M-SIDH and MD-SIDH: Countering SIDH Attacks by Masking Information

Tako Boris Fouotsa[1(✉)], Tomoki Moriya[2], and Christophe Petit[3,4]

[1] LASEC-EPFL, Lausanne, Switzerland
tako.fouotsa@epfl.ch
[2] The University of Tokyo, Tokyo, Japan
tomoki_moriya@mist.i.u-tokyo.ac.jp
[3] Université Libre de Bruxelles, Brussels, Belgium
Christophe.Petit@ulb.be
[4] University of Birmingham, Birmingham, UK

Abstract. The SIDH protocol is an isogeny-based key exchange protocol using supersingular isogenies, designed by Jao and De Feo in 2011. The protocol underlies the SIKE algorithm which advanced to the fourth round of NIST's post-quantum standardization project in May 2022. The algorithm was considered very promising: indeed the most significant attacks against SIDH were meet-in-the-middle variants with exponential complexity, and torsion point attacks which only applied to unbalanced parameters (and in particular, not to SIKE).

This security picture dramatically changed in August 2022 with new attacks by Castryck-Decru, Maino-Martindale and Robert. Like prior attacks on unbalanced versions, these new attacks exploit torsion point information provided in the SIDH protocol. Crucially however, the new attacks embed the isogeny problem into a similar isogeny problem in a higher dimension to also affect the balanced parameters. As a result of these works, the SIKE algorithm is now fully broken both in theory and in practice.

Given the considerable interest attracted by SIKE and related protocols in recent years, it is natural to seek countermeasures to the new attacks. In this paper, we introduce two such countermeasures based on partially hiding the isogeny degrees and torsion point information in the SIDH protocol. We present a preliminary analysis of the resulting schemes including non-trivial generalizations of prior attacks. Based on this analysis we suggest parameters for our M-SIDH variant with public key sizes of 4434, 7037 and 9750 bytes respectively for NIST security levels 1, 3, 5.

Keywords: Isogenies · SIDH attacks · Countermeasures · M-SIDH · MD-SIDH

1 Introduction

In 1994, Peter Shor [35] described a polynomial quantum algorithm to solve the integer factorization problem and the discrete logarithm problem. This implies

C. Hazay and M. Stam (Eds.): EUROCRYPT 2023, LNCS 14008, pp. 282–309, 2023.
https://doi.org/10.1007/978-3-031-30589-4_10

that the widely deployed cryptographic protocols we use today would become vulnerable in presence of a large-scale quantum computer. To mitigate this threat, research on *post-quantum cryptography*, namely cryptographic protocols that will hopefully remain secure against both classical and quantum computers, has considerably developed in the last two decades. Several standardization competitions were initiated, among which the NIST PQC [29]. Many new candidates for post-quantum hard problems have been suggested to date based on lattices, codes, *isogenies*, multivariate systems of equations, and other problems.

Isogenies are maps between elliptic curves. For cryptographic applications, we restrict ourselves to curves defined over finite fields $\mathbb{F}_q$. When there exists an isogeny $\phi : E \to E'$, we say the elliptic curves E and E' are *isogenous*. There are infinitely many isogenies connecting two isogenous elliptic curves. The *pure isogeny problem* is stated as follows.

Problem 1. Given two isogenous elliptic curves E and E', compute an isogeny from E to E'.

An isogeny from a curve E to itself is called an *endomorphism of* E, and the set of all the endomorphisms of E (together with the 0 map) is called the *endomorphism ring of* E.

Over finite fields, there are two categories of elliptic curves, namely *ordinary elliptic curves* and *supersingular elliptic curves*. The endomorphism ring of any ordinary curve is an order in a quadratic imaginary field (hence is commutative), whereas the endomorphism ring of a supersingular curve is a maximal order in a quaternion algebra (hence is non-commutative). Isogenies connect ordinary curves between themselves and supersingular curves between themselves.

There is a straightforward adaptation of the well-known Diffie-Hellman key exchange protocol to isogenies in the ordinary/commutative case. This is in fact what is done in isogeny based schemes like CRS [14,34], CSIDH [8], OSIDH [10] and derivatives. The high level idea is as follows: there is a starting curve E_0. Alice selects a secret isogeny $\phi_A : E_0 \to E_A$, and Bob selects a secret isogeny $\phi_B : E_0 \to E_B$. Both parties exchange E_A and E_B. Each party recomputes their secret isogeny from the other party's public curve. Since the isogenies "are commutative", they get the same end curve E_{AB} (up to isomorphism) whose j-invariant is used as the shared secret. Note that the isogeny $E_A \to E_{AB}$ computed by Alice is not exactly the isogeny ϕ_A since they do not have the same domain and codomain. This isogeny is in general denoted by ϕ'_A. Similarly, Bob's isogeny is also denoted by ϕ'_B. This is illustrated in Fig. 1 where we have $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A$.

When it comes to using supersingular curves, designing a Diffie-Hellman type key exchange is less straightforward because their endomorphism rings are non-commutative. In 2011, Jao and De Feo [24] had a brilliant idea on how to obtain a commutative diagram in the supersingular case. To achieve this goal:

1. One fixes the respective degrees A and B of the isogenies ϕ_A and ϕ_B, with A and B coprime;
2. Alice reveals the images of a basis of the B-torsion of E_0 (that is $E_0[B]$), and Bob reveals the image of a basis of the A-torsion of E_0 (that is $E_0[A]$).

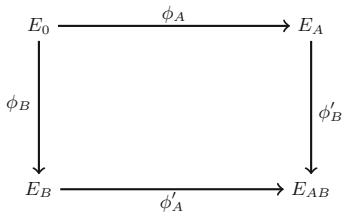

Fig. 1. Generic commutative isogeny key exchange.

This idea led to the Supersingular Isogeny Diffie-Hellman protocol (SIDH), which has received a lot of attention in the last decade. This protocol and other isogeny-based schemes are attractive for their very compact secret and public keys. This has been one of the most valuable advantages of SIKE, a Key Encapsulation Mechanism (KEM) derived from SIDH, compared to other post-quantum KEMs. SIKE became widely regarded as a promising post-quantum candidate for standardization, and in particular the algorithm made it to the 4th round of the NIST competition.

With the exception of the CGL hash function [9] and GPS signatures [21], most isogeny-based protocols in the literature do not directly rely on the pure isogeny problem, but on some variants of this problem. In the case of SIDH/SIKE, an attacker is provided with additional non-trivial information: the degree of the isogeny and the images of torsion points. More precisely, the security of SIDH relies on the following problem.

Problem 2. Let E_0 be a supersingular curve defined over $\mathbb{F}_{p^2}$ with $p = AB-1$. Set $E_0[A] = \langle P, Q\rangle$. Let $\phi : E_0 \to E'$ be an isogeny of degree B and let $P' = \phi(P)$, $Q' = \phi(Q)$. Given E_0, P, Q, E', P' and Q', compute the isogeny ϕ.

Furthermore, in SIDH/SIKE, the starting curve E_0 is special: its endomorphism ring is publicly available. In 2017, Petit [31] exploited the knowledge of the endomorphism ring of E_0 and the torsion point information to design attacks that recover the secret isogeny in polynomial time assuming that $A \ll B$. After a recent improvement [32], the attack still required imbalanced A and B, hence it does not apply to SIDH where $A \approx B$.

In July 2022, Castryck and Decru [7] described devastating attacks on SIDH that recovered the secret key in SIDH and SIKE, instantiated with the NIST parameters, in a few hours. The attacks were also developed in a concurrent work by Maino and Martindale [26]. Various follow-up works by other authors quickly improved the practical runtime time to minutes and seconds, and clarified the asymptotic complexities. The best attacks on balanced SIDH parameters had suddenly gone from exponential time to subexponential time, with a further reduction to polynomial time complexity when the endomorphism ring of the starting curve is available (as was the case in SIKE). Things could only get worse for SIDH, and a few days later they did, when Robert described an

improved attack with polynomial time complexity working for arbitrary starting curves [33].

We note that like Petit's attacks before them, the Castryck-Decru-Maino-Martindale-Robert attacks exploit knowledge of both torsion point information and the degree of the secret isogeny.

While the Castryck-Decru-Maino-Martindale-Robert attacks constitute a clear cryptanalysis breakthrough on a flagship isogeny-based cryptographic protocol, they do not apply to other isogeny-based schemes in which no torsion point information is revealed: CRS [14,34], CSIDH [8] and CSIDH-based signatures (SeaSign [15], SCI-FiSh [3], ...), CGL [9], GPS [21], SQISign [17], and many more. The new attacks do not imply that the whole of isogeny-based cryptography is insecure, but only that the field is getting mature! In particular, a natural question now is whether one can find countermeasures against the Castryck-Decru-Maino-Martindale-Robert attacks and repair the SIDH protocol.

Contributions[1]. In this paper, we propose and analyze two countermeasure candidates to the Castryck-Decru attack: Masked-Degree SIDH (MD-SIDH) and Masked torsion points SIDH (M-SIDH).

The main idea in MD-SIDH is to mask the degree of the secret isogeny: the degrees A and B of the secret isogenies in SIDH are no longer fixed, but uniformly random divisors of A and B respectively. To prevent the degree from being recovered by a pairing computation and some discrete logarithms in a group of smooth order, the images of the torsion points are scaled by a random integer.

The main idea in M-SIDH is to keep the degrees of the secret isogenies fixed as in SIDH, and mask only the torsion point information: the images of the torsion points are scaled by a random integer. To prevent an efficient recovery of the secret scalar used in M-SIDH (using pairings and discrete logarithms), we set the isogeny degrees A and B to have $t \geq 2\lambda$ distinct prime divisors, so that the scalar cannot be recovered despite the fact that its square modulo A or B is known.

We perform a thorough security analysis of the two countermeasures, including non-trivial extensions of prior attacks. In particular, we give an expected polynomial time attack on the M-SIDH variant when the starting curve has a known small endomorphism, and a reduction from any MD-SIDH instance to an M-SIDH instance. We also show that isogeny degrees in the M-SIDH variant must have at least 2λ distinct factors, where λ is the security parameter. Finally, we provide non-trivial variants of adaptive attacks on SIDH, including the GPST attack and the Fouotsa-Petit attack.

Based on our analysis, the M-SIDH variant is the most promising one as it features smaller keys at identical security levels. The variant must be used with a randomly generated starting curve to avert the attack mentioned above (note that this is not an issue in a key encapsulation mechanism as the starting curve may be constructed by the key generation algorithm). Our analysis suggests

[1] This paper is an extended merge of the preprints [18] and [27].

that public key sizes of 4434, 7037 and 9750 bytes are sufficient to reach AES-128, AES-192 and AES-256 security levels (NIST security levels 1, 3, 5), and asymptotically public keys should be a factor $O(\log \lambda)$ larger than in SIDH.

1.1 Outline

In Sect. 2 we briefly present the SIDH protocol and discuss attacks on SIDH. In Sect. 3 we describe our two constructions Masked-degree SIDH and Masked SIDH (M-SIDH). In Sects. 4, 5 and 6, we do a security analysis of both schemes and in Sect. 7 we suggest parameters. We conclude the paper in Sect. 8.

2 The SIDH Protocol and Attacks

The Supersingular Isogeny Diffie-Hellman protocol (SIDH) is a key exchange protocol designed by Jao and De Feo [24], which underlies the SIKE submission to NIST post-quantum cryptography project [23]. Interest in the SIDH protocol grew steadily since 2011, but passive cryptanalytic success remained limited until new attacks fully broke it in August 2022.

2.1 The SIDH Protocol

The SIDH protocol is a Diffie-Hellman-like key exchange scheme that uses torsion point information to complete a (pseudo) commutative diagram:

$$\begin{array}{ccc} (E_0, P_A, Q_A, P_B, Q_B) & \longrightarrow & (E_A, \phi_A(P_B), \phi_A(Q_B)) \\ \downarrow & & \downarrow \\ (E_B, \phi_B(P_A), \phi_B(Q_A)) & \longrightarrow & E_{AB} \cong E_{BA} \end{array}$$

The precise scheme is as follows:

Public parameter: Let E_0 be the elliptic curve of j-invariant 1728. Set a prime p as $p = 2^{e_A}3^{e_B} - 1$. Let P_A and Q_A (resp. P_B and Q_B) be points generating $E_0[2^{e_A}] \cong (\mathbb{Z}/2^{e_A}\mathbb{Z})^2$ (resp. $E_0[3^{e_B}] \cong (\mathbb{Z}/3^{e_B}\mathbb{Z})^2$).

Public key (Alice): Alice first generates a random value $k_A \in (\mathbb{Z}/2^{e_A}\mathbb{Z})^\times$ as her secret key. Let $R_A = P_A + k_A Q_A$. Alice computes an isogeny $\phi_A \colon E_0 \to E_A := E_0/\langle R_A \rangle$ and image points $\phi_A(P_B), \phi_A(Q_B)$. Alice sends to Bob E_A and these image points as a public key.

Public key (Bob): Bob first generates a random value $k_B \in (\mathbb{Z}/3^{e_B}\mathbb{Z})^\times$ as his secret key. Let $R_B = P_B + k_B Q_B$. Bob computes an isogeny $\phi_B \colon E_0 \to E_B := E_0/\langle R_B \rangle$ and image points $\phi_A(P_B), \phi_A(Q_B)$. Bob sends to Alice E_B and these image points as a public key. Let k_B be his secret key.

Shared key: Let $R'_A = \phi_B(P_A) + k_A\phi_B(Q_A)$, and let $R'_B = \phi_A(P_B) + k_B\phi_A(Q_B)$. Alice computes $E_{AB} := E_B/\langle R'_A \rangle$, and Bob computes an isogeny $E_{BA} := E_A/\langle R'_B \rangle$. The value $j(E_{AB}) = j(E_{BA})$ is the shared key.

The SIDH protocol is the basis of the SIKE algorithm, which was selected for Round 4 of NIST post-quantum standardization project in June 2022.

2.2 Cryptanalysis Attempts and Successes

A natural problem to consider in the cryptanalysis of SIDH is the isogeny with torsion problem: Problem 2 with $A = 2^{e_A}$ and $B = 3^{e_B}$. One approach to solve this problem is to entirely ignore the torsion point information, and recover the isogeny with some advanced brute force strategy such as a meet-in-the-middle algorithm or Van Oorschot-Wiener's algorithm [36]. These approaches have guided the parameter selection of SIKE submission to NIST [23].

The first passive attacks exploiting torsion point information[2] were introduced by Petit [31]. The key idea in his attack is to consider an endomorphism Ψ of E' of the form

$$\psi = d + \phi \circ \theta \circ \hat{\phi}$$

where $d \in \mathbb{Z}$ and θ is a trace 0 non scalar endomorphism of E_0 such that

$$\deg \psi = d^2 + B^2 \deg \theta$$

divides A. Provided such parameters, one can use torsion point information on ϕ to deduce torsion point information on ψ, then use this information to recover $\ker \psi$ via a (smooth order hence efficient) discrete logarithm computation, and finally deduce ϕ. Note that this strategy requires at least partial knowledge of the endomorphism ring of E_0 (for θ) and moreover it only works if A is large enough compared to B. (In particular, it does not work when $A \approx B$ as in SIKE.) Improvements in later work increased the range of parameters vulnerable to these attacks but they did not fundamentally change these limitations [5,32].

In August 2022, Castryck and Decru [7] (and independently Maino and Martindale [26]) introduced new powerful attacks against SIDH, with polynomial time complexity when the endomorphism ring of E_0 is known and subexponential complexity in general. Extensions by Robert [33] further reduced the complexity to polynomial time in the general case. In a sense, these attacks can be seen as generalizations of previous torsion point attacks[3], but with a key additional insight: they crucially embed the SIDH isogeny problem into a higher dimensional isogeny problem, where more endomorphisms are readily available. In a nutshell, Robert's attack considers the genus 8 Abelian variety $E_0^4 \times E'^4$, and its endomorphism

$$\Psi = \begin{pmatrix} \alpha_0 & \hat{\Phi} \\ -\Phi & \hat{\alpha'} \end{pmatrix},$$

[2] Torsion point information was previously used in active attacks against SIDH [20] and prompted the inclusion of a CCA transform (a variant of Fujisaki-Okamoto transform) within SIKE.

[3] The original Castryck-Decru's paper did not initially make a connection with prior torsion point attacks, but this connection then rapidly emerged and is clearly described in [26,33].

where Φ is the natural extension of ϕ on E_0^4, $\hat{\Phi}$ is its dual, and α_0 and α' are the endomorphisms on E_0^4 and E'^4 with action given by the matrix

$$M = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ -a_1 & a_0 & -a_4 & a_3 \\ -a_3 & a_4 & -a_1 & a_2 \\ -a_4 & -a_3 & a_2 & a_1 \end{pmatrix}$$

with a_0, a_1, a_2, a_3 such that $a := a_0^2 + a_1^2 + a_2^2 + a_3^2 = A - B$. We then have

$$\Psi\hat{\Psi} = AI_8$$

where $\hat{\Psi} = \begin{pmatrix} \hat{\alpha}_0 & -\hat{\Phi} \\ \Phi & \alpha' \end{pmatrix}$ is the dual of Ψ, i.e., Ψ is an endomorphism of degree A. As in previous torsion point attacks, one can evaluate Ψ on the A torsion using torsion point information provided in the SIDH problem. One can then compute Ψ and finally deduce ϕ.

From now on we will refer to these attacks as "the CD-MM-R attacks". Compared to previous torsion point attacks, these new attacks do not require any knowledge on the endomorphism ring of the starting curve, and work whenever $A \geq B$. One can further improve this to $A \geq \sqrt{B}$ as in the "dual isogeny variant" of [32]: let $a := A^2 - B$; recover the first halves of the endomorphisms Ψ and $\hat{\Psi}$ using torsion point information; and finally deduce the whole of Ψ and ϕ [33].

While they do not require any knowledge of $\mathrm{End}(E_0)$, the new attacks still use torsion point information and general SIDH parameters, including the isogeny degrees. In the following section, we describe two countermeasures: the first one consists in making the torsion point images while the second one consists in masking the isogeny degrees.

3 Masked SIDH Variants

Recall that the CD-MM-R attack requires two main ingredients:

1. the degree A of the secret supersingular isogeny $\phi : E_0 \to E$;
2. the images $\phi(P)$, $\phi(Q)$ of a torsion basis (P, Q) of the B-torsion $E_0[B]$ where B is an integer coprime to A such that $B > A$.

The countermeasures we suggest here consist in masking each of the above. Firstly, we suggest Masked torsion points SIDH or M-SIDH for short, in which one masks the torsion point images by scaling them with a random scalar. Secondly, we suggest Masked-degree SIDH or MD-SIDH for short, in which the isogenies computed do not have a fixed degree.

In the rest of this paper, we will often use the following lemma.

Lemma 3. *Let $\phi : E \longrightarrow E'$ be an isogeny of unknown degree d and let B be a smooth integer coprime to d such that $E[B] \subset E(\mathbb{F}_{p^2})$. Set $E[B] = \langle P, Q \rangle$. Then given P, Q, $\phi(P)$ and $\phi(Q)$, there exists a polynomial time algorithm to recover $d \bmod B$.*

Proof. One computes the Weil pairing values $e_B(P,Q)$ and $e_B(\phi(P),\phi(Q)) = e_B(P,Q)^{\deg\phi}$, then one solves a discrete logarithm instance between both quantities to recover $d \bmod B$. Since $E[B] \subset E(\mathbb{F}_{p^2})$, the pairing computations run in polynomial time. Since B is smooth, then using the Pohlig-Hellman algorithm the discrete logarithm computation runs in polynomial time as well. □

3.1 Masked Torsion Points Variant

The aim here is to instantiate SIDH such that the direct images $\phi(P)$, $\phi(Q)$ of P and Q are not available to adversaries, but the key exchange still succeeds: this means that when given a point $R \in E_0[B]$, one should be able to compute a generator of the group $\phi(\langle R\rangle)$.

To achieve this goal, the images $\phi(P)$, $\phi(Q)$ of P and Q are scaled by a random integer $\alpha \in \mathbb{Z}/B\mathbb{Z}^\times$. That is instead of revealing $\phi(P)$, $\phi(Q)$, one reveals $[\alpha]\phi(P)$, $[\alpha]\phi(Q)$. Note that since the degree of the secret isogeny ϕ is fixed, one can recover $\alpha^2 \deg\phi$ by applying Lemma 3, from which one derives $\alpha^2 \bmod B$. Taking a square root α_0 of α^2, one recovers $[\alpha\alpha_0^{-1}]\phi(P)$ and $[\alpha\alpha_0^{-1}]\phi(Q)$ where $(\alpha\alpha_0^{-1})^2 = 1 \mod B$. Hence one can sample α directly from $\mu_2(B)$ where

$$\mu_2(N) = \{x \in \mathbb{Z}/N\mathbb{Z} \mid x^2 = 1 \bmod N\}.$$

Note that for the scheme to be secure against the CD-MM-R attack, it is necessary that an attacker should not be able to recover the scalar α. The isogeny degrees are chosen such that there is an exponential number of square roots of 1 modulo B. This leads to the following variant of SIDH: M-SIDH.

Public parameter: Let λ be the security parameter and let $t = t(\lambda) \in \mathbb{N}$ be an integer depending on λ. Let $p = ABf - 1$ be a prime such that $A = \prod_{i=1}^{t} \ell_i$ and $B = \prod_{i=1}^{t} q_i$ are coprime integers, ℓ_i, q_i are distinct small primes, $A \approx B \approx \sqrt{p}$ and f is a small cofactor. Let E_0 be a supersingular curve defined over $\mathbb{F}_{p^2}$. Set $E_0[A] = \langle P_A, Q_A\rangle$ and $E_0[B] = \langle P_B, Q_B\rangle$. The public parameters are E_0, p, A, B, P_A, Q_A, P_B, Q_B.

Public key (Alice): Alice samples uniformly at random two integers α and a from $\mu_2(B)$ and $\mathbb{Z}/A\mathbb{Z}$ respectively. She computes the cyclic isogeny $\phi_A : E_0 \to E_A = E_0/\langle P_A + [a]Q_A\rangle$. Her public key is the tuple $\mathsf{pk}_A = (E_A, [\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B))$ and her secret key is $\mathsf{sk}_A = a$. The integer α is deleted.

Public key (Bob): Analogously, Bob samples uniformly at random two integers β and b from $\mu(A)$ and $\mathbb{Z}/B\mathbb{Z}$ respectively. His public key is $\mathsf{pk}_B = (E_B, [\beta]\phi_B(P_A), [\beta]\phi_B(Q_A))$ where $\phi_B : E_0 \to E_B = E_0/\langle P_B + [b]Q_B\rangle$ and his secret key is $\mathsf{sk}_B = b$. The integer β is deleted.

Shared key: Upon receiving Bob's public key (E_B, R_a, S_a), Alice checks that $e_A(R_a, S_a) = e_A(P_A, Q_A)^B$, if not she aborts. She computes the isogeny $\phi'_A : E_B \to E_{BA} = E_B/\langle R_a + [a]S_a\rangle$. Her shared key is $j(E_{BA})$. Similarly, upon receiving (E_A, R_b, S_b), Bob checks that $e_B(R_b, S_b) = e_B(P_B, Q_B)^A$, if not he

aborts. He computes the isogeny $\phi'_B : E_A \to E_{AB} = E_A/\langle R_b + [b]S_b\rangle$. His shared key is $j(E_{AB})$.

The problem underlying the security of M-SIDH is stated as follows.

Problem 4. Let $A = \ell_1 \cdots \ell_t$ and let $B = q_1 \cdots q_t$ be two smooth coprime integers, let f be a small cofactor such that $p = ABf - 1$ is a prime, with $A \approx B$. Let $E_0/\mathbb{F}_{p^2}$ be a supersingular elliptic curve such that $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (ABf)^2$, set $E_0[B] = \langle P, Q\rangle$. Let $\phi : E_0 \to E$ be a uniformly random A-isogeny and let α be a uniformly random element of $\mu_2(B)$.
Given $E_0, P, Q, E_A, P' = [\alpha]\phi(P), Q' = [\alpha]\phi(Q)$, compute ϕ.

It is immediate that Problem 4 is not hard for too small values of t. Recall that we want 1 to have an exponential number of square roots modulo A and modulo B. At first, one may be tempted to set $t = \lambda$ so that there are about 2^λ square roots of 1 modulo A and modulo B. But, as we will see in Sect. 4, this is not secure and t needs to be larger.

The main difference between Problem 4 and Problem 2 is that in Problem 4 torsion point images are only provided up to a scalar multiple (more precisely, a square root of unity). When trying to apply Robert's attack, the endomorphism Ψ appearing in this attack can no longer be evaluated exactly and its kernel can no longer be computed directly. The same holds for the attacks described in Castryck-Decru and Maino-Martindale papers.

3.2 Masked-Degree Variant

Rather than masking the torsion points as described in the previous section, we suggest a second countermeasure where one masks the degree of the secret isogeny.

Set the prime p to be of the form $p = ABf - 1$ where A and B are two smooth coprime integers, and f is a small cofactor. Alice will use cyclic isogenies of degree A' dividing A and Bob will use cyclic isogenies of degree B' dividing B. In an SIDH prime $A = \ell_A^{e_A}$ and $B = \ell_B^{e_B}$, hence A and B have only $e_A + 1$ and $e_B + 1$ divisors respectively. For this reason, one needs to move away from SIDH primes and use CSIDH-style primes with $A = \ell_1^{a_1} \cdots \ell_t^{a_t}$ and $B = q_1^{b_1} \cdots q_t^{b_t}$ where t, as well as the a_is and the b_is, depend on the security parameter λ.

To generate her public key, Alice samples a random degree A' (divisor of A) for her secret isogeny, samples a random point $R_A \in E_0[A']$, computes the A'-isogeny $\phi_A : E_0 \to E_A := E_0/\langle R_A\rangle$ and $\phi_A(P_B)$, $\phi_A(Q_B)$ where $E_0[B] = \langle P_B, Q_B\rangle$. But, by Lemma 3, any adversary can recover $A' = \deg\phi_A$. In order to avoid this, Alice also generates a uniformly random integer $\alpha \in \mathbb{Z}/B\mathbb{Z}^\times$ and outputs $(E_A, [\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B))$ as her public key. More precisely, Masked-degree SIDH (MD-SIDH) is as follows:

Public parameter: Let E_0 be a supersingular elliptic curve. Let $t = t(\lambda) \in \mathbb{N}$ be an integer depending[4] on λ. Let $A = \ell_1^{a_1} \cdots \ell_t^{a_t}$ and $B = q_1^{b_1} \cdots q_t^{b_t}$ be

[4] Note that we use the same notation $t = t(\lambda)$ for M-SIDH and MD-SIDH. It will always be clear from the context whether we are referring to M-SIDH or MD-SIDH.

two smooth coprime integers such that $p = ABf - 1$ is a prime (where f is a small cofactor). Set P_A and Q_A (resp. P_B and Q_B) be points generating $E_0[A] = \langle P_A, Q_A \rangle$ and $E_0[B] = \langle P_B, Q_B \rangle$.

Public key (Alice): Alice samples a uniformly random divisor A' of A and a random point $R_A \in E_0[A']$. Her secret key is $\mathsf{sk}_A = R_A$. She computes the isogeny $\phi_A : E_0 \to E_A := E_0/\langle R_A \rangle$ together with $\phi_A(P_B)$ and $\phi_A(Q_B)$. She samples a uniformly random integer[5] $\alpha \in \mathbb{Z}/B\mathbb{Z}^\times$ and her public key is $\mathsf{pk}_A = ([\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B))$.

Public key (Bob): Bob samples a uniformly random divisor B' of B and a random point $R_B \in E_0[B']$. His secret key is $\mathsf{sk}_B = R_B$. He computes the isogeny $\phi_B : E_0 \to E_B := E_0/\langle R_B \rangle$ together with $\phi_B(P_A)$ and $\phi_B(Q_A)$. He samples a uniformly random integer $\beta \in \mathbb{Z}/A\mathbb{Z}^\times$ and his public key is $\mathsf{pk}_B = ([\beta]\phi_B(P_A), [\beta]\phi_B(Q_A))$.

Shared key: From $[\beta]\phi_B(P_A)$ and $[\beta]\phi_B(Q_A)$, Alice recovers $\langle R'_A \rangle = \langle \phi_B(R_A) \rangle$. From $[\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B)$, Bob recovers $\langle R'_B \rangle = \langle \phi_A(R_B) \rangle$. Alice computes $E_{AB} := E_B/\langle R'_A \rangle$, and Bob computes $E_{BA} := E_A/\langle R'_B \rangle$. The value $j(E_{AB}) = j(E_{BA})$ is the shared key.

The problem underlying the security of MD-SIDH is stated as follows.

Problem 5. Let $A = \ell_1^{a_1} \cdots \ell_t^{a_t}$ and let $B = q_1^{b_1} \cdots q_t^{b_t}$ be two smooth coprime integers, let f be a small cofactor such that $p = ABf - 1$ is a prime, with $A \approx B$. Let $E_0/\mathbb{F}_{p^2}$ be a supersingular elliptic curve such that $\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (ABf)^2$, set $E_0[B] = \langle P, Q \rangle$. Let $A' = \ell_1^{a'_1} \cdots \ell_t^{a'_t}$ be a uniformly random divisor of A and and let α be a uniformly random element of $\mathbb{Z}/B\mathbb{Z}^\times$. Let $\phi : E_0 \to E_A$ be a uniformly random isogeny of degree A'. Given $E_0, P, Q, E_A, P' = [\alpha]\phi(P), Q' = [\alpha]\phi(Q)$, compute ϕ.

The MD-SIDH protocol can be seen as a generalization of the M-SIDH protocol, where the degree is no longer fixed and the torsion point hidden scalars α and β are no longer restricted to square roots of unity. From a security point of view, hiding the isogeny degrees might present the attacker with an additional obstacle in running the CD-MM-R attacks, since these degrees are explicitly used to construct attack parameters.

3.3 On the Effectiveness of the Countermeasures

We provide some arguments on why we believe the CD-MM-R attacks do not extend to the countermeasures.

The attacks embed the secret isogeny ϕ of degree B into a higher genus isogeny Ψ of degree $A = B + a$ where $a = A - B$ and A is the order of the torsion points. In M-SIDH, the degree of the secret isogeny is $\beta^2 B$ where β is a secret

[5] The integers α (for Alice) and β (for Bob) can be deleted immediately after key generation.

scalar and $\beta^2 \pmod A = 1$. Embedding this isogeny into a higher genus isogeny the same way would lead to an isogeny of degree

$$\beta^2 B + a = \beta^2 B + A - B = A + B(\beta^2 - 1) = A\left(1 + B\frac{\beta^2 - 1}{A}\right).$$

This degree is unknown to the attacker, because he does not know β. Now, since the degree of Ψ is $A(1+B\frac{\beta^2-1}{A})$, then $\Psi = \Psi_2 \circ \Psi_1$ where Ψ_1 has degree A and Ψ_2 has degree $1 + B\frac{\beta^2-1}{A}$. Evaluating Ψ on the A torsion (using the masked torsion point information available) gives you Ψ_1. The isogeny Ψ_2 whose unknown degree $1 + B\frac{\beta^2-1}{A}$ is larger than B and probably non smooth remains unknown. Hence, one cannot recover Ψ. Clearly, if $\beta = \pm 1$ then $\beta^2 = 1$ and Ψ_2 has degree 1 and Ψ is fully recovered.

A similar argument applies for MD-SIDH as well. In MD-SIDH, the degree of the secret isogeny is $\beta^2 B'$ where β is a secret scalar and B' is a random divisor of B (say $B = B'B_1$). Embedding this isogeny into a higher genus isogeny the same way would lead to an isogeny of degree

$$\beta^2 B' + a = \beta^2 B' + A - B'B_1 = A + B'(\beta^2 - B_1).$$

This degree is unknown to the attacker, because he does not know neither β nor B' (or B_1). Let $d = gcd(A, \beta^2 - B_1)$, then $\Psi = \Psi_2 \circ \Psi_1$ where Ψ_1 has degree d and Ψ_2 has degree $\frac{A}{d} + B'\frac{\beta^2-B_1}{d}$. Evaluating Ψ on the A torsion (using the masked torsion point information available) gives you Ψ_1. The isogeny Ψ_2 whose unknown degree $\frac{A}{d} + B'\frac{\beta^2-B_1}{d}$ is larger than B and probably non smooth remains unknown. Hence, one cannot recover Ψ.

Remark 6. One could try to use a different value a' instead of $a = A - B$ when embedding the secret isogeny into a higher genus isogeny. This would only make things more complicated. In fact the unknown degree of Ψ is $\beta^2 B + a'$ for M-SIDH and $\beta^2 B' + a'$ for MD-SIDH. As before, setting d to be the greatest common divisor of this degree and A, then $\Psi = \Psi_2 \circ \Psi_1$ where Ψ_1 has degree d. The isogeny Ψ_1, can be recovered, but Ψ_2 whose unknown degree is larger than A and B, and is probably non smooth cannot be efficiently recovered.

4 Security Analysis of the Masked Torsion Points Variant

Recall that the M-SIDH variant differs from SIDH in that parties send torsion point images only up to a constant α, which is a square root of unity.

We first describe a general attack that simply consists of guessing enough exact torsion point information to run the CD-MM-R attacks. This attack has exponential complexity in the number of prime divisors of A and B, and it works for any starting curve, even when the endomorphism ring is unknown.

We then describe a *polynomial time* attack when the initial curve is $j = 1728$, and we generalize it to starting curves with (known) small degree endomorphisms. We argue that it appears hard to extend this attack to the case where the endomorphism ring of the starting curve is unknown.

We end this section with a suggestion of parameters with respect to our analysis. In what follows we consider an isogeny of degree B, with images of torsion points of order A revealed up to a scalar α.

4.1 Guessing Enough Exact Torsion Point Information

Since Bob's isogeny has degree $B \approx A$, then we only need the exact images of the $\sqrt{B} \approx \sqrt{A}$ torsion points to run the CD-MM-R attacks. We are provided with the images of the A-torsion points where $A = \ell_1 \cdots \ell_t$.

Let $n \geq 1$ be the largest index such that $\sqrt{B} \leq \ell_n \cdots \ell_t$. Set $N = \ell_n \cdots \ell_t$. Then Bob's secret isogeny ϕ_B can be recovered from its action on the N-torsion points. From the action of $[\alpha] \circ \phi_B$ on the A-torsion points, one deduces the action of $[\alpha] \circ \phi_B$ on the N-torsion points. The only thing preventing us from applying the CD-MM-R attack is the unknown square root of unity α. Since N has $t - n + 1$ prime factors, then there are at most 2^{t-n+1} square roots of unity modulo N. One can hence try all these square roots of unity till one gets one for which the CD-MM-R attack is successful.

The overall complexity of this attack is $\tilde{O}(2^{t-n+1})$ using a classical computer. Since $N \approx \sqrt{A}$ and N is made up of the largest prime factors of A, then we must have $t/2 < n$, which implies $t - n + 1 \leq t/2$. This attack is summarized in Algorithm 1. We deduce Theorem 7.

Algorithm 1. Attack by using less torsion point information

Require: E_0, P_A, Q_A, E_B, $P' = [\beta]\phi_B(P_A)$, $Q' = [\beta]\phi_B(Q_A)$ from an M-SIDH instance.
Ensure: ϕ_B.
1: Set $N = \ell_n \cdots \ell_t$ where $n \geq 1$ be the largest index such that $\sqrt{B} \leq \ell_n \cdots \ell_t$;
2: Compute $P_1 = [\frac{A}{N}]P'$ and $Q_1 = [\frac{A}{N}]Q'$;
3: **for** each square root γ of unity modulo N **do**
4: try to run the CD-MM-R attack to recover ϕ_B from E_0, P_A, Q_A, E_B, $[\gamma^{-1}]P_1$,
5: $Q' = [\gamma^{-1}]Q_1$;
6: **if** The CD-MM-R attack is successful **then**
7: **return** ϕ_B.

Theorem 7. *Algorithm 1 is correct and runs in time* $\tilde{O}(2^{t-n+1})$ *using a classical computer with* $t - n + 1 \leq t/2$.

The above discussion only considers classical security. When we use a quantum computer, the complexity of the attack can be improved because Grover algorithm [22] allows us to find the correct γ in Algorithm 1 in time $\tilde{O}(2^{(t-n+1)/2})$. We deduce Theorem 8.

Theorem 8. *Algorithm 1 is correct and runs in time* $\tilde{O}(2^{(t-n+1)/2})$ *using a quantum computer with* $t - n + 1 \leq t/2$.

Remark 9. From Theorem 7 and 8, the value t should be greater than or equal to 2λ for AES-λ security (i.e. λ bits of classical security and $\lambda/2$ bits of quantum security).

4.2 Polynomial Time Attack When E_0 Has j-invariant $= 1728$

Castryck-Decru, Maino-Martindale and Robert's new SIDH attacks seem to require exact knowledge of torsion point images, motivating the M-SIDH variant. On the other hand, older torsion point attacks only required these images up to a constant [2,19], though of course they also required A much larger than B. This suggests looking for an improved attack combining the best of both worlds.

Let $\iota \in \mathrm{End}(E_0) : (x, y) \to (-x, iy)$ be a non-trivial automorphism of E_0 and let

$$\psi := \phi \circ \iota \circ \hat{\phi}$$

be the "lollipop endomorphism" constructed in Petit's attack and variants (see Sect. 2.2). As the images of torsion points through ϕ are provided up to a scalar α where $\alpha^2 = 1 \mod A$, then we have that

$$[\alpha]\phi \circ \iota \circ \widehat{[\alpha]\phi} = [\alpha^2] \circ \phi \circ \iota \circ \hat{\phi} = [\alpha^2] \circ \psi.$$

Hence, from the action of $[\alpha]\phi$ on the A-torsion, one can recover the action of $[\alpha^2] \circ \psi$ on the A-torsion. Since $\alpha^2 = 1 \mod A$, then the images of $A-$torsion points through ψ are exact. Moreover as ψ has degree $B^2 \approx A^2$ and images of torsion points of order A are known, we can apply Robert's attack to ψ instead of ϕ! After recovering ψ, one can recover φ efficiently.

Remark 10. One can recover ψ from ϕ as in [31]: compute $G := \ker\phi \cap E'[B]$ and extract the largest cyclic subgroup in G. Generically this is a large subgroup of $\ker\phi$, and the remaining part of $\ker\phi$ is simply guessed. Note that the powersmooth case (as in M-SIDH) is considered a worst case in [31], but even in this "worst case situation" the cost is shown to be polynomial time in expectation (for randomly chosen ϕ).

4.3 Generalization to Other Starting Curves

More generally, given the endomorphism ring of the starting curve, one can apply the LLL algorithm to compute a short non scalar endomorphism in it. One can then replace the endomorphism ι of degree 1 in the previous attack by another higher degree endomorphism θ.

As $\deg\psi = B^2 \deg\theta$ and Robert's attack requires $\deg\psi \leq A^2$, this strategy would require to first guess the action of ϕ on a torsion A' subgroup with $A' \geq \sqrt{\deg\theta}$, up to a scalar. This involves guessing the images of two cyclic A'-torsion subgroups, hence it requires $O(A'^2) \approx \deg\theta$ attempts.

The attack will be relevant for any starting curve with a small non-trivial endomorphism. For generic curves, we will have $\deg\theta \approx \sqrt{p}$, so the attack will not provide any improvement over a trivial guessing strategy.

4.4 Generalization Attempt to Unknown Endomorphism Rings

When the endomorphism ring of the starting curve is unknown to the attacker, one may hope to generalize the previous attack in the same way as previous torsion point attacks were generalized by the Castryck-Decru's, Maino-Martindale's and Robert's attacks: by embedding the isogeny problem into an isogeny problem of higher dimension.

In particular, one could try to achieve this is by considering the genus 3 product $A := E_0 \times E_0 \times E'$, and an endomorphism

$$\psi : (P_1, P_2, P_3) \to (P_1, P_2, \phi_2\theta_{12}\hat{\phi}_1(P_3))$$

where $\phi_i : E_0^{(i)} \to E'$ for $i = 1, 2$ are two copies of the secret isogeny ϕ, and $\theta_{12} : E_0^{(1)} \to E_0^{(2)}$ is a small degree isogeny from one copy of E_0 to the other one. However, a closer look at this attempt reveals that $\phi_2\theta_{12}\hat{\phi}_1$ is just a scalar multiplication on E', and it can therefore not help to recover ϕ. Other similar strategies we tried led to the same issue.

5 Security Analysis of the Masked Degree Variant

In this section, we discuss the security of our second countermeasure MD-SIDH. First, we prove that the square-free part of the degree of the secret isogeny can be recovered efficiently; this is done in Subsect. 5.1 below (together with 5.2 for a technical lemma). In Subsect. 5.3 we show how to reduce any instance of Masked-degree SIDH into an instance of Masked torsion points SIDH when the square-free part of the secret isogeny is known. The latter implies that all the attacks presented in Sect. 4 can be extended to MD-SIDH through this reduction. Taking this into account, we suggest parameters for MD-SIDH in Subsect. 7.3.

Recall that in the MD-SIDH setting, $A = \ell_1^{a_1} \cdots \ell_t^{a_t}$, $B = q_1^{b_1} \cdots q_t^{b_t}$ and $p = ABf - 1$ where f is a small cofactor. We are targeting Bob's secret isogeny $\phi : E_0 \to E_B$ whose degree $B' = q_1^{b'_1} \cdots q_t^{b'_t}$ is an unknown divisor of B, and we are provided with $E_0, P, Q, E_B, P' = [\beta]\phi(P), Q' = [\beta]\phi(Q)$ where $E_0[A] = \langle P, Q\rangle$ and $\beta \in \mathbb{Z}/A\mathbb{Z}^\times$ is unknown.

5.1 Recovering the Degree up to Squares

From Lemma 3, one can assume that $\beta^2 B' \bmod A$ is known. In this section we show how to deduce a small set of candidates for the square-free part of B'.

Let $\mathcal{D}(B)$ be the set of positive divisors of B. Given $\underline{b}' = (b'_1, \cdots, b'_t) \in \mathbb{Z}^t$, we write $B(\underline{b}') = q_1^{b'_1} \cdots q_t^{b'_t}$, and similarly if $B' = B(\underline{b}')$, we write $\underline{b}' = b(B')$. These maps restrict to a one-to-one correspondence between $\prod_{i=1}^t \mathbb{Z}/(b_i + 1)\mathbb{Z}$ and $\mathcal{D}(B)$. For simplicity, we suppose that the ℓ_is and the q_is are odd primes[6].

[6] The case where $\ell_i = 2$ in general does not fit our definition of χ since there are more than two square roots of 1 modulo 2^r for $r > 2$. Nevertheless, if the power of 2 diving A or B is at least 4, then the security of the scheme is not affected.

Let $\chi_{\ell_i^{a_i}}$ be the natural surjection $\chi_{\ell_i^{a_i}} : (\mathbb{Z}/\ell_i^{a_i}\mathbb{Z})^\times \to (\mathbb{Z}/\ell_i^{a_i}\mathbb{Z})^\times / ((\mathbb{Z}/\ell_i^{a_i}\mathbb{Z})^\times)^2 \cong \{-1, 1\}$. Consider the map

$$\begin{aligned} \Phi : \mathbb{Z}^t &\longrightarrow \{-1, 1\}^t \\ \underline{b}' &\longmapsto \left(\chi_{\ell_1^{a_1}}(B(\underline{b}')), \ldots, \chi_{\ell_t^{a_t}}(B(\underline{b}'))\right), \end{aligned}$$

where we regard $B(\underline{b}')$ as an element in $(\mathbb{Z}/\ell_i^{b_i}\mathbb{Z})^\times$. Clearly, Φ is a group morphism and $(2\mathbb{Z})^t \subset \ker \Phi$. This implies that the following group homomorphism is well-defined:

$$\begin{aligned} \overline{\Phi} : (\mathbb{Z}/2\mathbb{Z})^t &\longrightarrow \mathrm{Im}(\Phi) \subset \{-1, 1\}^t \\ \underline{b}' &\longmapsto \Phi(\underline{b}'). \end{aligned}$$

Since the cardinality of the domain of the group morphism $\overline{\Phi}$ is 2^t, then $\#\ker\overline{\Phi} = 2^{t_\Phi}$ for some $0 \leq t_\Phi \leq t$. This implies that when given $\Phi(b(B'))$, we have $\#\overline{\Phi}^{-1}(\Phi(b(B'))) = 2^{t_\Phi}$. In other words, giving $\Phi(b(B'))$ is the same as giving $t - t_\Phi$ bits of information about $b(B')$ mod 2. Furthermore, when $t_\Phi = 0$, that is when $\overline{\Phi}$ is an isomorphism, then $\Phi(\underline{b}')$ uniquely determines $b(B')$ mod 2. Note that for any representative $\underline{b}'$ in the class $b(B') + 2\mathbf{Z}^t$, the integers $B(\underline{b}')$ and B' have the same square-free factor.

Lemma 11. *Consider the notations above. Let B'_1 be the square-free part of B'. Then given E_0, P, Q, E_B, P' and Q', there exists a probabilistic polynomial time algorithm that reduces the search space for B'_1 to a set of order 2^{t_Φ} where $2^{t_\Phi} = \#\ker\overline{\Phi}$.*

Proof. From P, Q, E_B, P', and Q', use Lemma 3 to recover $d = \beta^2 B' \bmod A$. Compute

$$\Phi(b(d)) = \left(\chi_{\ell_1^{a_1}}(\beta^2 B'), \cdots, \chi_{\ell_t^{a_t}}(\beta^2 B')\right) = \left(\chi_{\ell_1^{a_1}}(B'), \cdots, \chi_{\ell_t^{a_t}}(B')\right) = \Phi(b(B')).$$

Compute $\ker\overline{\Phi}$ and a preimage $\underline{b_0}$ of $\Phi(b(d))$ with respect to $\overline{\Phi}$. Return the set $\{B(\underline{b}) \mid \underline{b} \in \underline{b_0} + \ker\overline{\Phi}\}$ of square-free integers.

Clearly, all the computational steps described above run in polynomial time. The correctness follows from the properties of the morphism Φ and $\overline{\Phi}$ discussed before the lemma. □

In the next subsection, we show that t_Φ is expected to be small for most parameters, and the squarefree part of B' can therefore be guessed with a high probability.

5.2 On the Value of t_Φ

In this section we estimate t_Φ. We start by observing the following.

Lemma 12. *Let t be an integer and let M be a random $t \times t$ matrix over $\mathbb{F}_2$. Then as t tends towards ∞, we have have $t - 2 \leq rank(M)$ with probability* 0.9947145498.

Proof. Let $p_t(k)$ be the probability that a uniformly random $t \times t$-matrix over $\mathbb{F}_2$ has rank $t-k$. Then, from [25, p.33] and [11, Theorem 1], it holds that

$$\pi(k) := \lim_{t\to\infty} p_t(k) = \begin{cases} \prod_{i=1}^{\infty}\left(1-\frac{1}{2^i}\right), & (k=0) \\ \frac{\prod_{i=k+1}^{\infty}(1-(1/2^i))}{\prod_{i=1}^{k}(1-(1/2^i))}\frac{1}{2^{k^2}}, & (k\geq 1). \end{cases}$$

From [11, Table 1], we have $\pi(0)$ is about 0.2887880951, $\pi(1)$ is about 0.5775761902, $\pi(2)$ is about 0.1283502645, and $\pi(k)$'s for $k \geq 3$ are less than 0.0052387863. Therefore,

$$\mathsf{Pr}(k \leq 2) = \pi(0) + \pi(1) + \pi(2) \approx 0.9947145498.$$

□

Consider the matrix M^* of the morphism $(1-\bar{\Phi})/2$ (operations are done component wise). Lemma 12 applies to random matrices and t needs to be somehow large. In practice, t is relatively small, $t \approx O(\lambda)$ where λ is the security parameter, and, A and B are system parameters, which could in theory be chosen to maximize t_ϕ. However, for the sake of the scheme's practicality, the integers A and B need to be as small as possible. Also, in order to not weaken one of the participants, A and B need to satisfy $A \approx B$. With these constraints, we do not expect to have $2 < t_\Phi$. Intuitively, if $2 < t_\Phi$, then there are t_Φ square-free integers that are all quadratic residues modulo all the t prime power divisors of A. Given a random square-free integer N, N is a quadratic residue modulo a given prime power with probability $\frac{1}{2}$, hence it is a quadratic residue modulo t "independent" distinct prime powers with probability roughly $\frac{1}{2^t}$. Since $t \approx O(\lambda)$, then $\frac{1}{2^t}$ is negligible. Hence the probability that there exists many such integers N rapidly decreases below $\frac{1}{2^\lambda}$. For example, given t, let $\ell_1, q_1, \ell_2, q_2, \ldots, \ell_t, q_t$ be the smallest $2t$ odd primes. For $t = 64, 96, 128, 192, 286, 420, 426, 566, 637, 856$, we obtained $t_\Phi = 1, 0, 1, 2, 0, 0, 1, 0, 1, 0$ respectively.

In conclusion, we believe that it is computationally hard in practice to come up with integers A and B such that $2 \ll t_\Phi$. Also, if ever such integers were computed, A and B would be too large, which would lead to an impractical scheme.

5.3 Reduction to the M-SIDH Variant

We now show how to reduce a MD-SIDH instance to an M-SIDH instance.

Recall that in the MD-SIDH case (Problem 5), we are given E_0, P, Q, $E_B, P' = [\beta]\phi(P), Q' = [\beta]\phi(Q)$, where $E_0[A] = \langle P, Q\rangle$, ϕ is a random isogeny of degree B' with B' being a random divisor of B, and β is a random integer coprime to A; and we are asked to recover ϕ.

Following Subsects. 5.1 and 5.2, we assume that the square-free part of the degree of the secret isogeny is known. Let B'_1 be the square-free part of B'. Let

B_0 be the largest divisor of B which is equal to B' up to squares, and let β_0 be the divisor of B such that $B_0 = \beta_0^2 B'$. Since B is a smooth integer and we know B'_1, we can compute B_0.

Let $\phi_0 = [\beta_0] \circ \phi$. We then know $\deg \phi_0 = \beta_0^2 \deg \phi = \beta_0^2 B' = B_0 \leq B$. Moreover, we have

$$\begin{cases} P' = [\beta]\phi(P) = [(\beta\beta_0^{-1}) \cdot \beta_0]\phi(P) = [\beta\beta_0^{-1}]\phi_0(P) \\ Q' = [\beta]\phi(Q) = [(\beta\beta_0^{-1}) \cdot \beta_0]\phi(Q) = [\beta\beta_0^{-1}]\phi_0(Q) \end{cases}$$

From Lemma 3, we can recover $\beta^2 B' \bmod A$, and compute

$$\beta_1^2 = \beta_0^2 B' \cdot (\beta^2 B')^{-1} \bmod A = (\beta_0 \cdot \beta^{-1})^2 \bmod A.$$

We sample a random square root β'_1 of $\beta_1^2 \mod A$, namely $\beta'_1 = \mu\beta_1$ where μ is some square root of unity modulo A. We compute

$$\begin{cases} [\beta'_1]P' = [\mu \cdot \beta_1]P' = [\mu \cdot \beta_0 \cdot \beta^{-1} \cdot \beta]\phi(P) = [\mu \cdot \beta_0]\phi(P) = [\mu]\phi_0(P) \\ [\beta'_1]Q' = [\mu \cdot \beta_1]P' = [\mu \cdot \beta_0 \cdot \beta^{-1} \cdot \beta]\phi(Q) = [\mu \cdot \beta_0]\phi(Q) = [\mu]\phi_0(Q) \end{cases}$$

From here, one solves for ϕ_0 where E_0, P, Q, E_B, $[\mu]\phi_0(P)$, $[\mu]\phi_0(Q)$ and $\deg \phi_0 = \beta_0^2 B'$ are provided. This is in fact an M-SIDH instance, with the only difference that the secret isogeny is not cyclic. This is not a problem since the higher genus torsion points attack has no restriction on the type of isogeny (cyclic or not) in play.

5.4 Reduction Impact: Porting M-SIDH Attacks to MD-SIDH

In this section we revisit the attacks of Sect. 4 and check that they still apply when the secret isogeny is not cyclic. Recall that the isogeny to recover here is $\phi_0 = [\beta_0] \circ \phi$ where β_0 is an unknown integer such that the degree $B_0 = \beta_0^2 B'$ of ϕ_0 divides B. We are provided with the $B_0 = \deg \phi_0$ and the action of ϕ_0 on the A torsion group up to a scalar (a root of unity modulo A).

We start with Robert's attack (see Sect. 2.2), and observe that neither the definition of the isogeny Ψ nor the way ϕ is deduced from this isogeny rely on ϕ being a cyclic isogeny. The attack of Sect. 4.1 simply guesses the exact torsion point images on the minimal amount of torsion point information needed for Robert's attack. As before, allowing for non cyclic isogenies does not affect Robert's attack.

Regarding the attack of Sect. 4.2, one first applies Robert's attack on $\psi = \phi_0 \circ \iota \circ \hat{\phi}_0 = [\beta_0^2] \circ \phi \circ \iota \circ \phi$, then one deduces the isogeny ϕ_0 using [31, §4.3]. The first part is again an application of Robert's attack to a non cyclic isogeny. The second part requires some clarification.

When $\phi_0 = \phi$ is cyclic ($\beta_0 = 1$), in [31, §4.3], the attacker computes $G := \ker \psi \cap E_B[B_0]$, which clearly contains $\ker \phi$, and is in general isomorphic to $\mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N'\mathbb{Z}$ for some $N'|N|B_0$ such that $NN' = B_0$. When $N' = 1$ we have $G = \ker \phi$. For a cyclic isogeny ϕ, we have $N' > 1$ exactly when ι leaves either $\ker \phi \cap E_0[N']$ or $\ker \hat{\phi} \cap E_B[N']$ invariant: this leaves at most 2^r candidates for

$\ker \phi$, where r is the number of prime factors in N'. But, since ϕ is uniformly random (because it is the secret isogeny), then N' is relatively small and hence has very few prime factors.

Let us return to the case where $\phi_0 = [\beta_0] \circ \phi$ with ϕ being cyclic and $\beta_0 > 1$ is unknown, and let us assume that $\psi = \phi_0 \circ \iota \circ \hat{\phi}_0 = [\beta_0^2] \circ \phi \circ \iota \circ \phi$ has been recovered. After evaluating ψ on the B_0 torsion, we will get a group isomorphic to $\mathbb{Z}/C\mathbb{Z} \times \mathbb{Z}/C'\mathbb{Z}$ where $C'|C|B_0$. Since ψ is divisible by β_0^2, the β_0^2-torsion is killed by ψ and the group $\mathbb{Z}/C\mathbb{Z} \times \mathbb{Z}/C'\mathbb{Z}$ is the group one would have got if one was evaluating $\phi \circ \iota \circ \phi$ on the $B' = B_0/\beta_0^2$ torsion. Hence $CC' = B' = \deg \phi$ as discussed in the previous paragraph and $\beta_0 = \sqrt{B_0/B'}$. One then recovers ϕ as in the previous paragraph.

6 Adaptive Attacks

In this section, we show that M-SIDH and MD-SIDH, as SIDH, are vulnerable to adaptive attacks. We also discuss the use of B-SIDH primes in M-SIDH and future work. We first discuss the Fouotsa-Petit attack, then the GPST attack.

6.1 Fouotsa-Petit Adaptive Attack

Fouotsa-Petit [19] adaptive attack consists in actively transforming a balanced SIDH instance ($A \approx B$) into an imbalanced one ($B < A^* = NA$ where $N \approx p$), then running Petit's torsion point attacks [31,32] on the imbalanced SIDH (where the secret isogeny has degree B and the torsion points have order $A^* = NA$) to recover the secret isogeny.

In [19, Section 3.2], the authors show that Petit's torsion point attacks can be run even when the torsion point images are scaled with an unknown scalar. Petit's attacks also apply to non cyclic isogenies. In fact, to recover an isogeny $\phi : E \longrightarrow E'$ from its action on large enough torsion points, Petit's attack (see Sect. 2.2) uses the torsion point information and a suitable endomorphism θ of E to compute the endomorphism $\psi = \phi \circ \theta \circ \phi$ of E'; then the techniques of [31, §4.3] (also see Sect. 5.4) are used to recover ϕ. As before, ϕ not being cyclic does not impact the first step where one recovers ψ. There are some subtleties when trying to recover ϕ from ψ when ϕ is non cyclic, but they were already covered in Sect. 5.4. However, Petit's attacks do require knowledge of the degree of the secret isogeny.

The generalization of the Fouotsa-Petit [19] adaptive attack to M-SIDH is therefore straightforward. For MD-SIDH, one can use the techniques from Sect. 5 to reduce the MD-SIDH instance to an M-SIDH instance, and then run the Fouotsa-Petit attack on the M-SIDH instance.

6.2 GPST Adaptive Attack

Recall that in the GPST attack on SIDH, Bob (the honest party) has a static secret key/public key pair (b, $(E_B, \phi_B(P_A), \phi_B(Q_A))$). Alice (the dishonest party)

maliciously generates public keys (E_A, R, S) with modified torsion points images, and repeatedly runs the key exchange with Alice using these malicious public keys. The attack assumes that Alice is provided with an oracle $O(E_A, R, S, E')$ that outputs 1 if E' is the shared secret computed by the honest Bob when using (E_A, R, S) as Alice's public key, and 0 otherwise. Then the GPST adaptive attack recovers the secret b with only $\log b$ queries to the oracle O. The attack provides the points (R_i, S_i) to be used at each query. We refer to [20] for details about the GPST adaptive attack.

One thing to notice here is that

$$\begin{aligned} O(E_A, R, S, E') = 1 &\iff E_A/\langle R + [b]S\rangle = E_A/\langle \phi_A(P_B) + [b]\phi_A(Q_B)\rangle \\ &\iff \langle R + [b]S\rangle = \langle \phi_A(P_B) + [b]\phi_A(Q_B)\rangle \end{aligned}$$

where the second equivalence holds except with neglible probability. Hence, assuming E_A is fixed, we can see O as an oracle that when given R, S, outputs 1 if $\langle R+[b]S\rangle = \langle \phi_A(P_B)+[b]\phi_A(Q_B)\rangle$, and 0 if not. Note that the malicious points R and S are obtained by doing a linear combination of $\phi_A(P_B)$ and $\phi_A(Q_B)$, say $R = [e_1]\phi_A(P_B) + [e_2]\phi_A(Q_B)$ and $S = [f_1]\phi_A(P_B) + [f_2]\phi_A(Q_B)$.

When it comes to M-SIDH, the image points are scaled with a secret invertible scalar β. But, as the scalar is invertible and everything is linear, the attack can proceed in the same way.

For MD-SIDH, the degree B' of Bob's secret isogeny is an unknown divisor of B. The torsion points R and S are first scaled by the secret integer $B_1 = \frac{B}{B'}$ before being used by Bob; that is Bob computes the isogeny $E_A \to E_A/\langle [B_1](R+[b]S)\rangle$. Hence our new oracle here acts as follows: when given R and S, it outputs 1 if $\langle [B_1](R + [b]S)\rangle = \langle [B_1](\phi_A(P_B) + [b]\phi_A(Q_B))\rangle$, and 0 otherwise.

Here, blindly applying the GPST adaptive attack would not work, as the attacker first needs to recover the degree B' of the secret isogeny or equivalently the integer $B_1 = \frac{B}{B'}$. Moreover unlike for the Fouotsa-Petit attack, one cannot simply apply our reduction from Sect. 4 to recover the degree and then apply GPST attack, because the GPST attack assumes a cyclic secret isogeny.

To recover the integer B_1, we instead use the above oracle. Let q^e be a prime power divisor of B. We would like to recover the largest integer $e' \leq e$ such that $q^{e'}$ divides B'. We repeatedly query the oracle with the points

$$R_i = \phi_A(P_B) + \left[\frac{B}{q^i}\right]\phi_A(Q_B), \quad S_i = \phi_A(Q_B), \quad 1 \leq i \leq e.$$

We have the following lemma.

Lemma 13. *With the notations as above, we have*

$$O\left(\phi_B(P_A) + \left[\frac{B}{q^i}\right]\phi_A(Q_B), \phi_A(Q_B)\right) = 1$$

if and only if q^i divides B_1.

Proof. Set $R_i = \phi_A(P_B) + \left[\frac{B}{q^i}\right]\phi_A(Q_B)$ and $S_i = \phi_A(Q_B)$. We have

$$\begin{aligned}\langle [B_1](R_i + [b]S_i)\rangle &= \langle [B_1](\phi_A(P_B) + \left[\tfrac{B}{q^i}\right]\phi_A(Q_B) + [b]\phi_A(Q_B))\rangle \\ &= \langle [B_1]\left(\phi_A(P_B) + [b]\phi_A(Q_B)\right) + [\tfrac{B}{q^i}][B_1]\phi_A(Q_B)\rangle.\end{aligned}$$

Clearly, the points $[B_1](\phi_A(P_B) + [b]\phi_A(Q_B))$ and $[B_1]\phi_A(Q_B)$ have order B' and are linearly independent. Hence

$$\langle [B_1]\left(\phi_A(P_B) + [b]\phi_A(Q_B)\right)\rangle = \langle [B_1]\left(\phi_A(P_B) + [b]\phi_A(Q_B)\right) + \left[\frac{B}{q^i}\right][B_1]\phi_A(Q_B)\rangle$$

if and only if $\frac{B}{q^i} = 0 \mod B'$, that is if and only if q^i divides B_1. The Lemma then follows from the definition of the oracle O. □

Using Lemma 13, one recovers each prime power divisor $q_i^{e'_i}$ of $B_1 = \frac{B}{B'}$ with at most e_i queries, where $B = q_1^{e_1} \cdots q_t^{e_t}$, $i = 1, \ldots, t$. The total maximum number of queries to recover the secret degree B' is $\sum_{i=1}^{t} e_i$. Once the degree is recovered, one then runs the usual GPST attack.

7 Parameter Selection and Efficiency

In this section, we discuss the choice of the starting curve E_0, and we use the analysis from the previous sections to infer parameter selections for both M-SIDH and MD-SIDH. We conclude that the M-SIDH variant is always more secure than the MD-SIDH variant at comparable parameter sizes, and discuss its efficiency.

7.1 Choosing the Starting Curve E_0

From the attack in Sect. 4.2 and its generalization to MD-SIDH in Sect. 5.4, an elliptic curve with a short-degree endomorphism (e.g., the curve of j-invariant 1728) should not be used for a starting curve in either scheme. Therefore the setup algorithm needs to generate E_0 as a curve with no short-degree endomorphism. There are three possibilities here:

1. the endomorphism ring of the curve is public,
2. the endomorphism ring of the curve is not public, but known by one party (either Alice or Bob);
3. the endomorphism ring is unknown to everyone.

The advantage of the first possibility is that since the endomorphism ring of E_0 is public, everyone can verify that E_0 does not have small endomorphisms by determining the norm of the shortest element $\mathrm{End}(E_0)$ (this is a dimension 4 lattice, so computing the shortest element is easy). One can use Bröker's algorithm [6] to generate E_0, or obtain E_0 by performing a random walk from a supersingular curve computed using Bröker's algorithm. The first option is not secure since the supersingular curves generated using Bröker's algorithm have

small endomorphisms. In the second option, the party that generates the curve could backdoor it. In fact, they could generate a weak curve in the sense of [32]. Weak curves are curves for which Petit's torsion point attack has the best efficiency for a given set of parameters.

In the second scenario, one of the participants generates the curve and does not reveal its endomorphism ring. This party could hence potentially cheat and use a curve with small endomorphisms or a weak curve, then use it to attack the other party. This is not acceptable for a key exchange protocol. Nevertheless, in the setting of a SIKE-type key encapsulation mechanism or public key encryption scheme, we can let the key generation algorithm generate the starting curve and publish it together with their public key: indeed using a weak curve here would only make their own secret key weaker.

Regarding the third scenario, one should note that generating a supersingular curve with unknown endomorphism ring is a hard problem [4,28]. Instead, one can rely on a trusted third party (possibly simulated by a multiparty protocol [1]) to generate a truly random supersingular E_0 curve by performing a long random walk for a known supersingular curve, and forgets (deletes) the walk they used. Then the obtained curve could be used as starting curve for the schemes suggested in this scheme.

In conclusion, restricting E_0 to curves which do not have small endomorphisms is sufficient when instantiating M-SIDH and MD-SIDH. Nevertheless, one would need to trust the party generating the curve since they could backdoor it (we could not find a method to generate curves with no small endomorphisms in the literature). Since we would need to trust them anyway, it is better to just ask this party to generate a curve with unknown endomorphism ring. This can also be done using the MPC techniques described in [1].

7.2 Parameter Selection for M-SIDH

Recall that the M-SIDH primes are of the form $p = ABf - 1$ where $A = \ell_1 \cdots \ell_t$ and $B = q_1 \cdots q_t$ are coprime integedoℓ_i, q_i are distinct small primes, $A \approx B \approx \sqrt{p}$ and f is a small cofactor. Let λ be the security parameter. From Subsect. 4.1, we need $t - n + 1 \geq \lambda$ for classical security and $t - n + 1 \geq 2\lambda$ for quantum security, where n is the largest integer satisfying $\sqrt{B} \leq \ell_n \cdots \ell_t$, where λ is a security parameter.

We now explain how to generate the public parameters of M-SIDH for AES-λ security (i.e., classical λ bits security and quantum $\lambda/2$ bits security). Given λ, we sample the $2t$ smallest primes for $t \geq 2\lambda$, we partition them into two sets of equal size, we use the first set to get A and we use the second to get B, such that $A \approx B$. We then check the value $t - n + 1$ described in Subsect. 4.1. If $\lambda < t - n + 1$, we restart with a larger t. If $\lambda \geq t - n + 1$, find a cofactor f such that $p = ABf - 1$ is prime.

For AES-128 (NIST level 1), AES-192 (NIST level 3) and AES-256 (NIST level 5) security levels, Table 1 presents the key sizes, including secret key, public key and compressed public key. The suggested primes for M-SIDH are

$$p_{128} = 2^2 \cdot \ell_1 \cdots \ell_{571} \cdot 10 - 1,$$

$$p_{192} = 2^2 \cdot \ell_1 \cdots \ell_{851} \cdot 207 - 1$$

and

$$p_{256} = 2^2 \cdot \ell_1 \cdots \ell_{1131} \cdot 13 - 1$$

respectively; where ℓ_i is the ith odd prime. Alice uses $A = 2^2 \cdot \ell_2 \cdot \ell_4 \cdots \ell_{t-2}$ and Bob uses $B = \ell_1 \cdot \ell_3 \cdots \ell_{t-1}$.

Table 1. Suggested parameters for 128, 192 and 256 bits of security.

AES	NIST	p (in bits)	secret key	public key	compressed pk
128	level 1	5911	$\approx$ 369 bytes	4434 bytes	$\approx$ 2585 bytes
192	level 3	9382	$\approx$ 586 bytes	7037 bytes	$\approx$ 4103 bytes
256	level 5	13000	$\approx$ 812 bytes	9750 bytes	$\approx$ 5687 bytes

7.3 Parameter Selection for MD-SIDH

We showed in previous sections that MD-SIDH can be broken by the same attacks as M-SIDH. Therefore, $t - n + 1$ must be greater than or equal to λ for AES-λ security, where n is the largest integer such that there is a subset $S \subset \{1, \ldots, t\}$ satisfying $\sqrt{B} \leq \prod_{i \in S} \ell_i^{a_i}$ and $n = t + 1 - \#S$. Moreover, to mask the degree of the secret isogeny, the size of the space of degrees needs to be $2^{\lambda+t}$ since the Weil pairing will reduce it by a factor 2^t.

Given λ, we sample the $2t$ smallest primes for $t \geq \lambda$, we set $a_1 = \cdots = a_\lambda = b_1 = \cdots = b_\lambda = 3$ and the other exponents are 1, and we partition them into two sets of equal size. We use the first set to get A and the second to get B, such that $A \approx B$. We check the value $t - n + 1$ described above. If $\lambda < t - n + 1$, we restart with a larger t. If $\lambda \geq t - n + 1$, we find a cofactor f such that $p = ABf - 1$ is prime.

For AES-128 (NIST level 1), AES-192 (NIST level 3) and AES-256 (NIST level 5) security levels, Table 2 presents the key sizes: secret key, public key and compressed public key. The suggested primes for M-SIDH are

$$p_{128} = 2^3 \cdot \ell_1^3 \cdots \ell_{255}^3 \ell_{256} \cdots \ell_{839} \cdot 537 - 1,$$

$$p_{192} = 2^3 \cdot \ell_1^3 \cdots \ell_{383}^3 \ell_{384} \cdots \ell_{1273} \cdot 131 - 1$$

and

$$p_{256} = 2^3 \cdot \ell_1^3 \cdots \ell_{511}^3 \ell_{512} \cdots \ell_{1711} \cdot 1485 - 1$$

respectively; where ℓ_i is the ith odd prime. Alice uses $A = 2^3 \cdot \ell_2^3 \cdots \ell_{\lambda-2}^3 \ell_\lambda \cdots \ell_{t-2}$ and Bob uses $B = \ell_1^3 \cdots \ell_{\lambda-1}^3 \ell_{\lambda+1} \cdots \ell_{t-1}$.

Table 2. Suggested parameters for 128, 192 and 256 bits of security.

AES	NIST	p (in bits)	secret key	public key	compressed pk
128	level 1	13810	$\approx$ 863 bytes	10358 bytes	$\approx$ 6040 bytes
192	level 3	22291	$\approx$ 1393 bytes	16719 bytes	$\approx$ 9751 bytes
256	level 5	31226	$\approx$ 1951 bytes	23420 bytes	$\approx$ 13660 bytes

7.4 Preliminary Efficiency Analysis

From the two subsections above, it is clear that the M-SIDH variant is more secure than the MD-SIDH variant for comparable parameter sizes.

Compressed public key sizes for M-SIDH have 2585, 4103 and 5687 bytes at security levels 128, 192 and 256. This is roughly 6.8, 7.3 and 7.8 bigger than previously suggested SIKE keys for the same security levels. Asymptotically, Keys scale quasi-linearly in the security parameter, whereas SIKE keys scaled linearly.

Computations required in M-SIDH are similar to those required in SIDH, with additional (comparably negligible) scalar multiplications to mask torsion points, individual isogeny steps of degrees $O(\lambda \log \lambda)$ instead of 2 and 3, and larger parameter sizes. In SIDH, we have $O(\lambda \log \lambda)$ isogeny steps with optimal strategies [16], with each step costing $O(1)$ field operations. Field sizes are $O(\lambda)$ so each field operation costs $O(\lambda \log \lambda)$ bit operations asymptotically, neglecting $\log \log$ factors. This leads to a total asymptotic bit complexity of $O(\lambda^2 \log^2 \lambda)$ bit operations. In M-SIDH, we use $O(\lambda)$ primes each of size $O(\log \lambda)$, so the total prime size is $O(\lambda \log \lambda)$. There are still $O(\lambda \log \lambda)$ steps involved with optimal strategies. Each step requires $O(\sqrt{\lambda \log \lambda})$ field operations using square root Vélu formulae. Field operations cost $O(\lambda \log^2 \lambda)$ bit operations asymptotically, again neglecting $\log \log$ factors. This gives a total of $O(\lambda^{2.5} \log^{7/2} \lambda)$ bit operations. Concrete efficiency should be determined in future work, but a slowdown compared to SIDH should be expected, with a factor in the order of $O(\sqrt{\lambda} \log^{3/2} \lambda)$.

Most efficiency and implementation tricks developed for SIDH should also be available for M-SIDH, and potentially more, but we argue in Appendix B that the B-SIDH approach will not be applicable.

8 Conclusion and Perspectives

We introduced two variants of the SIDH protocols aimed at defeating the Castryck-Decru-Maino-Martindale-Robert recent attacks. The two variants respectively hide the secret isogeny degree and the torsion point information to the attacker (more precisely they only reveal an integer multiple of the degree, and they reveal torsion point images only up to a scalar).

Our thorough security analysis of both variants suggests that the M-SIDH variant offers the best security-efficiency tradeoff. Public key sizes are 4434, 7037 and 9750 bytes respectively for AES-128 (NIST level 1), AES-192 (NIST level 3)

and AES-256 (NIST level 5) security, and efficiency is expected to asymptotically be a factor in the order of $O(\sqrt{\lambda}\log^{3/2}\lambda)$ slower compared to SIDH.

Our work suggests that it may be possible to repair the SIDH protocol, although at a non negligible efficiency cost, and it similarly offers a way forward to the numerous cryptographic schemes based on SIDH that were developed in recent years. Further work should aim at developing additional countermeasures and at improving the efficiency and security analysis of our schemes.

Acknowledgements. We thank Castryck and Onuki for their valuable feedback on a preliminary version of the results in this paper, as well as those of the participants at ANTS 2022 that also gave us some feedback. The first author thanks Andrea Basso and Luca De Feo for several discussions regarding this work. We thank anonymous reviewers for their valuable feedback. This research was in part conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan. Christophe Petit's work is in part supported by an EPSRC fellowship grant (EP/V011324/1).

A On the claims of ePrint 2022/1667

The ePrint 2022/1667 vaguely claims attacks on M-SIDH. Reading through it, it clearly does not contain any attack against M-SIDH; it is easy to see that the "experimental evidence" provided there only applies to SIDH parameters and does not generalize to the parameters we recommend.

This ePrint paper runs the Castryck-Decru attack on Masked SIDH instantiated with SIDH primes, that is $A = 2^a$ and $B = 3^b$. Note that using SIDH primes in Masked SIDH is totally insecure at the first place. Nevertheless, when the 2^a torsion points are masked, intuitively, one expects the Castryck-Decru attack to succeed 50% of the time. In fact, there are 4 roots of unity modulo 2^a, these are 1, -1, $2^{a-1}-1$ and $2^{a-1}+1$. As precised earlier in Sect. 3.3, the attack succeeds when $\beta = 1, -1$, hence one expects the Castryck-Decru attack to succeed when the masking scalar β is 1 or -1, and fail when β is $2^{a-1}-1$ or $2^{a-1}+1$. The ePrint 2022/1667 ran the attack and noticed that the attack always succeeds, then claimed that this would be the case even when the correct parameters are used. We have already explained why we do not expect the attack to work on Masked degree instantiated with the correct parameters (see Sect. 3.3). Now, why does the Castryck-Decru attack works 100% of the time (instead of 50%) when instantiated with SIDH parameters? Well, it turns out it is because the Castryck-Decru attack does not fully use the torsion points provided in the public key, but scales them by a small power of 2 first. This is because the implementation of the attack needs a' and b' such that $c = 2^{a'} - 3^{b'}$ is smooth and its prime factors are congruent to 1 mod 4 (this is required for the attack to be efficient, see [7]). This implies that the order of the torsion points

actually used in the attack divides 2^{a-1}. Therefore, the masking scalar β which lies in $\{1, -1, 2^{a-1}-1, 2^{a-1}+1\}$ becomes $\beta \mod 2^{a-1} = 1, -1 \pmod{2^{a-1}}$. This justifies why the Castryck-Decru attack always succeeds when SIDH primes are used.

The attack clearly does not succeed when the torsion point images having order $2^{a'}$ are masked with a scalar which is neither 1 nor -1 modulo $2^{a'}$. This can be verified using the sage implementation of the attack provided in [30]. One goes to the line where the torsion point images of order $2^{a'}$ are computed (for example, in line 57 of the file *castryck_decru_shortcut.sage* in https://github.com/jack4818/Castryck-Decru-SageMath), and replaces the torsion points $2^{alp} * P_B$ and $2^{alp} * Q_B$ by $(2^{a_i-1}-1) * 2^{alp} * P_B$ and $(2^{a_i-1}-1) * 2^{alp} * Q_B$ respectively.

Note. The non-applicability of the attacks claimed in the ePrint 2022/1667 to M-SIDH was also pointed out on Twitter by Luca De Feo, Steven Galbraith, Péter Kutas, Benjamin Wesolowski and other isogenists, and we thank them for that.

B Using B-SIDH primes in M-SIDH

B-SIDH is one variant of SIDH proposed by Costello [12]. The main characteristic of B-SIDH is the use of quadratic twists. This allows us to use the torsion points in $E[p-1]$ and $E[p+1]$ without extending the base field, while in the original SIDH, points which we can use must be in $E[p+1]$. Thus, the size of the prime for B-SIDH is at most half that for SIDH.

If we can adapt this technique to our scheme, then the size of the prime may be at most halved. Since the MD-SIDH primes are larger than twice the M-SIDH primes, we only consider the case of M-SIDH.

In the setting of SIDH, the size of A needs to be large enough for its security; however, in the setting of M-SIDH, the number of primes dividing A needs to large enough. Therefore, the restriction of smoothness is harder in M-SIDH than in SIDH.

To use the B-SIDH method for M-SIDH, we need to find a prime p satisfying the following property:

$$p + 1 = \ell_1 \cdots \ell_t \cdot f,$$
$$p - 1 = q_1 \cdots q_t \cdot f',$$

where $t \geq 2\lambda$, and $\ell_1, \ldots, \ell_t$ and $q_1, \ldots, q_t$ are distinct primes, respectively.

The basic approach to find the B-SIDH prime is to construct an integer m such that both m and $m+1$ are smooth. If $2m+1$ is prime, we set $p = 2m+1$. In [12] and [13], some methods to find such m's are proposed. The current most useful method is the method proposed in [13]. The main idea of this method is to use already known solutions of the Prouhet-Tarry-Escott (PTE) problem, which provide pairs of integer coefficient polynomials $a(x) = (x-a_1)\cdots(x-a_s)$

and $b(x) = (x - b_1) \cdots (x - b_s)$ whose difference is a constant value c. If we find an integer ℓ such that all $\ell - a_i$'s and $\ell - b_i$'s are smooth, and $a(\ell)/c$ and $b(\ell)/c$ are integers, then $b(\ell)/c$ can be taken as m.

The main issue with this approach is that such ℓ's have a very small probability to exist. For a polynomial $a \in \mathbb{Z}[x]$, define

$$\Psi_a(N, M) = \#\{1 \leq m \leq N \mid a(m) \text{ is } M\text{-smooth}\}.$$

Then, heuristically it holds that $\Psi_a(N, N^{1/u})/N \sim \rho(d_1 u) \cdots \rho(d_k u)$ as $N \to \infty$, where $d_1, \ldots, d_k$ are degrees of distinct irreducible factors of a, and ρ is the Dickman–de Bruijn function.

Since $t \geq 2\lambda$, both m and $m + 1$ are divided by at least 2λ distinct primes. Then, we heuristically assume that the target value m is $m^{1/\lambda}$-smooth. Since $\ell \approx m^{1/s}$, the probability of target ℓ's is

$$\frac{\Psi_a(m^{1/s}, m^{1/\lambda})}{m^{1/s}} \sim \rho(\lambda/s)^s.$$

Note that s is less than or equal to 12 for an already known solution of the PTE problem. With $\lambda = 128$, we have $\rho(\lambda/s)^s < 2^{-463}$; with $\lambda = 192$, we have $\rho(\lambda/s)^s < 2^{-835}$; and with $\lambda = 256$, we have $\rho(\lambda/s)^s < 2^{-1246}$.

References

1. Basso, A., et al.: Supersingular curves you can trust. Cryptology ePrint Archive, Report 2022/1469 (2022). https://eprint.iacr.org/2022/1469
2. 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. LNCS, vol. 13090, pp. 160–184. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92062-3_6
3. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: efficient isogeny based signatures through class group computations. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11921, pp. 227–247. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34578-5_9
4. Booher, J., et al.: Failing to hash into supersingular isogeny graphs. Cryptology ePrint Archive, Report 2022/518 (2022). https://eprint.iacr.org/2022/518
5. Bottinelli, P., de Quehen, V., Leonardi, C., Mosunov, A., Pawlega, F., Sheth, M.: The dark SIDH of isogenies. Cryptology ePrint Archive, Report 2019/1333 (2019). https://eprint.iacr.org/2019/1333
6. Bröker, R.: Constructing supersingular elliptic curves. J. Comb. Numb. Theory **1**(3), 269–273 (2009)
7. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Report 2022/975 (2022). https://eprint.iacr.org/2022/975
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. LNCS, vol. 11274, pp. 395–427. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_15

9. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. J. Cryptol. **22**(1), 93–113 (2007). https://doi.org/10.1007/s00145-007-9002-x
10. Colò, L., Kohel, D.: Orienting supersingular isogeny graphs. Cryptology ePrint Archive, Report 2020/985 (2020). https://eprint.iacr.org/2020/985
11. Cooper, C.: On the rank of random matrices. Rand. Struct. Algor. **16**(2), 209–232 (2000). https://doi.org/10.1002/(SICI)1098-2418(200003)16:2⟨209::AID-RSA6⟩3.0.CO;2-1
12. Costello, C.: B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12492, pp. 440–463. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_15
13. Costello, C., Meyer, M., Naehrig, M.: Sieving for twin smooth integers with solutions to the prouhet-tarry-escott problem. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 272–301. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_10
14. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006). https://eprint.iacr.org/2006/291
15. De Feo, L., Galbraith, S.D.: SeaSign: compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 759–789. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_26
16. 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
17. 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. LNCS, vol. 12491, pp. 64–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_3
18. Fouotsa, T.B.: SIDH with masked torsion point images. Cryptology ePrint Archive, Report 2022/1054 (2022). https://eprint.iacr.org/2022/1054
19. Fouotsa, T.B., Petit, C.: A new adaptive attack on SIDH. In: Galbraith, S.D. (ed.) CT-RSA 2022. LNCS, vol. 13161, pp. 322–344. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95312-6_14
20. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the security of supersingular isogeny cryptosystems. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 63–91. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_3
21. Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. J. Cryptol. **33**(1), 130–175 (2019). https://doi.org/10.1007/s00145-019-09316-0
22. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th Annual ACM Symposium on Theory of Computing, pp. 212–219. ACM Press, Philadephia (1996). https://doi.org/10.1145/237814.237866
23. Jao, D., et al.: SIKE. Technical report, National Institute of Standards and Technology (2020). https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
24. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
25. Kovalenko, I., Levitskaya, A., Savchuk, M.: Selected Problems in Probabilistic Combinatorics. Naukova Dumka, Kiev (1986)

26. Maino, L., Martindale, C.: An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Report 2022/1026 (2022). https://eprint.iacr.org/2022/1026
27. Moriya, T.: Masked-degree SIDH. Cryptology ePrint Archive, Report 2022/1019 (2022). https://eprint.iacr.org/2022/1019
28. Mula, M., Murru, N., Pintore, F.: Random sampling of supersingular elliptic curves. Cryptology ePrint Archive, Report 2022/528 (2022). https://eprint.iacr.org/2022/528
29. National Institute of Standards and Technology: Post-quantum cryptography standardization (2016). https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization
30. 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
31. Petit, C.: Faster algorithms for isogeny problems using torsion point images. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10625, pp. 330–353. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_12
32. de Quehen, V., et al.: Improved torsion-point attacks on SIDH variants. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12827, pp. 432–470. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84252-9_15
33. Robert, D.: Breaking SIDH in polynomial time. Cryptology ePrint Archive, Report 2022/1038 (2022). https://eprint.iacr.org/2022/1038
34. Rostovtsev, A., Stolbunov, A.: Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145 (2006). https://eprint.iacr.org/2006/145
35. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. IEEE Computer Society Press, Santa Fe (1994). https://doi.org/10.1109/SFCS.1994.365700
36. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with cryptanalytic applications. J. Cryptol. **12**(1), 1–28 (1999). https://doi.org/10.1007/PL00003816