to E_1$ and $\varphi_{N_2}\colon E_0 \to E_2$ of coprime degrees $\deg(\varphi_{N_1}) = N_1$ and $\deg(\varphi_{N_2}) = N_2$. Then, the subgroup*

$$\langle([N_2]\varphi_{N_1}(P), [N_1]\varphi_{N_2}(P)), ([N_2]\varphi_{N_1}(Q), [N_1]\varphi_{N_2}(Q))\rangle \subset E_1 \times E_2,$$

where $\langle P, Q\rangle = E_0[N_1 + N_2]$, is the kernel of a $(N_1 + N_2, N_1 + N_2)$-polarised isogeny Φ having product codomain. Furthermore, the matrix form of Φ is given by

$$\begin{pmatrix} \widehat{\varphi}_{N_1} & -\widehat{\varphi}_{N_2} \\ g_{N_2} & \widehat{g}_{N_1} \end{pmatrix},$$

where g_{N_i} are N_i-isogenies such that $\varphi_{N_2} \circ \widehat{\varphi}_{N_1} = g_{N_1} \circ g_{N_2}$.

In the context of the SIDH attacks, Kani's criterion is used to learn information about either Alice or Bob's secret isogeny. In [9], whether the (N_1+N_2, N_1+N_2)-isogeny splits into the product of supersingular elliptic curves is used as an oracle to determine if a guess of a step along the secret isogeny path was correct. In [36], the entire secret isogeny is recovered from Kani's criterion by noticing that up to isomorphism, the dual of the secret isogeny $-\widehat{\varphi}_{N_2}$ can be recovered from one element of the matrix representation of the

$(N_1 + N_2, N_1 + N_2)$-isogeny. In [48], this strategy is generalised to higher dimension to allow provable polynomial-time attacks in the general case.

In the high-level description of FESTA, we write TorAtk to denote a generic attack that can be implemented with different techniques. In other words, given the points $P' = \psi(P)$ and $Q' = \psi(Q)$, for some unknown d-isogeny $\psi\colon E \to E'$, points P, Q such that $E[2^b] = \langle P, Q\rangle$ and $b \in \mathbb{Z}_{>0}$, $\mathsf{TorAtk}(E, P, Q, E', P', Q', d)$ outputs a description of the isogeny $\psi\colon E \to E'$. The concrete description of the attack used for our tailored parameter set is introduced in Sect. 6.

As in the case of elliptic curves, isogenies between principally polarised abelian surfaces can be computed as a chain of (ℓ, ℓ)-isogenies, where ℓ is prime. There exist some algorithms to compute (ℓ, ℓ)-isogenies, where ℓ is an odd prime (see, for instance, [16]). Some recent work has improved existing algorithms for the case $\ell = 3$ [23]. However, for $\ell = 2$, a classical result of Richelot provides an efficient algorithm to compute $(2, 2)$-polarised isogenies between Jacobians of genus-2 hyperelliptic curves [47,50]. For this reason, we will restrict ourselves to N_i-isogenies between elliptic curves such that $N_1 + N_2 = 2^b$, for some $b \in \mathbb{Z}_{>0}$, when searching for parameter sets. This choice allows to implement our protocol only with $(2, 2)$-isogenies in dimension two.

3 The FESTA Trapdoor Function

In this section, we introduce FESTA: a family of quantum-resistant trapdoor functions. The function evaluation consists of computing two isogenies starting from two curves, linked by a secret isogeny: the outputs of the function are then the image curves, together with some scaled torsion images. Roughly speaking, the one-wayness depends on scaling the torsion points, which makes the SIDH attacks unapplicable. The secret trapdoor information is a matrix that undoes the scaling action on the torsion points, which enables the inverter to apply the SIDH attacks and extract the input.

More formally, let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, and fix a basis $\langle P_b, Q_b\rangle = E_0[2^b]$. These values, together with the isogeny degrees, form the parameters common to each function in the trapdoor family. The public key of each trapdoor function is generated by computing a secret d_A-isogeny from E_0 to E_A and consist of the curve E_A, together with the torsion images of P_b, Q_b, scaled by a matrix $\mathbf{A}$ of special form. We write $\mathcal{M}_b$ to denote the set of possible matrices $\mathbf{A}$, and we postpone a precise definition of it until after we introduce the trapdoor inversion procedure.

The public keys are defined by the following set:

$$\mathcal{A}^{\mathsf{pk}} := \left\{ (E_A, R_A, S_A) \;\middle|\; \begin{array}{l} \varphi_A\colon E_0 \to E_A,\ \deg(\varphi_A) = d_A, \\ \mathbf{A} \in \mathcal{M}_b,\ \begin{pmatrix} R_A \\ S_A \end{pmatrix} = \mathbf{A} \begin{pmatrix} \varphi_A(P_b) \\ \varphi_A(Q_b) \end{pmatrix} \end{array} \right\}.$$

For each $(E_A, R_A, S_A) \in \mathcal{A}^{\mathsf{pk}}$, we highlight the dependence of the trapdoor function f on the public key (E_A, R_A, S_A) by using the notation $f_{(E_A, R_A, S_A)}$.

Evaluating the trapdoor function $f_{(E_A,R_A,S_A)}$ consists of computing the d_1-isogeny $\varphi_1 : E_0 \to E_1$ and the d_2-isogeny $\varphi_2 : E_A \to E_2$. The output of the function is the curves E_1, E_2, together with the torsion images of P_b, Q_b under φ_1 and the images of R_A, S_A under φ_2, both scaled by the matrix $\mathbf{B} \in \mathcal{M}_b$. These computations are summarised in Algorithm 1, and we denote its output by $(E_1, R_1, S_1, E_2, R_2, S_2)$.

Algorithm 1. $f_{(E_A,R_A,S_A)}(\langle K_1\rangle, \langle K_2\rangle, \mathbf{B})$

Input: Two cyclic subgroups $\langle K_1\rangle \subset E_0[d_1]$ and $\langle K_2\rangle \subset E_A[d_2]$ of order d_1 and d_2, respectively, and $\mathbf{B} \in \mathcal{M}_b$.

Output: $(E_1, R_1, S_1, E_2, R_2, S_2)$.

1: Compute the d_1-isogeny $\varphi_1 \colon E_0 \to E_1$ having kernel $\langle K_1\rangle$.

2: Compute the d_2-isogeny $\varphi_2 \colon E_A \to E_2$ having kernel $\langle K_2\rangle$.

3: Acting with scalar multiplication compute

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \varphi_1(P_b) \\ \varphi_1(Q_b) \end{pmatrix} \qquad \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \varphi_2(R_A) \\ \varphi_2(S_A) \end{pmatrix}.$$

4: **return** $(E_1, R_1, S_1, E_2, R_2, S_2)$.

To invert the function, we would like to scale the torsion points R_2, S_2 on E_2 to undo the scaling-by-$\mathbf{A}$ transform that was applied during the public-key generation. However, the public points on E_2 have already been scaled by $\mathbf{B}$; we thus need that $\mathbf{A}$ and $\mathbf{B}$ commute. In practice, we require that the matrices are diagonal:[1] applying the matrices then becomes scaling the two torsion points by two independent scalars.

Given diagonal matrices $\mathbf{A}, \mathbf{B}$, we can recover the images of the points R_1, S_1 on E_1 under the composition isogeny $\psi := \varphi_2 \circ \varphi_A \circ \widehat{\varphi}_1$. Note that

$$d_1 \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = \mathbf{B} \cdot \mathbf{A} \cdot \mathbf{B}^{-1} \begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix} = \mathbf{A} \begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix}.$$

Hence, after scaling $d_1 \begin{pmatrix} R_2 & S_2 \end{pmatrix}^T$ by $\mathbf{A}^{-1}$, we can apply the torsion-point attacks on E_1 and E_2 to recover the isogeny ψ, from which we can reconstruct the kernels of φ_1 and φ_2, denoted $\langle K_1\rangle$, $\langle K_2\rangle$ respectively, along with the scaling matrix $\mathbf{B}$. In other words, we have that $\mathsf{TorAtk}(E_1, R_1, S_1, E_2, R'_2, S'_2, d_1 d_A d_2) = \psi$, where the points R'_2, S'_2 are computed by scaling the points $[d_1]R_2, [d_1]S_2$ by the matrix $\mathbf{A}^{-1}$. The procedure to invert $f_{(E_A,R_A,S_A)}$ is summarised in Algorithm 2. Note that our trapdoor can be inverted using any torsion-point attack that works with a starting curve of unknown endomorphism ring. We detail the specifics of the attack algorithm we use in Sect. 6.

[1] This is not the only option: for instance, circulant matrices, i.e. those of the form $\left[\begin{smallmatrix} a & b \\ b & a \end{smallmatrix}\right]$, form a commutative algebra. However, using such matrices does not seem to have any major advantage over diagonal matrices.

The torsion-point attacks can only recovery isogenies up to automorphisms, and, in our setting, the automorphism groups of the curves E_1 and E_2 coincides with $\langle -\mathsf{id} \rangle$.[2] Hence, we define $\mathcal{M}_b$, the set from which the matrices $\mathbf{A}$ and $\mathbf{B}$ are sampled, to be the commutative subset of invertible diagonal matrices over $(\mathbb{Z}/2^b\mathbb{Z})^\times$ modulo $\langle -\mathbf{I}_2 \rangle$, where $\mathbf{I}_2$ represents the 2×2 identity matrix. The modulo $\langle -I_2 \rangle$ condition translates to picking a canonical choice in each equivalence class. For instance, the canonical representative $\mathbf{A}$ of an equivalence class can be the matrix $\mathbf{A}$ that verifies $\mathbf{A}_{1,1} < -\mathbf{A}_{1,1}$, where the comparison is over the integers. Throughout this paper, we always implicitly fix a canonical representative in any equivalence class; as such, we identify the equivalent classes in $\mathcal{M}_b$ with their canonical representatives. We can extend the definition to the more general case by $\mathcal{M}_n$ to denote the commutative subset of invertible diagonal matrices over $(\mathbb{Z}/n\mathbb{Z})^\times$ modulo $\langle -\mathbf{I}_2 \rangle$, for any smooth integer n.

Algorithm 2. $f^{-1}_{(E_A,R_A,S_A)}(E_1, R_1, S_1, E_2, R_2, S_2)$

Input: A tuple $(E_1, R_1, S_1, E_2, R_2, S_2)$, the trapdoor $(\mathbf{A} \in \mathcal{M}_b, \varphi_A \colon E_0 \to E_A)$.
Output: $(\langle K_1 \rangle, \langle K_2 \rangle, \mathbf{B})$ such that $f_{(E_A,R_A,S_A)}(\langle K_1 \rangle, \langle K_2 \rangle, \mathbf{B}) = (E_1, R_1, S_1, E_2, R_2, S_2)$.

1: Recover R'_2, S'_2 by inverting $\mathbf{A}$ and acting with scalar multiplication:

$$\begin{pmatrix} R'_2 \\ S'_2 \end{pmatrix} = d_1 \mathbf{A}^{-1} \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}.$$

2: Compute $\psi = \varphi_2 \circ \varphi_A \circ \widehat{\varphi}_1 \colon E_1 \to E_2$ via $\mathsf{TorAtk}(E_1, R_1, S_1, E_2, R'_2, S'_2, d_1 d_A d_2)$.
3: Recover the kernel $\langle K_1 \rangle$ of the d_1-isogeny $\varphi_1 \colon E_0 \to E_1$ from ψ using φ_A.
4: Recover the kernel $\langle K_2 \rangle$ of the d_2-isogeny $\varphi_2 \colon E_A \to E_2$ from ψ using φ_A.
5: Compute $\mathbf{B} \in \mathcal{M}_b$ such that

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \varphi_1(P_b) \\ \varphi_1(Q_b) \end{pmatrix}.$$

6: **return** $(\langle K_1 \rangle, \langle K_2 \rangle, \mathbf{B})$.

Remark 5. As in SIDH, upon choosing a canonical basis $\langle P, Q \rangle$ of $E[d]$, we can restrict ourselves to isogenies whose kernels are cyclic subgroups of the form $\langle P + [x]Q \rangle$, without affecting the security of the protocol. This makes it possible to represent every isogeny with an element in $\mathbb{Z}/d\mathbb{Z}$. This representation is injective if the automorphisms on the curve E are only $\pm\mathsf{id}$: to avoid such issues, we choose the starting curve E_0 to have j-invariant $\neq 0, 1728$.

Hence, the domain of $f_{(E_A,R_A,S_A)}$ is $\mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b$. Additionally, we denote its codomain by $\mathcal{S}$.

The trapdoor function we are proposing is correct, i.e. the inversion algorithm produces the original function input. The isogeny ψ is uniquely determined by

[2] Unless either $j(E_1)$ or $j(E_2) \in \{0, 1728\}$, which happens with negligible probability.

its action on the 2^b torsion [37, Section 4]; in other words, there is only one isogeny of degree $d_1 d_A d_2$ that maps R_1 and S_1 to R'_2 and S'_2. Hence, the function TorAtk recovers the unique isogeny up to automorphisms. If all the automorphism groups of the curves E involved in the protocol are trivial (i.e. $\mathrm{Aut}(E) = \langle -\mathsf{id} \rangle$), which is the case for all curves with j-invariant $\notin \{0, 1728\}$, the kernels are uniquely defined and the images of torsion points are defined up to inversions. This is because the matrix $\mathbf{B}$ is a canonical representative of the equivalence class modulo $\langle -\mathbf{I}_2 \rangle$. Additionally, the matrix $\mathbf{B}$ is invertible, and thus the torsion-point scaling is also an injection. Hence, the inversion algorithm produces the correct output with overwhelming probability, which also implies that the function is injective.

4 Security of the FESTA Trapdoor

In this section, we analyse the security of the FESTA trapdoor. We first introduce a computational and a decisional variant of the problem that asks to either compute an isogeny or distinguish whether an isogeny exists between two curves, given the image of torsion points scaled by a matrix $\mathbf{A} \in \mathcal{M}_b$.[3] These problems can be seen as a generalisation of the classic isogeny problems [22] to the scaled-torsion setting.

Problem 6 (Decisional isogeny with scaled-torsion (DIST) problem). Let E_0 be a supersingular elliptic curve, and P_0, Q_0 be two points spanning $E_0[n]$, for some smooth order n. Fix a smooth degree d, coprime with n, and given an elliptic curve E_1 and two points P_1, Q_1, sampled with probability 1/2 from either distribution:

- $\mathcal{D}_0 = \{E_1, P_1, Q_1\}$, where E_1 is the codomain of a d-isogeny $\varphi : E_0 \to E_1$, and the points P_1, Q_1 are given by $(P_1\ Q_1)^T = \mathbf{A}(\varphi(P_0)\ \varphi(Q_0))^T$, where the matrix $\mathbf{A} \xleftarrow{\$} \mathcal{M}_n$,
- $\mathcal{D}_1 = \{E_1, P_1, Q_1\}$, where E_1 is a random supersingular elliptic curve with the same order as E_0, and (P_1, Q_1) is a random basis of $E_1[n]$,

distinguish from which distribution the values were sampled.

Problem 7 (Computational isogeny with scaled-torsion (CIST) problem). Let $\varphi : E_0 \to E_1$ be an isogeny of smooth degree d between supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, and let n be a smooth integer coprime with d.

Given the curves E_0 with a basis P_0, Q_0 of $E_0[n]$ and the curve E_1 with a basis $\mathbf{A}(\varphi(P_0)\ \varphi(Q_0))^T$, where $\mathbf{A} \xleftarrow{\$} \mathcal{M}_n$, compute the isogeny φ.

Problem 6 is the decisional variant of Problem 7, and as such it is at least as hard as Problem 7. The converse is also partially true: given an oracle that solves Problem 6 for any degree, it is possible to solve Problem 7 using the search-to-decision reduction for classic isogeny problems [29].

[3] The problem definitions can easily be extended to the case of circulant matrices.

The CIST assumption guarantess the hardness of extracting the trapdoor information from the public parameters of a FESTA trapdoor function. However, the output of the FESTA one-way function produces two pairs of curves and torsion points, scaled by the same matrix. The correlated scaling can potentially make inverting the one-way function easier than solving Problem 7. Thus, to guarantee the one-wayness of the FESTA function, we need to introduce the following problem.

Problem 8 (Computational isogeny with double scaled-torsion (CIST^2) problem). Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, and let E_0' be a random supersingular elliptic curves defined over the same field. Consider two isogenies $\varphi : E_0 \to E_1$ and $\varphi' : E_0' \to E_1'$ of smooth degrees d and d', respectively. Let n be a smooth integer coprime with d and d', and let $\mathbf{A}$ be a matrix sampled as $\mathbf{A} \xleftarrow{\$} \mathcal{M}_n$.

Given the curves E_0, E_1, E_0', E_1', two bases P, Q of $E_0[n]$ and P', Q' of $E_0'[n]$, and the points $\mathbf{A}(\varphi(P)\ \varphi(Q))^T$ and $\mathbf{A}(\varphi'(P')\ \varphi'(Q'))^T$, compute the isogenies φ and φ'.

Since this problem provides additional information (two sets of torsion images, scaled by the same matrix), the hardness of Problem 7 is implied by the CIST^2 assumption, whereas the converse may not be true.

Having introduced the relevant computational assumptions, we can now prove the one-wayness of the FESTA trapdoor function.

Theorem 9. *The function $f_{(E_A,R_A,S_A)} : \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b \to \mathcal{S}$, defined in Algorithm 1, is a one-way function, assuming the hardness of Problem 6 for $d = d_A$ and $n = 2^b$ and of Problem 8 for $d = d_1$, $d' = d_2$ and $n = 2^b$.*

Proof. In the definition of one-wayness (see Definition 1), the attacker $\mathcal{A}$ receives the FESTA public parameters, including the d_A-isogenous curves E_0 and E_A, and the FESTA output comprising of the curves E_0, E_1 and the points P_0, Q_0, P_1, Q_1, computed as in Algorithm 1, and produces the isogenies φ_1, φ_2 and the matrix $\mathbf{B}$.

Through a hybrid argument, we can replace curve E_A, which is the d_A-isogenous to E_0, with a random starting curve. Any attacker that can distinguish between the two cases can be used as a distinguisher for Problem 6. Now, any attacker that can invert the FESTA trapdoor function when the curves E_0 and E_1 are randomly generated can be used to solve Problem 8, since the input and outputs are the same. □

Under the strong assumption that an attacker that can solve Problem 6 can do so for any degree d, Problem 6 and 7 are equivalent, and the hardness of Problem 7 is implied by that of Problem 8. In that case, Theorem 9 can thus be simplified to rely only on the CIST^2 assumption.

Hardness Analysis. We now discuss potential strategies that an attacker may employ to solve the presented problems and introduce several arguments to justify the presumed hardness of the corresponding computational assumptions.

First, let us consider the torsion point attacks [9,36,48]: as argued in [48, Section 6.4], given a d-isogeny φ, it is possible to recover φ if the image of the n-torsion is available, provided $n^2 > 4d$. While FESTA does reveal torsion point images of sufficiently large order, these are scaled by a random invertible diagonal matrix. An attacker may recover the determinant of such a matrix through pairing computations since $e([\alpha]\varphi(P), [\beta]\varphi(Q)) = e(P,Q)^{\alpha\beta \deg \varphi}$ and P, Q and $\deg \varphi$ are known. This information can be used to remove one variable: given $P' = [\alpha]\varphi(P)$, $Q' = [\beta]\varphi(Q)$ and $\alpha\beta$, scaling Q' by $(\alpha\beta)^{-1} \pmod n$ yields the point $Q'' = [1/\alpha]\varphi(Q)$. Thus, P' and Q'' are the images of P, Q scaled by a random diagonal matrix of determinant one, where the scaling depends uniquely on the value α. While this change reduces the number of variables, it does not affect security because α is randomly sampled from an exponentially large set. Due to this reduction, in the rest of the paper we can restrict the matrices to those with unitary determinant without affecting the security of the protocol.

If the attacks on SIDH do not apply to FESTA, it is natural to wonder whether the attacks could be extended to cover the case of scaled torsion points. This seems unlikely, because the torsion information revealed by FESTA is significantly less than that in SIDH, or even the variants of SIDH called M-SIDH and MD-SIDH [26] that are believed to be secure—we compare FESTA to M-SIDH and MD-SIDH at the end of this section.

Another attack strategy consists of guessing (or brute forcing) the scaling value α. While the scaling values are sampled from an exponentially large set, the attacker can also focus on recovering only part of α. Given the scaled points $P' = [\alpha]\varphi(P)$ and $Q' = [\alpha^{-1}]\varphi(Q)$, the attacker can scale them by a power of two and obtain

$$2^{b-j}P' = [\alpha \bmod 2^j]\varphi([2^{b-j}]P), \quad 2^{b-j}Q' = [\alpha^{-1} \bmod 2^j]\varphi([2^{b-j}]Q).$$

This means that it is possible to guess only $\alpha \bmod 2^j$ if the images of points of order 2^j is enough to apply the SIDH attacks on the secret isogeny. However, FESTA uses isogenies of degree $2^{2\lambda}$, which implies this guessing attack requires $j = \lambda$ and has thus a computational cost of 2^λ. Thus, as long as the isogenies have degree at least $2^{2\lambda}$, the best known attack against Problem 6 and 7 is a simple meet-in-the-middle attack that ignores the additional torsion information.

Remark 10. Some isogeny protocols have been known to be vulnerable when the starting curve has known endomorphism ring [4,6], when the known endomorphism ring contains small endomorphisms [26], or when the starting curve (and potentially the underlying prime) is maliciously chosen [45]. In many of these cases, a trusted setup is a necessary countermeasure [2]. This does not appear to be the case in FESTA, where Problem 7 remains hard even when the starting curve E_0 is a special curve with known endomorphism ring, such as the case $j(E_0) = 1728$ or a close neighbour. Nonetheless, any potential future attack that exploits unknown endomorphism ring can be avoided by generating E_0 during key generation and including its description in the public key.

Very recent analysis [11] has shown that it is possible to recover an isogeny given its scaled action, i.e. efficiently solve Problem 7 when the attacker knows

an endomorphism on E_0 (or an endomorphism composed with the Frobenius map from E_0 to its Frobenius conjugate) that acts as scalar multiplication on the starting basis P, Q. However, a random basis, such as that deterministically generated from its curve, is subject to such an attack only with probability negligible in the security parameter. The parameters chosen in the implementation described in Sect. 7 are thus not affected by this attack.

Up until now, we focused on the hardness of the CIST assumption. However, the security of FESTA relies on the CIST^2 assumption, which might be easier to break. This is because the attacker has access to two CIST samples, where the scaling matrix $\mathbf{A}$ is the same. This may be useful, for instance, because an attacker that successfully recovers the isogeny in one of the CIST samples can obtain the correct torsion images in the other sample by scaling by $\mathbf{A}^{-1}$, recovered in the first sample. Applying the SIDH attacks then yields the second isogeny in polynomial time. However, this approach relies on one CIST instance being already broken. More generally, it seems that the correlated scaling matrix does not reveal significantly more information: the correlation between the instances is very tenuous, as the two samples have different starting curves and use isogenies of different large degrees (usually, the two degrees are coprime). Thus, it is unclear how an attacker may exploit the correlation to devise a strategy to break either CIST instance.

If we consider quantum-enabled adversaries, the security profile of FESTA remains mostly unchanged. Similarly to the classical case, it appears to be hard for a quantum attacker to exploit the scaled torsion images. Thus, such an attacker would be limited to attempting to solve the torsionless version of the isogeny problem, i.e. recover the secret isogeny given only the end curves and the isogeny degree. In this setting, we can rely on the quantum security analysis of SIDH [31], which shows that sufficiently long isogenies are hard to recover even with a quantum computer.

Comparison with Existing Protocols. In SIDH, and M-SIDH and MD-SIDH [26], the parties reveal the scaled action $[\alpha]\varphi(P), [\alpha]\varphi(Q)$ of a secret isogeny φ on a torsion basis P, Q (in SIDH, $\alpha = 1$), but crucially the scaling value is the same for both points. This information is sufficient to compute the images of exponentially-many full-order subgroups: for any subgroup $\langle [x]P + [y]Q \rangle$, its image under the secret isogeny is $\langle [x]([\alpha]\varphi(P)) + [y]([\alpha]\varphi(Q)) \rangle$. This is not the case in FESTA: since the torsion images are scaled by different values, only the pushforward of two subgroups of full order is revealed: the pushforward of $\langle P \rangle$ is $\langle [\alpha]\varphi(P) \rangle$ and that of $\langle Q \rangle$ is $\langle [1/\alpha]\varphi(Q) \rangle$. Hence, FESTA reveals significantly less information about its secret isogenies than SIDH, M-SIDH, and MD-SIDH, which makes an extension of the SIDH attacks to FESTA unlikely.

We can also compare FESTA to other isogeny-based protocols. In binSIDH and terSIDH [3], the parties also reveal the images of two torsion points scaled by different values, similarly to what happens in FESTA. Indeed, Problem 7 is very similar to [3, Problem 5 (SSIP-A)]; however, the torsion points in binSIDH and terSIDH have highly composite order. This means that the pushforward of

exponentially many subgroups of full order is still available, although the number is much smaller than if the points were scaled by the same value. Thus, FESTA uses torsion points of prime power order, and thus also reveals less information than binSIDH and terSIDH.

Lastly, if we consider CSIDH (in its many variants), we see that CSIDH also implicitly reveals the images of some subgroups: the image of the subgroup $\ker(\pi - 1) \cap E_0[\ell]$ under a secret isogeny from E_0 to E_1 is $\ker(\pi - 1) \cap E_1[\ell]$, and the image of $\ker(\pi + 1) \cap E_0[\ell]$ is $\ker(\pi + 1) \cap E_1[\ell]$. While this suggests a relationship between the CSIDH assumption and the hardness of Problem 7, the isogenies used in CSIDH are $\mathbb{F}_p$-rational. This may be a small difference, but it makes the two assumptions different enough that they cannot be compared. For instance, consider the attack by Castryck and Vercauteren [11]: while it applies to specially crafted instances of FESTA, the attack cannot be extended to CSIDH.

5 The FESTA Public-Key Encryption Protocol

In this section, we show how the FESTA trapdoor function can be used to build public-key encryption protocols with different security guarantees.

5.1 IND-CCA Encryption in the QROM

Given an injective partial-domain trapdoor function, Ebrahimi [25] showed it is possible to obtain a IND-CCA PKE, secure in the Quantum Random Oracle Model (QROM) by using the OAEP transform.

To use the OAEP transform in our construction, we first need to prove that the FESTA function is indeed a partial-domain trapdoor function.

Theorem 11. *The function $f_{(E_A,R_A,S_A)} : \mathbb{Z}/d_1\mathbb{Z} \times (\mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b) \to \mathcal{S}$ defined in Algorithm 1 is a quantum partial one-way function, under the hardness of Problem 7 and 8.*

Proof. We show a stronger statement, i.e. that recovering any of three inputs is as hard as full-domain inversion: given the isogeny φ_1, the matrix $\mathbf{B}$ can be obtained by computing the change-of-basis matrix between $\varphi_1(P_b), \varphi_1(Q_b)$ and R_1, S_1. The remaining input, the isogeny φ_2, can be computed as the output of $\mathsf{TorAtk}(E_A, R_A, S_A, E_2, R_2^\star, S_2^\star, d_2)$, where the points $R_2^\star, S_2^\star$ are obtained by scaling the points R_2, S_2 by $\mathbf{B}^{-1}$. □

After applying the OAEP transform, we obtain the following PKE protocol: the prime p, the curve E_0, the values d_1, d_2, d_A, b, and a description of the set $\mathcal{M}_b$ form the PKE parameters. We also rely on two random oracles, $G\colon \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b \to \mathbb{Z}/d_1\mathbb{Z}$, and $H\colon \mathbb{Z}/d_1\mathbb{Z} \to \mathbb{Z}/d_2\mathbb{Z} \times \mathcal{M}_b$. The KeyGen algorithm is similar to that in the trapdoor function, and it produces the trapdoor public parameters E_A, R_A, S_A.

To encrypt, we first evaluate G at a randomly sampled input (r, R), and we use its output, combined with the message m, to determine the kernel of the isogeny φ_1. The isogeny φ_2 and the matrix $\mathbf{B}$, which are the remaining part of the input for the trapdoor function, are deterministically derived from the randomness (r, R) and the kernel of φ_1 via H. The output of the trapdoor function determines the ciphertext of the encryption algorithm.

During decryption, the trapdoor information is used to recover the isogenies φ_1, φ_2 and the matrix $\mathbf{B}$, from which the message can be extracted. These procedures are formalised in Algorithms 3 and 4.

Note that, unlike the trapdoor definition used in [25], the input and output spaces of our trapdoor function are not binary strings; thus, the xor operation used in the transform presented in [25] would not produce correct results. We replaced it with ring addition for the values representing kernel generators and matrix multiplications for the scaling matrix, without affecting the security of the transform.

Algorithm 3. FESTA.Enc(pk, m)

Input: The public key $\mathsf{pk} = (E_A, R_A, S_A)$ and the message m to be encrypted.
Output: The ciphertext $(E_1, R_1, S_1, E_2, R_2, S_2)$.

1: Sample $r \xleftarrow{\$} \mathbb{Z}/d_2\mathbb{Z}$ and $R \xleftarrow{\$} \mathcal{M}_b$.
2: Write $m' = m \,||\, 0^k \bmod d_1$ and compute $s = m' + G(r, R)$.
3: Write $(x, X) = H(s)$ and compute $t = x + r$, $T = XR$.
4: Compute $\mathsf{ct} = f_{(E_A, R_A, S_A)}(s, t, T)$. ▷ Using Algorithm 1
5: **return** $\mathsf{ct} = (E_1, R_1, S_1, E_2, R_2, S_2)$.

Algorithm 4. FESTA.Dec(sk, ct)

Input: The secret key $\mathsf{sk} = (\mathbf{A}, \varphi_A)$ and the ciphertext $\mathsf{ct} = (E_1, R_1, S_1, E_2, R_2, S_2)$.
Output: The decrypted message m or $\perp$ on failure.

1: Compute $(s, t, T) = f^{-1}_{(E_A, R_A, S_A)}(\mathsf{sk}, \mathsf{ct})$. ▷ Using Algorithm 2
2: Write $(x, X) = H(s)$ and compute $r = t - x$, $R = X^{-1}T$.
3: Compute $m' = s - G(r, R)$ and write $m \,||\, m_k = m'$, where $|m_k| = k$.
4: **if** $m_k = 0^k$ **then**
5: **return** m.
6: **else**
7: **return** $\perp$.

Remark 12. Most post-quantum encryption protocols attain IND-CCA security using the Fujisaki-Okamoto transform [27], which requires re-evaluating the encryption procedure during decryption. Besides the computational overhead, the re-evaluation check has enabled a wide range of side-channel attacks [52]. These issues are entirely avoided by FESTA: decryption does not require to run

the encryption algorithm, which reduces the latency of the decryption algorithm and brings a significant advantage in the development of side-channel-resistant implementations.

Partial-Input Extraction. The OAEP transform requires that the entire input is computed in Line 1 of Algorithm 4. This is necessary to avoid trivial CCA attacks: in the case of FESTA, if the matrix $\mathbf{B}$ is not recovered, an attacker may scale all torsion points by the same diagonal matrix to obtain a different but valid ciphertext. However, recovering all inputs limits our choice of parameters (as we will show in Sect. 6, extracting both isogenies φ_1 and φ_2 requires d_1, d_2, and d_A to be pairwise coprime) and reduces the efficiency of the inversion algorithm.

In the Random Oracle Model, we can modify the trapdoor function with a technique similar to the Fujisaki-Okamoto transform [28]. The new function f receives as input only the kernel corresponding to φ_1: the isogeny φ_2 and the matrix $\mathbf{B}$ are obtained deterministically from φ_1 through a random oracle. The inversion function also needs to be modified to extract the isogeny φ_1 and $\mathbf{B}$. Then, from the knowledge of $\ker(\varphi_1)$, we check that the kernel of φ_2 is correct to ensure that the output matches what an honest evaluator would have computed. If we only need to recover φ_1 during inversion, we would not require d_2 to be coprime with d_A; this would translate in a prime p that is about λ bits shorter than the prime currently proposed.

The parameters proposed in Sect. 7.3 and the implementation discussed in Sect. 7 consider a full inversion and do not integrate this optimisation. This is to keep FESTA simple and maintain a cleaner security proof. We leave a thorough analysis of the benefits of this optimisation for future work.

5.2 IND-CCA Encryption in the Standard Model

While trapdoor functions from group actions are known in the literature [1], FESTA is currently the only secure trapdoor function based on non-group-action isogenies. Besides enabling efficient encryption, as described in the previous section, this allows us to apply the techniques presented in [30] to obtain a PKE protocol that is IND-CCA secure in the standard model. To the best of our knowledge, this is the first PKE based on non-group-action isogenies to be IND-CCA secure in the standard model.

The construction relies upon two building blocks: a randomness-recoverable IND-CPA PKE and a tagged set commitment protocol. The first can be built from an *almost-all-keys injective trapdoor function*. This requires that for nearly all private/public key pairs, the trapdoor inversion function outputs precisely the same input that the function was evaluated at for *all* inputs. This is generally not the case in FESTA since for a large class of public keys, it may be possible that a specific input produces an output curve with j-invariant in $\{0, 1728\}$. In that case, the function may not be injective because the target curve has additional automorphisms. However, in FESTA, whether the inversion is correct depends

entirely on public information: hence, we can check whether an input may lead to issues by evaluating the trapdoor function and checking the j-invariant of the output. We can thus satisfy the almost-all-keys injective property by redefining the function input to exclude the particular inputs that may be problematic.

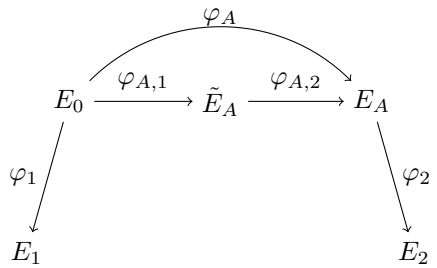
The construction of tagged set commitment protocol requires, besides a trapdoor function, a strongly secure one-time signature. Such a signature can be constructed from any one-way function [33]: we can thus use the FESTA function to construct all the elements needed to obtain an isogeny-based PKE that is IND-CCA secure in the standard model.

6 Concrete Instantiation

In this section, we propose concrete parameter sets for FESTA. Such parameters are specifically tailored to make the recovery of the d_i-isogenies as fast as possible via Theorem 4.

6.1 Recovering an Isogeny from Torsion Point Images

We now describe how to invert the trapdoor functions proposed in Sect. 3. Let $\varphi_1 \colon E_0 \to E_1$, $\varphi_A \colon E_0 \to E_A$ and $\varphi_2 \colon E_A \to E_2$ be three isogenies between supersingular elliptic curves having odd degrees d_1, d_A and d_2, respectively, such that $\gcd(d_1, d_A) = \gcd(d_1, d_2) = \gcd(d_2, d_A) = 1$. The isogeny φ_A is computed as the composition of two isogenies $\varphi_{A,1} \colon E_0 \to \tilde{E}_A$ and $\varphi_{A,2} \colon \tilde{E}_A \to E_A$ of coprime degrees $d_{A,1}$ and $d_{A,2}$, respectively. Graphically, we have



Suppose now we have found $m_1, m_2, b \in \mathbb{Z}_{>0}$ such that

$$m_1^2 d_{A,1} d_1 + m_2^2 d_{A,2} d_2 = 2^b, \tag{1}$$

for some odd m_i coprime to d_1, d_2 and d_A. Specialising Theorem 4 to the case where $\varphi_{N_1} = [m_1] \circ \varphi_1 \circ \widehat{\varphi}_{A,1}$ and $\varphi_{N_2} = [m_2] \circ \varphi_2 \circ \varphi_{A,2}$, the isogeny Φ with kernel

$$\langle ([m_2 d_{A,2} d_2]\varphi_1(P_b), [d_1 m_1]\varphi_2 \circ \varphi_A(P_b)), ([m_2 d_{A,2} d_2]\varphi_1(Q_b), [d_1 m_1]\varphi_2 \circ \varphi_A(Q_b)) \rangle,$$

where the points P_b, Q_b form a basis of $E_0[2^b]$, has matrix form

$$\Phi = \begin{pmatrix} [m_1] \circ \varphi_{A,1} \circ \widehat{\varphi}_1 & -[m_2] \circ \widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2 \\ [m_2] \circ g_{d_2 d_{A,2}} & [m_1] \circ \widehat{g}_{d_{A,1} d_1} \end{pmatrix}.$$

Additionally, we have that $\varphi_2 \circ \varphi_A \circ \widehat{\varphi}_1 = g_{d_{A,1}d_1} \circ g_{d_2 d_{A,2}}$, $\deg(g_{d_{A,1}d_1}) = d_{A,1}d_1$ and $\deg(g_{d_2 d_{A,2}}) = d_2 d_{A,2}$.

Given a security parameter λ, we define

$$\mathsf{params}_\lambda = (m_1, m_2, b, p, d_1, d_{A,1}, d_{A,2}, d_2, E_0)$$

to be a parameter set, where E_0 is a supersingular curve whose j-invariant $\neq 0, 1728$ and $E_0(\mathbb{F}_{p^2}) \simeq \mathbb{Z}/(p+1)\mathbb{Z} \times \mathbb{Z}/(p+1)\mathbb{Z}$, the prime p is of the form $f 2^b d_1 (d_{A,1}d_{A,2})_{\mathsf{sf}} d_2 - 1$ for some small $f > 0$, and $m_1^2 d_{A,1} d_1 + m_2^2 d_{A,2} d_2 = 2^b$.

On input params_λ, we give a precise description of the trapdoor evaluation algorithm (Algorithm 1) in Algorithm 5. The kernels, which are cyclic subgroups $\langle K_1 \rangle \subset E_0[d_1]$ and $\langle K_2 \rangle \subset E_A[d_2]$, are chosen such that they are generated by an element of the form $P + [x]Q$, for some basis (P, Q): thus, they can respectively be represented by $s_1 \in [0, d_1 - 1]$ and $s_2 \in [0, d_2 - 1]$, given two bases (P_{d_1}, Q_{d_1}) of $E_0[d_1]$ and $(P^A_{d_2}, Q^A_{d_2})$ of $E_A[d_2]$.

Since d_1 and d_2 both divide $p+1$, the d_1-torsion of E_0 and the d_2-torsion of E_A are defined over $\mathbb{F}_{p^2}$. This choice allows us to compute d_i-isogenies using points that are $\mathbb{F}_{p^2}$-rational, making the entire computation faster. As for the d_A-isogeny, only $(d_A)_{\mathsf{sf}}$ is included as a factor in $p+1$ as we need not represent the kernel of φ_A explicitly.

Algorithm 5. $f_{(E_A, R_A, S_A)}(s_1, s_2, \mathbf{B})$

Input: Two integers $s_1 \in [0, d_1 - 1]$ and $s_2 \in [0, d_2 - 1]$, and $\mathbf{B} \in \mathcal{M}_b$.
Output: $(E_1, R_1, S_1, E_2, R_2, S_2)$.

1: Compute the bases $(P_{d_1}, Q_{d_1}) \leftarrow \mathsf{TorGen}(E_0, d_1)$ and $(P^A_{d_2}, Q^A_{d_2}) \leftarrow \mathsf{TorGen}(E_A, d_2)$.
2: Compute the d_1-isogeny $\varphi_1 \colon E_0 \to E_1$ having kernel $\langle P_{d_1} + [s_1]Q_{d_1} \rangle$.
3: Compute the d_2-isogeny $\varphi_2 \colon E_A \to E_2$ having kernel $\langle P^A_{d_2} + [s_2]Q^A_{d_2} \rangle$.
4: Acting with scalar multiplication compute

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \varphi_1(P_b) \\ \varphi_1(Q_b) \end{pmatrix} \qquad \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \varphi_2(R_A) \\ \varphi_2(S_A) \end{pmatrix}.$$

5: **return** $(E_1, R_1, S_1, E_2, R_2, S_2)$.

To invert $f_{(E_A, R_A, S_A)}$, given the tuple $(E_1, R_1, S_1, E_2, R_2, S_2)$, we compute the points

$$\begin{pmatrix} R'_2 \\ S'_2 \end{pmatrix} = \mathbf{A}^{-1} \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}.$$

Thus, as we explained above, the isogeny Φ with kernel

$$\langle ([m_2 d_{A,2} d_2]R_1, [d_1 m_1]R'_2), ([m_2 d_{A,2} d_2]S_1, [d_1 m_1]S'_2) \rangle,$$

has matrix form

$$\Phi = \begin{pmatrix} [m_1] \circ \varphi_{A,1} \circ \widehat{\varphi}_1 & -[m_2] \circ \widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2 \\ [m_2] \circ g_{d_2 d_{A,2}} & [m_1] \circ \widehat{g}_{d_{A,1}d_1} \end{pmatrix}.$$

If F is the image curve of $g_{d_2 d_{A,2}} \colon E_1 \to F$, then Φ maps $E_1 \times E_2$ onto $\tilde{E}_A \times F$, up to polarised isomorphisms.

Let $(P^1_{d_1}, Q^1_{d_1}) \leftarrow \mathsf{TorGen}(E_1, d_1)$ and $(P^2_{d_2}, Q^2_{d_2}) \leftarrow \mathsf{TorGen}(E_2, d_2)$. Then, we have

$$\begin{pmatrix} L \\ - \end{pmatrix} := \Phi \begin{pmatrix} P^1_{d_1} + R_1 \\ P^2_{d_2} \end{pmatrix} = \begin{pmatrix} [m_1]\varphi_{A,1} \circ \widehat{\varphi}_1(P^1_{d_1} + R_1) - [m_2]\widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2(P^2_{d_2}) \\ \hline \end{pmatrix},$$

from which we can compute

$$\begin{aligned} [2^b d_2 m_1]\varphi_{A,1} \circ \widehat{\varphi}_1(P^1_{d_1}) &= [2^b d_2]L, \\ [2^b d_1 m_2]\widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2(P^2_{d_2}) &= -[2^b d_1]L, \\ [d_1 d_2 m_1]\varphi_{A,1} \circ \widehat{\varphi}_1(R_1) &= [d_1 d_2]L. \end{aligned}$$

Similarly, we evaluate $\Phi \begin{pmatrix} Q^1_{d_1} + S_1 & Q^2_{d_2} \end{pmatrix}^T$ to obtain $[2^b d_2 m_1]\varphi_{A,1} \circ \widehat{\varphi}_1(Q^1_{d_1})$, $[2^b d_1 m_2]\widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2(Q^2_{d_2})$ and $[d_1 d_2 m_1]\varphi_{A,1} \circ \widehat{\varphi}_1(S_1)$.

Using the knowledge of $\varphi_{A,i}$, we can extract the images $\widehat{\varphi}_1(P^1_{d_1})$, $\widehat{\varphi}_1(Q^1_{d_1})$, $\widehat{\varphi}_2(P^2_{d_2})$ and $\widehat{\varphi}_2(Q^2_{d_2})$. With these, we compute s_1 and s_2 such that $\ker(\varphi_1) = \langle P_{d_1} + [s_1]Q_{d_1} \rangle$ and $\ker(\varphi_2) = \langle P^A_{d_2} + [s_2]Q^A_{d_2} \rangle$. This is slightly more complicated than the more common case when d_1 is a prime power and can be done following Algorithm 6.

Algorithm 6. $\mathsf{ComputeCanonicalKernel}(\widehat{\varphi}(P'), \widehat{\varphi}(Q'), d)$

Input: $\widehat{\varphi}(P')$ and $\widehat{\varphi}(Q')$, where $\varphi \colon E \to E'$ is a d-isogeny and $\langle P', Q' \rangle = E'[d]$.
Output: $s \in [0, d-1]$ such that $\ker(\varphi) = \langle P + [s]Q \rangle$, where $(P, Q) \leftarrow \mathsf{TorGen}(E, d)$.[4]

Compute the canonical basis $(P, Q) \leftarrow \mathsf{TorGen}(E, d)$, and let $d = \prod_{i=1}^n \ell_i^{e_i}$.
Compute $a_1, b_1 \in [0, d-1]$ such that $\widehat{\varphi}(P') = [a_1]P + [b_1]Q$.
Compute $a_2, b_2 \in [0, d-1]$ such that $\widehat{\varphi}(Q') = [a_2]P + [b_2]Q$.
for $i = 1, \ldots, n$ **do**
    if $a_1 = 0 \pmod{\ell_i}$ **then**
        Impose $t_1 = 0 \pmod{\ell_i^{e_i}}$ and $t_2 = a_2^{-1} \pmod{\ell_i^{e_i}}$.
    else
        Impose $t_1 = a_1^{-1} \pmod{\ell_i^{e_i}}$ and $t_2 = 0 \pmod{\ell_i^{e_i}}$.
Lift t_1 and t_2 in $\mathbb{Z}/d\mathbb{Z}$, and define $s \leftarrow t_1 b_1 + t_2 b_2$.
return s.

Finally, we highlight that

$$\begin{pmatrix} \widehat{\varphi}_1(R_1) \\ \widehat{\varphi}_1(S_1) \end{pmatrix} = d_1 \mathbf{B} \begin{pmatrix} P_b \\ Q_b \end{pmatrix},$$

which implies we can recover the matrix $\mathbf{B}$ by solving a discrete logarithm problem, which is efficient as the order of our points is a power of two. To ensure

[4] We highlight that we already know that $\ker(\varphi)$ can be expressed as $\langle P + [s]Q \rangle$, for some $s \in \mathbb{Z}/d\mathbb{Z}$.

that the inversion input was computed as the correct output of the FESTA trapdoor function, we check that $\mathbf{B} \in \mathcal{M}_b$: if not, the inversion algorithm fails and returns $\perp$. We summarise the inversion procedure in Algorithm 7.

Algorithm 7. $f^{-1}_{(E_A,R_A,S_A)}(E_1, R_1, S_1, E_2, R_2, S_2)$

Input: TDF output $y := (E_1, R_1, S_1, E_2, R_2, S_2)$, and $\mathsf{sk} = (\mathbf{A}, \varphi_{A,1}, \varphi_{A,2})$.
Output: The TDF input $x := (s_1, s_2, \mathbf{B})$ such that $f_{(E_A,R_A,S_A)}(x) = y$.

1: Compute $\begin{pmatrix} R'_2 \\ S'_2 \end{pmatrix} = \mathbf{A}^{-1} \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}$.
2: Define Φ to be the isogeny with kernel

$$\langle([m_2 d_{A,2} d_2]R_1, [m_1 d_1]R'_2), ([m_2 d_{A,2} d_2]S_1, [m_1 d_1]S'_2)\rangle.$$

3: **if** The codomain of Φ does not split **then return** $\perp$.
4: Set $(P^1_{d_1}, Q^1_{d_1}) \leftarrow \mathsf{TorGen}(E_1, d_1)$ and $(P^2_{d_2}, Q^2_{d_2}) \leftarrow \mathsf{TorGen}(E_2, d_2)$.
5: Evaluate

$$\begin{pmatrix} L_1 \\ - \end{pmatrix} = \Phi \begin{pmatrix} P^1_{d_1} + R_1 \\ P^2_{d_2} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} L_2 \\ - \end{pmatrix} = \Phi \begin{pmatrix} Q^1_{d_1} + S_1 \\ Q^2_{d_2} \end{pmatrix}.$$

6: Unpack L_1 to obtain $\varphi_{A,1} \circ \widehat{\varphi}_1(P^1_{d_1})$, $\varphi_{A,1} \circ \widehat{\varphi}_1(R_1)$ and $\widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2(P^2_{d_2})$.
7: Unpack L_2 to obtain $\varphi_{A,1} \circ \widehat{\varphi}_1(Q^1_{d_1})$, $\varphi_{A,1} \circ \widehat{\varphi}_1(S_1)$ and $\widehat{\varphi}_{A,2} \circ \widehat{\varphi}_2(Q^2_{d_2})$.
8: Set $s_1 \leftarrow \mathsf{ComputeCanonicalKernel}(\widehat{\varphi}_1(P^1_{d_1}), \widehat{\varphi}_1(Q^1_{d_1}), d_1)$. ▷ Via Algorithm 6
9: Set $s_2 \leftarrow \mathsf{ComputeCanonicalKernel}(\widehat{\varphi}_2(P^2_{d_2}), \widehat{\varphi}_2(Q^2_{d_2}), d_2)$. ▷ Via Algorithm 6
10: Compute $\mathbf{B} \in \mathcal{M}_b$ such that

$$\begin{pmatrix} \widehat{\varphi}_1(R_1) \\ \widehat{\varphi}_1(S_1) \end{pmatrix} = d_1 \mathbf{B} \begin{pmatrix} P_b \\ Q_b \end{pmatrix}.$$

11: **if** $\mathbf{B} \notin \mathcal{M}_b$ **then**
12: **return** $\perp$.
13: **else**
14: **return** $(s_1, s_2, \mathbf{B})$.

Remark 13. To compute the images

$$\widehat{\varphi}_1(P^1_{d_1}) = [d_{A,1}]^{-1} \widehat{\varphi}_{A,1}\left(\varphi_{A,1} \circ \widehat{\varphi}_1(P^1_{d_1})\right), \quad \text{and}$$
$$\widehat{\varphi}_1(Q^1_{d_1}) = [d_{A,1}]^{-1} \widehat{\varphi}_{A,1}\left(\varphi_{A,1} \circ \widehat{\varphi}_1(Q^1_{d_1})\right),$$

we need to evaluate the isogeny $\widehat{\varphi}_{A,1}$ on points of order d_1. We can avoid such a computation by precomputing the image of the isogeny on a basis of $\tilde{E}_A[d_1]$ during KeyGen and expressing the points in terms of such a basis.

The same approach can be used to compute the points $\widehat{\varphi}_2(P^2_{d_2})$ and $\widehat{\varphi}_2(Q^2_{d_2})$, where we precompute the action of φ_{A_2} on a basis of $\tilde{E}_A[d_2]$, and the points $\widehat{\varphi}_1(R_1)$ and $\widehat{\varphi}_1(S_1)$ used to recover the matrix $\mathbf{B}$.

6.2 Computing Parameters

We propose a method to generate solutions of Eq. (1), i.e. finding parameters that allow us to efficiently run the trapdoor inversion algorithm. Given the security analysis of Sect. 4, we also have several additional requirements on the solutions we can use. In particular, we want all the solution values, i.e. $m_1, d_1, m_2, d_2, d_{A,1}, d_{A,2}$, to be odd, so that the isogenies have degree coprime with the torsion points order. Moreover, we require that isogeny degrees d_1, $d_A = d_{A,1}d_{A,2}$ and d_2 are pairwise coprime and sufficiently long, i.e. $\log(d_1), \log(d_A), \log(d_2) \geq 2\lambda$, to prevent meet-in-the-middle and torsion-guessing attacks.

The number of solutions and the corresponding protocol efficiency crucially depends on the smoothness of the degrees of the isogenies we are using. Let us denote our smoothness bound as B. Let c be a positive integer such that the number $T := 2^c - 1$ is B-smooth. We start by finding primitive solutions, i.e. solutions $(x, y) \in \mathbb{Z} \times \mathbb{Z}$ with $\gcd(x, y) = 1$, for the equation

$$x^2 + y^2\, T = 2^b. \tag{2}$$

We do so by ranging the value b within a reasonable interval, and finding solutions of Eq. (2) via Cornacchia's algorithm [15]. Given a primitive solution (x, y) for some even $b > 0$, we have

$$y^2\, T = (2^{b/2} - x)(2^{b/2} + x).$$

Define T_1 to be the B-smooth part of $2^{b/2} - x$ and T_2 to be the B-smooth part of $2^{b/2} + x$. Then, there exist $m_1, m_2 \in \mathbb{Z}_{>0}$ such that $m_1^2 T_1 = 2^{b/2} - x$ and $m_2^2 T_2 = 2^{b/2} + x$. In particular, we have

$$m_1^2 T_1 + m_2^2 T_2 = 2^{b/2+1}.$$

If $T_1 T_2$ and T_i are sufficiently large to guarantee security (we need $T_1 T_2 > 2^{6\lambda}$ and $T_i > 2^{2\lambda}$), we define d_i to be the smoothest factor of T_i such that $d_i \sim 2^{2\lambda}$ for $i = 1, 2$. Additionally, we define $d_{A,i}$ to be the smoothest part of T_i/d_i such that $d_{A,1}d_{A,2} > 2^{2\lambda}$ and multiply m_i by $\sqrt{T_i/(d_i d_{A,i})}$, ensuring that the values d_1, $d_{A,1}$, $d_{A,2}$, d_2, m_1, m_2 are pairwise coprime. We thus have found solutions of Eq. (1) that guarantee our size requirements, i.e. a valid set of parameters.

To find parameter sets, we perform an exhaustive search ranging over different values of b and c within a reasonable interval. Experimentally, this approach is highly efficient, and it is easy to generate parameter sets for any security level. Ideally, to have a small prime p, we look for small b's satisfying the conditions above. Allowing for a larger smoothness bound B, it is possible to find smaller b's and in turn smaller primes p. This comes with a potential slowdown in performances. Different trade-offs between efficiency and bandwidth can be achieved: if bandwidth is a more valuable asset, then one could allow larger smoothness bound. Note that as well as reducing the size of the base field characteristic, reducing the size of b shortens the length of the $(2^b, 2^b)$-isogeny, which in turn speeds up the decryption algorithm.

6.3 Further Optimisations

We designed FESTA and chose its parameters to obtain an optimal trade-off between the performance of the three PKE algorithms (KeyGen, Enc, Dec), the size of public keys and ciphertexts, the hardness of the security assumptions, and the simplicity of the protocol. Many other options are possible: for instance, increasing the smoothness bound for the solutions of Eq. (2) leads to smaller primes (and thus smaller ciphertexts), at the cost of slower isogeny computations.

In this section, we discuss further optimisations that may lead to different trade-offs or that require further work to investigate.

Using Larger (ℓ, ℓ)-Isogenies. In the search for parameters, we restrict ourselves to torsion points of order a power of two. There is no fundamental reason why torsion points of odd order cannot be used; however, the inversion function needs to compute (ℓ, ℓ)-isogenies for any ℓ dividing the order of the torsion points. Currently, the formulae to compute (ℓ, ℓ)-isogenies are most practical for $\ell = 2$. However, future developments in the computation of isogenies between principally polarised abelian surfaces may render new parameter sets feasible.

The method we propose to find parameters in Sect. 6.2 generalises to any prime power. In other words, it appears that analysing the equation $x^2 + y^2 T = \ell^b$, when T is of the form $T = \ell^c - 1$ for increasing $b \in \mathbb{Z}_{>0}$, leads to smooth solutions of $m_1^2 d_1 d_{A,1} + m_2^2 d_2 d_{A,2} = \ell^{b/2+1}$. However, the same method does not appear to generalise for products of prime powers; for those parameters, it may be necessary to develop new tools to efficiently find parameter sets.

Higher-Dimensional Trapdoor Inversion. In [48, Section 6.4], Robert proposed a method that relies on isogenies in dimension four (heuristically, eight otherwise) to recover a d-isogeny from the map of the m-torsion under such an isogeny for $m^2 > 4d$. This method could be employed to obtain smaller parameters. For instance, given a security parameter λ, we define e_1, e_2 and e_3 such that $\ell_i^{e_i} > 2^{2\lambda}$ for some distinct odd small prime ℓ_i; then, we can use isogenies φ_1, φ_A and φ_2 with degrees $\ell_1^{e_1}$, $\ell_2^{e_2}$ and $\ell_3^{e_3}$, respectively. We can also choose b such that $2^b > 2\sqrt{\ell_1^{e_1}\ell_2^{e_2}\ell_3^{e_3}}$. With such parameters, we obtain a significantly smaller prime (roughly, $p \approx 2^{7\lambda}$) since the isogeny φ_A does not need to be rational. We expect a major improvement in the protocol bandwidth, as well as the running times of key generation and encryption. This would come at the cost of decryption efficiency, which would require the computation of higher dimensional isogenies.

It is also possible to introduce even more extreme trade-offs: by using irrational isogenies φ_1 and φ_2,[5] one can achieve very small primes that further improve the compactness of the protocol and the efficiency of the key generation

[5] The isogenies φ_i are irrational in the sense that each prime-degree isogeny factoring φ_i has kernel defined over $\mathbb{F}_{p^2}$, but the kernel of φ_i is not necessarily defined over $\mathbb{F}_{p^2}$.

and encryption procedures; however, this requires computing higher dimensional isogenies several times, which slows down the decryption algorithm even more.

7 Implementation

We provide a proof-of-concept implementation of FESTA in SageMath [51] and make it available at:

https://github.com/FESTA-PKE/FESTA-SageMath

We designed our implementation to be modular to facilitate translation into a high-performance language. In particular, we aimed to explicitly implement the algorithms for both the isogenies between elliptic curves and abelian varieties, without relying on the generic Sagemath implementations. In what follows, we explain some of the techniques we employed, and we propose concrete parameters.

7.1 Montgomery Curve x-Only Isogenies

To compute isogenies between elliptic curves, we leverage the efficient x-only formulae between Montgomery curves [17,46]. Additionally, we include $\sqrt{\text{élu}}$ [5] to evaluate isogenies of large prime degree and the formula using twisted Edwards curves in [38] for the computation of the codomain curves. Working with the x-only isogenies allows a significant improvement in performance, however, the y-coordinates of image points must eventually be recovered in order to compute the chain of $(2,2)$-isogenies between elliptic products. We use the following method to reconstruct the valid y-coordinates from the x-only point evaluation.

Let $E\colon y^2 = x^3 + Ax^2 + x$ and $E'\colon y^2 = x^3 + A'x^2 + x$ be two elliptic curves in Montgomery form connected by a d-isogeny $\varphi\colon E \to E'$ and suppose we want to evaluate the action of φ on P, Q generating the n-torsion, where $n \neq d$. Using the x-only isogeny formulae, we compute $x(\varphi(P))$ and $x(\varphi(Q))$. Then, we lift the x-coordinates onto the curve by computing y_P and y_Q, which we allow to be any square root of $x(\varphi(P))^3 + A'x(\varphi(P))^2 + x(\varphi(P))$ and $x(\varphi(Q))^3 + A'x(\varphi(Q))^2 + x(\varphi(Q))$. From this lifting, we effectively recover $\varphi(P) = \pm(x(\varphi(P)), y_P)$ up to an overall sign.

Although we cannot recover the correct sign for one point, we can recover a relative sign such that we recover the tuple $\pm(\varphi(P), \varphi(Q))$. To do this, we use the Weil pairing and compare

$$e_n^E(P,Q)^d \stackrel{?}{=} e_n^{E'}\big((x(\varphi(P)), y_P), (x(\varphi(Q)), y_Q)\big).$$

If the equality holds, then the lifted points can be used as the image, otherwise we flip a sign such that $\varphi(Q) = (x(\varphi(Q)), -y_Q)$. In this way, we evaluate the action of either φ or $-\varphi$ on the torsion basis. Given that we use canonical representations of the scaling matrices in $\mathcal{M}_b$, evaluating either of these isogenies does not represent a problem for the trapdoor. Overall, this allows us to perform two isogeny evaluations using x-only formula with the additional cost of two square-roots and two Weil pairings.

7.2 Optimisations of the (2, 2)-Isogeny Chain

The most expensive step of decryption is the evaluation of the $(2,2)-$isogeny chain between elliptic products. Of this computation, the majority of the cost is spent computing the isogenies via the Richelot correspondence between Jacobians of genus-2 hyperelliptic curves. The computational cost, just as in the elliptic case, is split between doubling to recover divisors of order two, and the evaluation of the isogenies themselves.

One optimisation, which we can borrow from our experience with elliptic curve isogenies, is to employ the optimal strategies introduced in [22] to minimise the cost of long isogeny chains. The generalisation of this problem to higher dimensional isogeny chains was recently studied in [13]. By measuring the relative cost of divisor doubling and isogeny evaluations, we can compute an optimal strategy using identical methods to the elliptic case.

Another performance improvement we made comes from deriving explicit addition and doubling algorithms for divisors of Jacobians of genus-2 hyperelliptic curves using only base field operations. Previously in the literature, effort was made to derive low cost genus-2 addition and doubling for the context of hyperelliptic Diffie-Hellman [8,18,34]. In this case, the hyperelliptic curve is fixed, and isomorphisms can be used to minimise the number of non-zero coefficients of the hyperelliptic curve model, effectively reducing the cost of divisor arithmetic.

For our implementation, the curve model of the codomain when computed via the Richelot correspondence is some generic (non-monic) sextic polynomial and the previously derived efficient formula are unsuitable for our doubling chains. To recover efficient formula, we generalise the work of [18] by using similar methods without restricting the hyperelliptic curve model to a friendly form. Ultimately, what is required is to solve linear equations to express the arithmetic in the polynomial ring $\mathbb{F}_q[X]$ as operations of the base field and pass as arguments to the addition and doubling formula the coefficients of the curve equation as well as the reduced Mumford coordinates.

Representing the cost of $\mathbb{F}_{p^2}$ inversion, multiplication and squaring as $\mathbf{I}, \mathbf{M}$ and $\mathbf{S}$, we have derived affine addition at a cost of $25\mathbf{M} + 4\mathbf{S} + 1\mathbf{I}$ and affine doubling at a cost of $32\mathbf{M} + 6\mathbf{S} + 1\mathbf{I}$. Practically, we find that our formulae are about four-times faster than the in-built SageMath divisor arithmetic and two-times faster than the optimised formula used in the SageMath implementation of the Castryck-Decru attack on SIDH [41]. We consider further improving these formulae to be future work.

7.3 Parameters

Following the approach in Sect. 6.2, we generated parameter sets for FESTA. We highlight that the proposed techniques allow for different trade-offs between the smoothness of the isogeny degrees and the length b of the chain of $(2,2)$-isogenies.

For FESTA-128, we define the following parameter set:

$$
\begin{aligned}
b &:= 632,\\
d_1 &:= (3^3 \cdot 19 \cdot 29 \cdot 37 \cdot 83 \cdot 139 \cdot 167 \cdot 251 \cdot 419 \cdot 421 \cdot 701 \cdot 839 \cdot 1009 \cdot \\
&\quad\ 1259 \cdot 3061 \cdot 3779)^2,\\
d_2 &:= 7 \cdot (5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 41 \cdot 43 \cdot 71 \cdot 89 \cdot 127 \cdot 211 \cdot 281 \cdot 503 \cdot 631 \cdot \\
&\quad\ 2309 \cdot 2521 \cdot 2647 \cdot 2729)^2,\\
d_{A,1} &:= (59 \cdot 6299 \cdot 6719 \cdot 9181)^2,\\
d_{A,2} &:= (3023 \cdot 3359 \cdot 4409 \cdot 5039 \cdot 19531 \cdot 22679 \cdot 41161)^2,\\
m_1 &:= 14921849450934765925202420839250441821039 21,\\
m_2 &:= 256173313364299393001666930 69,\\
f &:= 107.
\end{aligned}
$$

The values d_1 and d_2 are 2^{12}-smooth, while $d_A = d_{A,1}d_{A,2}$ is 2^{16}-smooth. The corresponding prime, defined as $p = 2^b d_1 (d_{A,1}d_{A,2})_{\mathsf{sf}} d_2 f - 1$, is 1292-bit long. The public key and ciphertext sizes are, respectively, 561 and 1,122 bytes. The same approach can be used to produce parameter sets for higher security levels.

To reduce the bandwidth of FESTA, we compress the torsion points by expressing them in terms of linear coefficients of canonical bases, as proposed in [40,43]. Unlike in SIDH, however, our protocol needs the exact torsion images. This means the points cannot be scaled, and their compressed representation requires four coefficients of size equal to their order. However, since an attacker can always reconstruct the determinant of the scaling matrices, we can restrict ourselves to unitary matrices in $\mathcal{M}_b$. Then, given three of the four coefficients representing the scaled points, it is possible to retrieve the fourth one via the compatibility of the Weil pairing with isogenies. We remark the bandwidth of FESTA is affected by the choice of parameters: future developments may lead to smaller parameters, which would translate to significantly smaller public keys and ciphertexts.

We benchmarked our proof-of-concept implementation on an Apple M1 PRO CPU clocked at 3.2 GHz using a single performance core. Averaging 100 executions, we obtained that KeyGen, Enc and Dec run in 4.47, 3.09 and 9.14 s, respectively. The slowness of Dec compared to the other components is mainly caused by the computation of $(2, 2)$-isogenies. Due to the lack of research on optimizing such computations in the past, we expect future work to significantly improve on this aspect, leading to a much faster decryption algorithm.

8 Conclusion

In this paper, we have introduced FESTA, an efficient isogeny-based public-key encryption (PKE) protocol that constructively relies on an application of the SIDH attacks. Preliminary experimental results show that our proof-of-concept

implementation is competitive with optimised implementations of other isogeny-based PKEs. We are also currently working on an optimised implementation of FESTA, and we are looking forward to obtaining concrete running times.

The efficiency of the protocol is highly dependent on the smoothness bounds and size of the parameter sets: in future work, we will investigate different approaches to find more efficient parameters. In particular, our current choice of parameters is limited by the requirement of ensuring a fast decryption: a more optimised implementation of $(2, 2)$-isogenies will allow us to use smoother values. An interesting project for future work is to compare the performance of isogenies via the Richelot correspondence against those computed using theta functions.

Additionally, we chose a conservative approach when imposing the security requirements: we believe a more detailed analysis of the cost of certain attacks may lead to better parameter sets. Moreover, we highlight that FESTA can inherently benefit from advancements in computations of higher dimensional isogenies: new developments in those areas could lead to both smaller field characteristics and faster encryption.

Lastly, we believe that the flexible design of FESTA and the new techniques proposed in this work may lead to new mechanics that can be exploited to develop new advanced protocols, such as digital signatures and oblivious pseudorandom functions.

Acknowledgments. The authors are indebted to Tako Boris Fouotsa for fruitful feedback on a preliminary version of this paper that led to a more complete security analysis. The authors would also like to thank Ross Bowden, James Clements, Péter Kutas, and Chloe Martindale for useful discussions regarding the parameter generation, and Yan Bo Ti for an interesting discussion on potential adaptive attacks.

References

1. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 411–439. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_14
2. Basso, A., et al.: Supersingular curves you can trust. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023. LNCS, vol. 14005, pp. 405–437. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30617-4_14
3. Basso, A., Fouotsa, T.B.: New SIDH countermeasures for a more efficient key exchange. Cryptology ePrint Archive, Paper 2023/791, To appear in "Asiacrypt 2023" (2023). https://eprint.iacr.org/2023/791
4. Basso, A., Kutas, P., Merz, S.-P., Petit, C., Sanso, A.: Cryptanalysis of an oblivious PRF from supersingular isogenies. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 160–184. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92062-3_6
5. Bernstein, D.J., de Feo, L., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. In: Proceedings of the Fourteenth Algorithmic Number Theory Symposium, pp. 39–55. THE OPEN BOOK SERIES 4 (2020). https://doi.org/10.2140/obs.2020.4.39

6. Boneh, D., Kogan, D., Woo, K.: Oblivious pseudorandom functions from isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 520–550. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_18
7. Boneh, D., Shoup, V.: A Graduate Course in Applied Cryptography. Version 0.6 (2023). http://toc.cryptobook.us/
8. Bos, J.W., Costello, C., Hisil, H., Lauter, K.: Fast cryptography in genus 2. J. Cryptol. **29**(1), 28–60 (2016). https://doi.org/10.1007/s00145-014-9188-7
9. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023. LNCS, vol. 14008, pp. 423–447. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_15
10. 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
11. Castryck, W., Vercauteren, F.: A polynomial time attack on instances of M-SIDH and FESTA. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023. LNCS, vol. 14444, pp. 127–156. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-8739-9_5
12. Chen, M., Imran, M., Ivanyos, G., Kutas, P., Leroux, A., Petit, C.: Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH. Cryptology ePrint Archive, Paper 2023/779, To appear in "Asiacrypt 2023" (2023). https://eprint.iacr.org/2023/779
13. Chi-Domínguez, J.J., Pizarro-Madariaga, A., Riquelme, E.: Computing quotient groups of smooth order with applications to isogenies over higher-dimensional abelian varieties. Cryptology ePrint Archive, Paper 2023/508 (2023). https://eprint.iacr.org/2023/508
14. 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., 349–368 (2022). https://doi.org/10.1007/s13389-021-00271-w
15. Cornacchia, G.: Su di un metodo per la risoluzione in interi dell'equazione $\sum_{h=0}^{n} x^{n-h}y^h$. In: Giornale di Matematiche di Battaglini 46, pp. 33–90 (1908)
16. Cosset, R., Robert, D.: Computing (ℓ, ℓ)-isogenies in polynomial time on Jacobians of genus 2 curves. Math. Comput. **84**, 1953–1975 (2015). https://doi.org/10.1090/S0025-5718-2014-02899-8
17. Costello, C., Hisil, H.: A simple and compact algorithm for SIDH with arbitrary degree isogenies. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 303–329. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_11
18. Costello, C., Lauter, K.: Group law computations on Jacobians of hyperelliptic curves. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 92–117. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28496-0_6
19. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQISignHD: new dimensions in cryptography. Cryptology ePrint Archive, Paper 2023/436 (2023). https://eprint.iacr.org/2023/436
20. De Feo, L.: Mathematics of isogeny based cryptography. arXiv (2017). http://arxiv.org/abs/1711.04062
21. De Feo, L., et al.: Séta: supersingular encryption from torsion attacks. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 249–278. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92068-5_9

22. 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
23. Decru, T., Kunzweiler, S.: Efficient computation of $(3^n, 3^n)$-isogenies. In: El Mrabet, N., De Feo, L., Duquesne, S. (eds.) Progress in Cryptology - AFRICACRYPT 2023, pp. 53–78. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37679-5_3
24. Decru, T., Maino, L., Sanso, A.: Towards a quantum-resistant weak verifiable delay function. In: Aly, A., Tibouchi, M. (eds.) Progress in Cryptology – LATINCRYPT 2023. LATINCRYPT 2023. LNCS, vol. 14168. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-44469-2_8
25. Ebrahimi, E.: Post-quantum security of plain OAEP transform. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part I. LNCS, vol. 13177, pp. 34–51. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-030-97121-2_2
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.) Advances in Cryptology - EUROCRYPT 2023, pp. 282–309. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_10
27. Fujisaki, E., Okamoto, T.: Statistical zero knowledge protocols to prove modular polynomial relations. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 16–30. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052225
28. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34
29. Galbraith, S.D., Vercauteren, F.: Computational problems in supersingular elliptic curve isogenies. Quantum Inf. Process. **17**(10), 265 (2018). https://doi.org/10.1007/s11128-018-2023-6
30. Hohenberger, S., Koppula, V., Waters, B.: Chosen ciphertext security from injective trapdoor functions. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 836–866. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56784-2_28
31. Jaques, S., Schanck, J.M.: Quantum cryptanalysis in the RAM model: claw-finding attacks on SIKE. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 32–61. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_2
32. Kani, E.: The number of curves of genus two with elliptic differentials (1997). https://doi.org/10.1515/crll.1997.485.93
33. Lamport, L.: Constructing digital signatures from a one way function. Technical report, SRI International (1979)
34. Lange, T.: Efficient arithmetic on genus 2 hyperelliptic curves over finite fields via explicit formulae. Cryptology ePrint Archive, Paper 2002/121 (2002). https://eprint.iacr.org/2002/121
35. Leroux, A.: A new isogeny representation and applications to cryptography. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part II. LNCS, vol. 13792, pp. 3–35. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-22966-4_1
36. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023, pp. 448–471. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_16
37. Martindale, C., Panny, L.: How to not break SIDH. Cryptology ePrint Archive, Report 2019/558 (2019). https://eprint.iacr.org/2019/558

38. Meyer, M., Reith, S.: A faster way to the CSIDH. In: Chakraborty, D., Iwata, T. (eds.) INDOCRYPT 2018. LNCS, vol. 11356, pp. 137–152. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05378-9_8
39. Milne, J.S.: Arithmetic Geometry. Springer, New York (1986). https://doi.org/10.1007/978-1-4613-8655-1_5
40. Naehrig, M., Renes, J.: Dual isogenies and their application to public-key compression for isogeny-based cryptography. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part II. LNCS, vol. 11922, pp. 243–272. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34621-8_9
41. Oudompheng, R., Pope, G.: A note on reimplementing the Castryck-Decru attack and lessons learned for SageMath. Cryptology ePrint Archive, Paper 2022/1283 (2022). https://eprint.iacr.org/2022/1283
42. 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
43. Pereira, G.C.C.F., Doliskani, J., Jao, D.: x-only point addition formula and faster compressed SIKE. J. Cryptogr. Eng. **11**(1), 57–69 (2021). https://doi.org/10.1007/s13389-020-00245-4
44. Petit, C.: Faster algorithms for isogeny problems using torsion point images. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 330–353. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_12
45. de Quehen, V., Kutas, P., Leonardi, C., Martindale, C., Panny, L., Petit, C., Stange, K.E.: Improved torsion-point attacks on SIDH variants. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part III. LNCS, vol. 12827, pp. 432–470. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84252-9_15
46. Renes, J.: Computing isogenies between montgomery curves using the action of (0, 0). In: Lange, T., Steinwandt, R. (eds.) PQCrypto 2018. LNCS, vol. 10786, pp. 229–247. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_11
47. Richelot, F.: Ueber die Integration eines merkwürdigen Systems Differentialgleichungen (1842)
48. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023, pp. 472–503. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_17
49. Silverman, J.H.: The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics, vol. 106. Springer, New York (1986). https://doi.org/10.1007/978-1-4757-1920-8
50. Smith, B.: Explicit endomorphisms and correspondences. Ph.D. thesis, The University of Sydney (2005). http://hdl.handle.net/2123/1066
51. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 9.8) (2023). https://www.sagemath.org
52. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: a generic power/EM analysis on post-quantum KEMs. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**(1), 296–322 (2021). https://doi.org/10.46586/tches.v2022.i1.296-322