New SIDH Countermeasures for a More Efficient Key Exchange

Andrea Basso[1] and Tako Boris Fouotsa[2]

[1] University of Bristol, Bristol, United Kingdom
andrea.basso@bristol.ac.uk
[2] EPFL, Lausanne, Switzerland
tako.fouotsa@epfl.ch

Abstract. The Supersingular Isogeny Diffie-Hellman (SIDH) protocol has been the main and most efficient isogeny-based encryption protocol, until a series of breakthroughs led to a polynomial-time key-recovery attack. While some countermeasures have been proposed, the resulting schemes are significantly slower and larger than the original SIDH.
In this work, we propose a new countermeasure technique that leads to significantly more efficient and compact protocols. To do so, we introduce the concept of artificially oriented curves, which are curves with an associated pair of subgroups. We show that this information is sufficient to build parallel isogenies and thus obtain an SIDH-like key exchange, while also revealing significantly less information compared to previous constructions.
After introducing artificially oriented curves, we formalize several related computational problems and thoroughly assess their presumed hardness. We then translate the SIDH key exchange to the artificially oriented setting, obtaining the key-exchange protocols binSIDH, or binary SIDH, and terSIDH, or ternary SIDH, which respectively rely on fixed-degree and variable-degree isogenies.
Lastly, we also provide a proof-of-concept implementation of the proposed protocols. Despite being implemented in a high-level language, terSIDH has very competitive running times, which suggests that terSIDH might be the most efficient isogeny-based encryption protocol.

1 Introduction

Given two elliptic curves, finding an isogeny between them is widely believed to be a computationally hard problem. This has led to the development of several cryptographic protocols, whose security relies on the hardness of some isogeny-related problem. While the first constructions date back to 1996 [Cou06], the first practical isogeny-based protocol was the Supersingular Isogeny Diffie-Hellman (SIDH) key exchange [JD11]. After a decade of improvements and analysis, the protocol became the most efficient and well-known encryption scheme from isogenies, and it progressed through the four rounds of the NIST standardization process.

The security of the protocol, however, did not rely on the pure isogeny problem: finding an isogeny between two supersingular curves. The problem is hard, but its lack of structure makes it hard to obtain cryptographic functionalities off it. Thus, SIDH needed to reveal additional information in the form of torsion images: not only were the domain and codomain of the secret isogenies known, but also their actions on a torsion subgroup of coprime order. This additional information has been studied over the years, and it has been shown to lead to some active attacks [GPST16] and key-recovery attacks when the endomorphism ring of the two curves is known [GPST16] or when the protocol uses unbalanced parameters [Pet17,dQKL+21]. However, all these attacks came short of affecting the security of SIDH. The situation changed when a series of works [CD23,MMP+23,Rob23] developed a polynomial-time attack against SIDH for all possible parameters.

These attacks do not affect other isogeny-based protocols, such as CSIDH [CLM+18] and SQISign [DKL+20], but they affect those protocols that reveal images of torsion points, such as SÉTA [DdF+21]. Some countermeasures against the SIDH attacks have been proposed [FMP23]: they are based on scaling the torsion images (M-SIDH) or computing variable-degree isogenies (MD-SIDH). However, the complexity of the attacks against these protocols scale with number of distinct primes dividing the isogeny degrees: thus, to be secure, these protocols require extremely large parameters, which lead to high running times and communication costs.

Besides M-SIDH and MD-SIDH, the only valid encryption protocols based on isogenies are CSIDH [CLM+18], pSIDH [Ler22], and FESTA [BMP23]. However, the first two are both vulnerable to a subexponential

quantum attack [Pei20], which makes it hard to estimate the quantum security of a given parameter set. The more conservative estimates require large primes, which lead to impractically inefficient running times [CSCDJRH22]. While pSIDH has never been implemented, we can expect even lower efficiency than CSIDH, which leaves no practical encryption protocols based on isogenies. Very recently, the SIDH attacks were used constructively to develop a new public-key encryption protocol, FESTA. While the initial results are encouraging, the lack of an optimized implementation makes it hard to estimate the practicality of its running times.

In this work, we aim to fill the gap by proposing new countermeasures against the SIDH attacks that lead to a practically efficient SIDH-like key-exchange protocol. To do so, we introduce the concept of artificial orientations: an artificial A-orientation $\mathfrak{A}$ on a supersingular curve E is a pair of cyclic disjoint subgroups of $E[A]$ of order A. Given an artificial orientation $\mathfrak{A} = (G_1, G_2)$, an $\mathfrak{A}$-isogeny ϕ is an isogeny whose kernel is a cyclic subgroup of $G_1 \oplus G_2$. In other words, ϕ can be written as the composition $\phi = \phi_2 \circ \phi_1$, where $\ker \phi_1 \subset G_1$, $\ker \phi_2 \subset \phi_1(G_2)$, and the degrees of ϕ_1 and ϕ_2 are coprime. While an artificial orientation does not reveal the same information as a real orientation [CK20], it provides an interpolation between the original SIDH construction and the oriented protocols, such as CSIDH [CLM+18], OSIDH [CK20], and SCALLOP [DFFK+23]. On one hand, artificial orientations and their images provide enough information to compute parallel isogenies, similarly to torsion images in SIDH; on the other, orientations always imply an artificial orientation, because given an orientation it is possible to recover the images of two cyclic disjoint groups, i.e. an artificial orientation. For example, in CSIDH, the images of the groups $\ker(\pi - 1) \cap E[\ell]$ and $\ker(\pi + 1) \cap E[\ell]$ under the secret isogeny $\phi : E \to E'$ are given by $\ker(\pi - 1) \cap E'[\ell]$ and $\ker(\pi + 1) \cap E'[\ell]$, respectively [CHM+23, Section 6.1].

Contributions. In this paper, we formalize the concept of artificial orientations and introduce some computational problems related to artificially oriented isogenies. We thoroughly assess the presumed hardness of these problems and we survey potential attacks. Then, we propose binSIDH, or binary SIDH, the first protocol that translates SIDH to the artificially-oriented setting. As in SIDH, this key exchange is limited to fixed-degree isogenies, which is helpful to develop constant-time implementations and zero-knowledge proofs of isogeny knowledge. Then, we generalize binSIDH to the case of variable-degree isogenies to obtain terSIDH, or ternary SIDH, which achieves smaller parameters.

The two protocols, binSIDH and terSIDH, require both parties to use artificially oriented isogenies, which results in a balanced protocol where the computational requirements of both parties is similar. We thus propose a new technique that allows one party to compute SIDH-like isogenies, at the cost of the other party computing longer oriented ones. This allows one party to be significantly more efficient, which is particularly useful in advanced protocols between clients and servers with unbalanced computational power: not only the client can be more efficient than the server, but if the protocol requires proofs of isogeny knowledge, those of the client can be computed much more efficiently as well. Since the same technique can be applied to binSIDH and terSIDH, we obtain two new variants: $\mathsf{binSIDH}^{\mathsf{hyb}}$ and $\mathsf{terSIDH}^{\mathsf{hyb}}$.

Lastly, we generate parameter sets for all four protocols, for all security levels, and we provide a SageMath proof-of-concept implementation of all proposed protocols. Despite being implemented in a high-level language, terSIDH has very competitive running times when compared to existing implementations of other isogeny-based encryption schemes.

2 Preliminaries

In this section, we briefly introduce the SIDH protocol and the recent key-recovery attacks. For more background information on elliptic curves and isogenies, we refer the reader to [Sil09].

2.1 SIDH

SIDH, or Supersingular Isogeny Diffie-Hellman [JD11], is a key-exchange protocol based on isogenies between supersingular elliptic curves. The main protocol parameter is a prime p of the form $p = ABf - 1$, where $A = 2^a$ and $B = 2^b$, and a starting supersingular curve E_0 defined over $\mathbb{F}_{p^2}$. The protocol also specifies two bases P_A, Q_A and P_B, Q_B that generate, respectively, $E_0[A]$ and $E_0[B]$.

The first party, say Alice, generates her public key by sampling a random secret key $\mathsf{sk}_A = \alpha \in \mathbb{Z}_A$, computing the isogeny $\phi_A : E_0 \to E_A$ with kernel $\ker \phi_A = \langle P_A + [\alpha]Q_A \rangle$, and revealing $\mathsf{pk}_A = (E_A, R_A =$

$\phi_A(P_B), S_A = \phi_A(Q_B))$. The second party, say Bob, proceeds analogously with an isogeny of degree B: he samples $\mathsf{sk}_B = \beta \in \mathbb{Z}_B$, computes the isogeny $\phi_B : E_0 \to E_B$ with kernel $\ker \phi_B = \langle P_B + [\beta]Q_B \rangle$, and reveals $\mathsf{pk}_B = (E_B, R_B = \phi_B(P_A), S_B = \phi_B(Q_A))$. Then, after exchanging public keys, both parties can obtain the same shared secret by computing the push-forward of their isogeny under the other party's isogeny. Concretely, Alice computes the isogeny $\phi'_A : E_B \to E_{AB}$ with kernel $\ker \phi'_A = \langle R_B + [\alpha]S_B \rangle = \phi_B(\ker \phi_A)$, while Bob computes the isogeny $\phi'_B : E_A \to E_{BA}$ with kernel $\ker \phi'_B = \langle R_A + [b]S_A \rangle = \phi_A(\ker \phi_B)$. The two isogenies are the correct push-forwards, and thus $\phi_A, \phi_B, \phi'_A, \phi'_B$ form a commutative diagram. Hence, the codomain curves E_{AB} and E_{BA} are isomorphic, and their j-invariant is the shared secret known to both Alice and Bob.

2.2 Polynomial time attacks on SIDH

The security of the SIDH protocol relies on the hardness of recovering a secret isogeny from its action on a torsion basis. In a series of works by Castryck and Decru [CD23], Maino, Martindale, Panny, Pope and Wesolowski [MMP+23], and Robert [Rob23], the authors show the problem can be solved in polynomial time when the torsion information is sufficiently large compared to the degree of the isogeny. This leads to an efficient key-recovery attack on all instances of SIDH.

The attacks slightly vary in their techniques, but they all rely on Kani's theorem [Kan97], which states that given an SIDH square with specific properties, there exists a genus-two isogeny between the abelian surface obtained by gluing two curves in the SIDH square to the abelian surface obtained by gluing the other two curves in the square. It is possible to generate an SIDH square with the desired properties and compute the genus-two isogeny from the image points revealed in SIDH; evaluating such an isogeny allows an attacker to evaluate the secret isogeny on any point, which in turn can be used to recover the secret isogeny.

For the purpose of this work, the SIDH attacks can be abstracted as a generic algorithm that recovers a isogeny $\phi : E_0 \to E_1$ of degree d when it receives the curves E_0, E_1, the degree d, and the points P_0, Q_0 and $\phi(P_0), \phi(Q_0)$, where P_0, Q_0 are linearly independent points of order n and $n^2 \geq d$. There is no known technique that allows extending such attacks to a case where the image points are not known exactly: indeed, all attacks on the proposed countermeasures [FMP23], as well as the potential attacks discussed in this work, need to recover the exact torsion images to apply the attacks.

3 Artificial orientations

In this section, we introduce artificial orientations, the main ingredient that powers the countermeasures against the SIDH attacks. Unless stated otherwise, the integers A and B are assumed to be smooth and co-prime.

Artificial orientations are composed of two independent subgroups. This is formalized in the Definition 1, and we provide more information on how to explicitly compute such isogenies in Eq. (2) in Section 4.1.

Definition 1. *Let E be a supersingular curve defined over $\mathbb{F}_{p^2}$, and let A be an integer. An* artificial A-orientation (of E) *is a pair $\mathfrak{A} = (G_1, G_2)$ where $G_1, G_2 \subset E[A]$ are cyclic groups of order A and $G_1 \cap G_2 = \{0\}$. $(E, \mathfrak{A})$ is called an* artificially A-oriented curve.

Given an artificially A-oriented curve $(E, \mathfrak{A})$, one can compute a range of isogenies whose kernels arise from $\mathfrak{A} = (G_1, G_2)$. We formalize this concept, which we call $\mathfrak{A}$-isogenies, in the following definition.

Definition 2. *Let $(E, \mathfrak{A})$ where $\mathfrak{A} = (G_1, G_2)$ be an artificially A-oriented curve. An isogeny $\phi : E \to E'$ is said to be an* $\mathfrak{A}$-isogeny *if $\ker \phi$ is the direct sum of a subgroup of G_1 and a subgroup of G_2, that is $\ker \phi = H_1 \oplus H_2$ where H_i is a subgroup of G_i for $i = 1, 2$.*

If $(E, \mathfrak{A})$ is an artificially A-oriented curve and $\phi : E \to E'$ is a non-trivial $\mathfrak{A}$-*isogeny*, then the artificially A-orientation on E cannot be carried onto E' through ϕ. In fact, since ϕ is non-trivial and $\ker \phi$ is the direct sum of a subgroup of G_1 and a subgroup of G_2, then at least one of the groups $\phi(G_1)$ and $\phi(G_2)$ has order strictly smaller than A. In order to be able to carry the artificially A-orientation on E onto E' it is necessary that the degree of the isogeny considered is coprime to A. We have to following definition for artificially A-oriented B-isogenous curves.

Definition 3. *Let $(E, \mathfrak{A})$ and $(E', \mathfrak{A}')$ be two artificially A-oriented curves and let B be an integer coprime to B. We say that $(E, \mathfrak{A})$ and $(E', \mathfrak{A}')$ are* B-isogenous *if there exist an isogeny $\phi : E \to E'$ of degree B such that $\mathfrak{A}' = \phi(\mathfrak{A})$, that is if $\mathfrak{A} = (G_1, G_2)$ and $\mathfrak{A}' = (G_1', G_2')$, then $G_1' = \phi(G_1)$ and $G_2' = \phi(G_2)$.*

Remark 4. Note that B-isogenous oriented curves include images of subgroups. These can be represented by a choosing a random generator. Thus, fixed generators $\langle P_1 \rangle = G_1$ and $\langle P_2 \rangle = G_2$, the subgroups G_1' and G_2' are represented by $[\alpha]\phi(P_1)$ and $[\beta]\phi(P_2)$ respectively, for some unknown $\alpha, \beta \in \mathbb{Z}_A$.

3.1 A comparison of $\mathfrak{A}$-isogenies with existing techniques

In this section, we discuss the main differences between artificially oriented isogenies and isogenies arising from group actions (real orientations) on one hand, SIDH-like isogenies and M-SIDH-like isogenies.

$\mathfrak{A}$-isogenies vs group actions. Artificially oriented isogenies share similarities with those that arise from group actions, such as the isogenies in CSIDH, OSIDH, and SCALLOP. In both instances, isogenies are restricted to specific subsets of all possible isogenies, and the action of secret isogenies on two independent subgroups is revealed. However, artificially oriented isogenies are significantly different from those in CSIDH and SCALLOP: first, given any supersingular curve, it is always possible to attach an artificial orientation to it, unlike in CSIDH, where the curves need to be defined over $\mathbb{F}_p$ and the orientation is already available through the Frobenius; or SCALLOP, where not all supersingular curves are oriented and a real orientation needs to be provided. Most importantly, artificial orientations do not give rise to a commutative group action like in real orientations, which means that the quantum subexponential attack by Childs, Jao, and Soukharev [CJS14] does not apply. Similarly, artificial orientations are also immune to the attacks on OSIDH [DD21].

$\mathfrak{A}$-isogenies vs SIDH. The main difference between SIDH-like isogenies and artificially A-oriented isogenies is the amount of information needed to compute their push-forwards. In the SIDH case, the kernel of the isogeny (say ψ) is generated by a point of the form $P + [\alpha]Q$. The kernel of the push-forward of ψ through ϕ is generated by the point $\phi(P) + [\alpha]\phi(Q)$. Therefore, the images of torsion points P and Q are needed in order to compute the push-forward of ψ through ϕ. Conversely, $\mathfrak{A}$-isogenies are limited to those that arise from $\mathfrak{A}$. Hence, only the push-forward of the artificial orientation is needed, which means only the images of two cyclic torsion groups are revealed. This prevents torsion point attacks [Pet17,dQKL+21,CD23,MMP+23,Rob23].

$\mathfrak{A}$-isogenies vs M-SIDH. In M-SIDH and MD-SIDH [FMP23], isogenies are defined as in SIDH, but to compute their push-forwards, the torsion points images are revealed while scaled (or masked) with the same scalar β. This means that instead of revealing $\phi(P)$ and $\phi(Q)$ as in SIDH, one reveals $[\beta]\phi(P)$ and $[\beta]\phi(Q)$. This is significantly more information than what is revealed to compute the push-forwards of $\mathfrak{A}$-isogenies, since the image of an artificial orientation is equivalent, as discussed in Remark 4, to revealing $[\alpha]\phi(P)$ and $[\beta]\phi(Q)$, for independent values α and β. From a subgroup perspective, push-forwards of $\mathfrak{A}$-isogenies require the images of two cyclic disjoint subgroups, whereas M-SIDH reveals two image points scaled with the same value, which is equivalent to the images of three cyclic disjoint groups of order $\mathrm{ord}(P)$ (see [BKM+21, Lemma 1] and [FP22, Lemma 1]).

3.2 Security assumptions

Having introduced artificial orientations, we now introduce three computational problems that relate to artificially oriented curves and isogenies. The first problem, which we refer to as the Supersingular Isogeny Problem for artificially A-oriented curves (SSIP-A), asks to recover an isogeny given its domain, together with an artificial orientation, and its codomain, together with a compatible orientation.

Problem 5 (SSIP-A). Let $(E, \mathfrak{A})$ be an artificially A-oriented curve and let B be an integer coprime to A. Let $\phi : E \to E'$ be a cyclic isogeny of degree B and let $\mathfrak{A}' = \phi(\mathfrak{A})$. Given $(E, \mathfrak{A})$ and $(E', \mathfrak{A}')$ and the degree $\deg \phi$, compute ϕ.

In Problem 5, there is no constraint on the isogeny ϕ, apart from its degree being B. When an artificial B-orientation $\mathfrak{B}$ is provided on E, then one may restrict to $\mathfrak{B}$-isogenies. This leads to the (supersingular) Artificially Oriented Isogeny Problem (AOIP).

Problem 6 (AOIP). Let $(E, \mathfrak{A})$ an artificially A-oriented curve and let B be an integer coprime to B. Let $\mathfrak{B}$ be an artificial B-orientation on E. Let $\phi : E \to E'$ be a cyclic $\mathfrak{B}$-isogeny of degree B and let $\mathfrak{A}' = \phi(\mathfrak{A})$. Given $(E, \mathfrak{A})$ and $(E', \mathfrak{A}')$, compute ϕ.

We can also study a problem that is, in some sense, the converse of Problem 5. Rather than considering general isogenies and the image of an artificial orientation, we can focus on the case where the isogeny is artificially oriented, but more torsion image information is revealed. This is summarized in the Supersingular Isogeny Problem for $\mathfrak{B}$-isogenies (SSIP-B) problem.

Problem 7 (SSIP-B). Let $(E, \mathfrak{A})$ be an artificially A-oriented curve and let B be an integer coprime to A. Let $\phi : E \to E'$ be a cyclic $\mathfrak{A}$-isogeny of degree A, with $B \ll A$. Given $(E, \mathfrak{A})$, the curve E' and the unscaled image of the B-torsion $\phi(E[B])$, compute ϕ.

Such a problem can be solved with the techniques introduced in the SIDH attacks [CD23,MMP+23,Rob23]. However, as we will see in Section 5, it is useful to study Problem 7 for those parameters where the SIDH attacks do not apply.

3.3 Hardness analysis

In this section, we study the computational problems that we introduced, analyze potential attacks, and justify their assumed hardness.

Finding an isogeny from the orientation image. The first problem, Problem 5 is already known in the literature, as it was recently introduced with a different notation in [BMP23, Problem 7], where it was called the Computational isogeny with scaled-torsion (CIST) problem. As argued in [BMP23], the problem appears to be hard because the images of two subgroups do not provide enough information for the SIDH attacks to be applicable. Given two images $[\alpha]\phi(P)$ and $[\beta]\phi(Q)$, scaled by independent values α and β, an attacker can easily recover the product $\alpha\beta$ from pairing computations, but this is similarly insufficient to recover the exact images that would enable the SIDH attacks. An attacker may attempt to bruteforce the missing information, but this is computationally infeasible if the degree of the secret isogeny is sufficiently long, which in turn makes the order of the torsion information to be guessed large enough for the attack to be infeasible. Note that the information revealed in Problem 5 is comparable to that in CSIDH and SCALLOP, and significantly less than that in M-SIDH and MD-SIDH. It is thus likely that an attack that can solve Problem 5 in its most general form, can do so for such protocols as well.

Since not enough information is revealed for the SIDH attacks to apply, the attack on starting curves with small endomorphisms [FMP23] does also not apply here. It is thus possible to choose a starting curve with known endomorphism ring. Very recent analysis [CV23] has shown it is possible to recover an isogeny from its scaled action and thus solve Problem 5 when the starting curve E_0 and the corresponding orientation have specific properties relative to the Frobenius conjugate of E_0. It is thus important to select parameters that avoid these issues; since the endomorphism ring of the starting curve can be public, this can be done in a transparent manner without the need of a trusted setup.

Finding an oriented isogeny from the orientation image. In Problem 6, the degree of the isogeny ϕ is not necessarily known: the degree of a $\mathfrak{B}$-isogeny can range across all values dividing the order B of the subgroups in $\mathfrak{B}$, which poses a first barrier to the application of the SIDH attacks. However, even if we restrict to isogenies of full degree, i.e. $\deg \phi = B$, the torsion information that is revealed is the same as that in Problem 5, and thus a similar analysis follows. The fact that the unknown isogeny is a $\mathfrak{B}$-isogeny does not interact in any meaningful way with the SIDH attacks or the revealed torsion information: as such, it appears to be hard for an attacker to exploits such attacks to solve Problem 6. Hence, it seems likely that any attack would have to disregard the artificial orientation and focus on recovering an isogeny between two given curves; however, since the isogeny is a $\mathfrak{B}$-isogeny, this problem is easier than the general case.

First, an attacker can simply brute force all the possible isogenies. If we restrict ourselves to isogenies of full degrees, there are 2^t possible $\mathfrak{B}$-isogeny, where t is the numbers of primes dividing B. This suggests that the degree of the isogeny should be the product of at least $t = \lambda$ distinct primes. Second, generic attacks to recover an isogeny between to given curves, such as the meet-in-the-middle, van Orschoot-Wiener [vW99], Delfs-Galbraith [DG16] attacks, are not applicable since the prime characteristic and the isogeny degree, being the product of at least λ distinct primes, are sufficiently large to make these attacks computationally infeasible. However, it is possible to devise an enhanced meet-in-the-middle attack that exploits the nature of the $\mathfrak{B}$-isogenies: fixed an attack parameter $0 \leq t' \leq t$, the attacker computes $2^{t'}$ $\mathfrak{B}$-isogenies starting from E_0. These are chosen of the largest degree, i.e. the attacker first computes the isogenies with degree corresponding to the largest primes dividing B, so that the end curves are as close to E' as possible. The attacker stores the j-invariants of the codomain curves and starts a random walk of the correct degree from E', in the hope of finding a collision. The cost of the attack depends on the choice of t': the first part requires $2^{t'}$ computations, while the second part requires computing all the possible isogenies of a specific degree (the product of the smaller $t - t'$ primes dividing B, assuming that B is square-free) starting from E'. This technique yields a better attack than a simple bruteforce approach, and thus it would require larger parameters, albeit only moderately larger ones.

Example 8. For instance, when B is the product of the first 128 primes (the case most suitable to this attack), the attack is optimal for $t = 106$, which corresponds to an attack where 2^{106} isogenies are computed and 2^{106} j-invariants are stored in memory. Thus, to obtain $\lambda = 128$ bits of security, we would need the $\mathfrak{B}$-isogeny needs to have a degree B that is the product of the $t = 154$ smallest primes. We remark that the security estimates depend not only on the number of distinct primes dividing B, but also on the size of the specific primes.

The previous attack considers an attacker that has accesses to unbounded memory. This is far from realistic, and we can obtain better estimates of the attack possibilities when we impose an upper bound to the amount of memory available. We follow the security analysis of SIDH [ACC+19,JAC+20], and we limit our analysis to attackers with 2^{80} units of memory for any security level.[3] In this setting, the best attack is a van Orschoot-Wiener version of the enhanced meet-in-the-middle attack presented before, which allows the attacker to trade higher computational costs for a lower memory requirement. As shown in [ACC+19], a van Orschoot-Wiener search has a computational cost of approximately

$$N^{3/2}/w^{1/2},$$

where N is the number of collision points and w is the memory units available. In our case, we have $N = 2^{t'}$, and $w = 2^{80}$. This suggests that, for $\lambda = 128$, this attack outperforms a bruteforce search, but only marginally. If we set the degree B to be the product of the first t distinct primes, the enhanced van Orschoot-Wiener attack requires $t = 137$ (compared to $t = 128$, as suggested by the bruteforce attack). However, for higher security levels, the bruteforce attack outperforms enhanced van Orschoot-Wiener attack, because the memory bound remains constant across all security levels, and thus it has a larger performance impact on higher security levels. Thus, for $\lambda \in \{192, 256\}$, we can choose $t = \lambda$.

The case with variable-degree isogenies follows similarly. The parameter t initially needs to be selected to avoid the bruteforce attack, where the specific value depends on the exponents of the primes dividing B. The enhanced MITM and vOW attacks similarly apply to the variable-degree case: in this case, however, the enhanced vOW attack outperforms the bruteforce attack at all commonly used security levels, and thus the parameters need to be slightly larger than what the bruteforce attack would suggest.

Finding an isogeny from the full torsion image. Lastly, Problem 7 is vulnerable to the SIDH attacks, as discussed when introduced. However, the $\mathfrak{A}$-isogeny needs to have a large degree A to be secure from the attacks outlined above, and thus the torsion points would need a large order B for the SIDH attacks to be applicable. More precisely, the attacks are possible when $B^2 \geq A$, but an attacker could guess part of the isogeny so that the remaining part is short enough to be recovered through the SIDH attacks. This would suggest that if $2^{t'} B^2 \approx A$, an attacker can recover the unknown isogeny after

[3] More precisely, we consider attackers that can store up to 2^{80} j-invariants. Given the size of the primes used, this corresponds to more than 2^{90} bits of memory.

iterating through $2^{t'}$ isogenies. This is the case for generic isogenies, but in the case of oriented ones, the attacker can bruteforce much longer isogenies at the same cost, since there are only limited options for any prime degree dividing A. In particular, after $2^{t'}$ computations, the attacker obtains isogenies of degree $A_{t'}$, the product of the t' largest primes dividing A. Thus, Problem 7 is secure against the SIDH attacks when $A_{t'}B^2 \leq A$.

Assuming this condition is satisfied, Problem 7 appears to be secure since the oriented-isogeny structure does not interact with the revealed torsion information, which does not make the problem easier. Lastly, before the attacks by Castryck and Decru, Maino, Martindale, Panny, Pope and Wesolowski, and Robert, SIDH with unbalanced parameter was vulnerable to torsion-point attacks [Pet17,dQKL+21] that relied on knowledge of the endomorphism ring of the starting curve. These attacks similarly do not apply to Problem 7 since the torsion information is much lower than what is needed.

4 The binSIDH and terSIDH protocols

In this section, we propose two new protocols: binSIDH and terSIDH. Both protocols translate the SIDH key exchange to the setting of artificially oriented curves and isogenies. The former restricts itself to fixed-degree isogenies, while the latter relies on variable-degree isogenies to improve on efficiency and compactness.

4.1 binSIDH

We first introduce binSIDH, which restricts itself to isogenies of full degree. The protocols rely on the fact that A-oriented curves provide sufficient information to compute parallel isogenies. More formally, let A be a product of t distinct primes $A = \prod_{i=1}^{t} p_i$ and write $A = A_1A_2$ for a multiplicative splitting of A with $\gcd(A_1, A_2) = 1$. Then, given two A-oriented curves $(E, \mathfrak{A})$ and $(E', \mathfrak{A}')$ connected by a B-isogeny $\phi : E \to E'$, where $\mathfrak{A} = (\langle G_1\rangle, \langle G_2\rangle)$ and $\mathfrak{A}' = (\langle G_1'\rangle, \langle G_2'\rangle)$, the isogenies

$$\psi : E \to E/\langle [A_1]G_1 + [A_2]G_2\rangle, \qquad \psi' : E' \to E'/\langle [A_1]G_1' + [A_2]G_2'\rangle$$

are parallel, i.e. we have $\ker \psi' = \phi(\ker \psi)$ and the codomain curves are also B-isogenous, connected by the isogeny ϕ' with kernel $\ker \phi' = \psi(\ker \phi)$.

The isogenies ψ and ψ' are thus determined by the splitting of A as $A = A_1A_2$. In other words, if we represent the subgroups $\langle G_1\rangle$ and $\langle G_2\rangle$ as

$$\begin{aligned} \langle G_1\rangle &= \langle G_1^1, G_1^2, \ldots, G_1^t\rangle, \\ \langle G_2\rangle &= \langle G_2^1, G_2^2, \ldots, G_2^t\rangle, \end{aligned} \quad \text{where} \quad \begin{cases} \operatorname{ord}(G_i^1) = p_i, \\ \operatorname{ord}(G_i^2) = p_i, \end{cases} \tag{1}$$

then the kernel of ψ is determined by selectively choosing either G_1^i or G_2^i to be in the kernel of ψ, for every $i \in [t]$. The same holds for the isogeny ψ' and the generators G_1' and G_2'. This suggests the following notation: fixed an artificial A-orientation $(E, \mathfrak{A} = (G_1, G_2))$, where $A = \prod_{i=1}^{t} p_i$, we can associate a vector $\mathbf{a} \in \{1, 2\}^t$ to any $\mathfrak{A}$-oriented isogeny ϕ by writing

$$\ker \phi = \langle G_{\mathbf{a}_1}^1, G_{\mathbf{a}_2}^2, \ldots, G_{\mathbf{a}_t}^t\rangle, \tag{2}$$

where the points G_1^j and G_2^j are defined as in Eq. (1) and $\mathbf{a}_i$ denotes the i-th element of $\mathbf{a}$. Throughout the rest of the paper, we write $\langle \mathbf{a}, \mathfrak{A}\rangle$ to denote the subgroup corresponding to the orientation $\mathfrak{A}$ with secret vector $\mathbf{a}$, as computed in Eq. (2).

We showed in Section 3.2 that we consider secure to reveal artificially oriented curves since the SIDH attacks are inapplicable. Moreover, artificial orientations allow computations of parallel isogenies, and if the order A is sufficiently composite, the number of potential parallel isogenies is exponentially large. That is because the value A is the product of t distinct primes, which means there are 2^t potential splittings $A = A_1A_2$. This suggests it is possible to replicate the SIDH key exchange with artificially oriented isogenies and to obtain a secure protocol that is immune to the SIDH attacks. We call the resulting construction binSIDH, and we represent it in Fig. 1.

Setup. Let λ be the security parameter and t 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 $\mathbb{F}_{p^2}$. Let $\mathfrak{A}$ be an artificial A-orientation on E_0 and let $\mathfrak{B}$ be an artificial B-orientation on E_0. The public parameters are E_0, p, A, B, $\mathfrak{A}$ and $\mathfrak{B}$.

KeyGen. Alice samples uniformly at random a vector $\mathbf{a}$ from $\{1, 2\}^t$ and computes the $\mathfrak{A}$-oriented isogeny $\phi_A : E_0 \to E_A$ of degree A defined by $\mathbf{a}$. She also computes the push forward $\mathfrak{B}'$ of $\mathfrak{B}$ on E_A through ϕ_A. Her secret key is $\mathbf{a}$ and her public key is $(E_A, \mathfrak{B}')$. Analogously, Bob samples uniformly at random a vector $\mathbf{b}$ from $\{1, 2\}^t$ and computes the $\mathfrak{B}$-oriented isogeny $\phi_B : E_0 \to E_B$ of degree B defined by $\mathbf{b}$. He also computes the push forward $\mathfrak{A}'$ of $\mathfrak{A}$ on E_A through ϕ_B. His secret key is $\mathbf{b}$ and his public key is $(E_B, \mathfrak{A}')$.

SharedKey. Upon receiving Bob's public key $(E_B, \mathfrak{A}')$, Alice checks that $\mathfrak{A}'$ is an artificial A-orientation on E_B, if not she aborts. She computes the $\mathfrak{A}'$-oriented isogeny $\phi'_A : E_B \to E_{BA}$ of degree A defined by $\mathbf{a}$. Her shared key is $j(E_{BA})$. Similarly, upon receiving $(E_A, \mathfrak{B}')$, Bob checks that $\mathfrak{B}'$ is an artificial B-orientation on E_A, if not he aborts. He computes the $\mathfrak{B}'$-oriented isogeny $\phi'_B : E_A \to E_{AB}$ of degree B defined by $\mathbf{b}$. His shared key is $j(E_{AB})$.

Fig. 1: The binSIDH protocol.

4.2 The terSIDH variant

We now introduce terSIDH, a variant of binSIDH that is more efficient and more compact, but these improvements come at the cost of relying on variable-degree isogenies. In binSIDH, every A-oriented isogeny ϕ is determined by a binary choice for each prime p_i dividing A: the p_i-degree isogeny has kernel generated by either G_1^i or G_2^i. However, we can introduce a third option by allowing the isogeny to not have a p_i component. In other words, write ϕ as the composition of t isogenies $\phi = \phi_t \circ \dots \phi_2 \circ \phi_1$; then, the isogeny ϕ_i has kernel generated by G_1^i, G_2^i, or $\mathcal{O}$. We thus extend the notation introduced in the previous section by letting the vector $\mathbf{a}$ to have entries in $\{0, 1, 2\}$, and we set $G_0^i = \mathcal{O}$ for all $i \in [t]$. The full protocol is described in Fig. 2.

Compared to binSIDH, terSIDH introduces more choices for each prime p_i. In particular, it provides three choices, which means that every p_i dividing $p + 1$ provides $\log_2 3 \approx 1.6$ bits of security. Interestingly, terSIDH is the first countermeasure technique against the SIDH attacks that can provide more than one bit of security per prime p_i. This means that, to provide enough security, the isogeny degrees should be at least the product of $t \approx \lambda/1.6$ primes, and thus terSIDH can use significantly smaller parameters and shorter isogenies, leading to a more efficient and more compact protocol. However, to achieve this, we necessarily rely on variable-degree isogenies. This has some disadvantages: from an implementation perspective, the varying degree may make it harder to obtain constant-time implementations, as seen in the case of CSIDH implementations [BBC+21,CSCDJRH22]. The other issue, as argued in [Bas23,BFGP23], is that it appears to be hard to construct zero-knowledge proofs of variable-degree isogenies because all known approaches invariably leak the secret isogeny degree. This causes a major issue in the development of proofs of terSIDH public key correctness, and it may prevent terSIDH from being an SIDH drop-in replacement for advanced constructions.

Setup. Let λ be the security parameter and t 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 $\mathbb{F}_{p^2}$. Let $\mathfrak{A}$ be an artificial A-orientation on E_0 and let $\mathfrak{B}$ be an artificial B-orientation on E_0. The public parameters are E_0, p, A, B, $\mathfrak{A}$ and $\mathfrak{B}$.

KeyGen. Alice samples uniformly at random a vector $\mathbf{a}$ from $\{0,1,2\}^t$ and computes the $\mathfrak{A}$-oriented isogeny $\phi_A : E_0 \to E_A$ of degree A defined by a. She also computes the push forward $\mathfrak{B}'$ of $\mathfrak{B}$ on E_A through ϕ_A. Her secret key is a and her public key is $(E_A, \mathfrak{B}')$. Analogously, Bob samples uniformly at random a vector $\mathbf{b}$ from $\{0,1,2\}^t$ and computes the $\mathfrak{B}$-oriented isogeny $\phi_B : E_0 \to E_B$ of degree B defined by b. He also computes the push forward $\mathfrak{A}'$ of $\mathfrak{A}$ on E_A through ϕ_B. His secret key is b and his public key is $(E_B, \mathfrak{A}')$.

SharedKey. Upon receiving Bob's public key $(E_B, \mathfrak{A}')$, Alice checks that $\mathfrak{A}'$ is an artificial A-orientation on E_B, if not she aborts. She computes the $\mathfrak{A}'$-oriented isogeny $\phi'_A : E_B \to E_{BA}$ of degree A defined by a. Her shared key is $j(E_{BA})$. Similarly, upon receiving $(E_A, \mathfrak{B}')$, Bob checks that $\mathfrak{B}'$ is an artificial B-orientation on E_A, if not he aborts. He computes the $\mathfrak{B}'$-oriented isogeny $\phi'_B : E_A \to E_{AB}$ of degree B defined by b. His shared key is $j(E_{AB})$.

Fig. 2: The terSIDH protocol. This is nearly the same as Fig. 1, with the main difference being that KeyGen samples ternary secrets.

4.3 One more variant

It is possible to define a third variant of these protocols that relies on partial artificial orientations. Rather than revealing the images of two linearly independent points G_1 and G_2, the protocol only reveals the image of one point G. Then, for each G^i of coprime order that make up G, the possible isogenies are computed by choosing whether G^i in the kernel of the isogeny or not. Using the vector notation, its entries are chosen in $\{0,1\}$.

Since the choice is binary, such a protocol would require similar parameters as binSIDH, while also having the disadvantages of variable-degree isogenies discussed in the context of terSIDH. As such, it does not appear to have any meaningful advantage over the proposed constructions. However, the information that is revealed about the secret isogeny is less: not only its degree remains unknown, as in terSIDH, but its action on a single cyclic group is revealed. This suggests that such a variant might be relevant if further cryptanalytic breakthroughs affect the security of binSIDH and terSIDH.

5 An oriented/non-oriented hybrid approach

There are applications where it is desirable for one party to be significantly more efficient than the other. For example, this is the case for resource-constrained devices communicating to powerful servers, but it also arises in advanced constructions: for instance, in oblivious pseudorandom function protocols, it is generally desired that the client is more efficient than the server. In this section, we propose a technique that allows us to introduce trade-offs between the two parties and enable one participant to obtain more efficient zero-knowledge proofs, which makes this approach more appealing for advanced protocols that requires proofs of isogeny knowledge. This technique has the added benefit of reducing the overall prime size for binSIDH, while the ternary variant has primes of comparable size as terSIDH.

In the previously presented protocols, both parties relied on artificially oriented isogenies to avoid the SIDH attacks. However, the artificial orientation also requires to use significantly longer isogenies than those used in the original SIDH protocol. This suggests that it may be possible to reveal some unscaled torsion information without affecting the security of the protocol, and if the isogeny is sufficiently long, the revealed torsion may be large enough to allow the computation of parallel isogenies that also guarantee sufficient security. In other words, we can build a secure protocol through a hybrid approach where one party computes binSIDH-like (or terSIDH-like) isogenies while the other party computes SIDH-like isogenies.

More formally, let Bob denote the party computing binSIDH-like isogenies, which means he computes artificially B-oriented isogenies where $B = \ell_1 \cdots \ell_n$; let Alice be the party computing SIDH-like isogenies of degree A, i.e. isogenies whose kernel is generated by $P_A + [\alpha]Q_A$, for some secret $\alpha \in \mathbb{Z}_A$ and fixed points P_A, Q_A. Fix a starting curve E_0, points P_A, Q_A, and a B-orientation $\mathfrak{B} = (G_1, G_2)$, Alice's public key consists of the codomain of her secret isogeny, together with the image of $\mathfrak{B}$ under her secret isogeny,

while Bob's public key includes the codomain of his secret B-oriented isogeny, together with the images of P_A and Q_A.

Since Alice is computing SIDH-like isogenies, the degree of her secret isogeny can be very smooth (concretely, this will be a power of two); while this reduces the size of the isogeny degree of one party, the degree of the other party needs to increase to guarantee sufficient security. Thus, the resulting prime p is generally of comparable size to that used in binSIDH and terSIDH. With this setup, we can take A to be considerably smaller and smoother than B; this means that Alice can be much more efficient in computing her isogenies. Not only that, but zero-knowledge proofs of knowledge of an A-isogeny, both ad-hoc [DDGZ22] and generic [CLL23], can be much more compact and efficient. More generally, computing SIDH-like isogenies allows one party to fully reuse the range of techniques developed for SIDH. The resulting schemes are described in Fig. 3.

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

KeyGen (Alice). Alice samples uniformly at random an integer $\alpha \in \mathbb{Z}/A\mathbb{Z}$ and computes $\phi_A : E_0 \to E_A$ of kernel $\langle P_A + [\alpha]Q_A \rangle$. Her secret key is $\mathsf{sk}_A = \alpha$ and her public key is the artificially B-oriented curve $\mathsf{pk}_A = (E_A, \phi_A(\mathfrak{B}))$.

KeyGen (Bob). Bob samples uniformly at random a vector $\mathbf{b}$ from $\{1, 2\}^t$ and computes the $\mathfrak{B}$-oriented isogeny $\phi_B : E_0 \to E_B$ of degree B defined by b. His secret key is $\mathsf{sk}_B = b$ and his public key is $\mathsf{pk}_B = (E_B, \phi_B(P_A), \phi_B(Q_A))$.

SharedKey (Alice). Upon receiving Bob's public key (E_B, R, S), Alice checks that $e_A(R, S) = e_A(P_A, Q_A)^B$, if not she aborts. She computes the isogeny $\phi'_A : E_B \to E_{BA}$ of kernel $\langle R + [\alpha]S \rangle$. Her shared key is $j(E_{BA})$.

SharedKey (Bob). Upon receiving $(E_A, \mathfrak{B}')$, Bob checks that $\mathfrak{B}'$ is an artificial B-orientation on E_A, if not he aborts. He computes the $\mathfrak{B}'$-oriented isogeny $\phi'_B : E_A \to E_{AB}$ of degree B defined by b. His shared key is $j(E_{AB})$.

Fig. 3: The $\mathsf{binSIDH}^{\mathsf{hyb}}$ protocol. A similar variant, based on terSIDH, can be obtained by changing the Bob's KeyGen algorithm to sample vectors from $\{0, 1, 2\}$.

6 Security analysis

In this section, we analyze the security of the proposed protocols, both binSIDH and terSIDH, as well as their hybrid variants $\mathsf{binSIDH}^{\mathsf{hyb}}$ and $\mathsf{terSIDH}^{\mathsf{hyb}}$.

We analyzed the hardness assumptions relative to artificial orientations in Section 3.2, which guarantees it is unfeasible for an attacker to recover a secret key from a public key. In particular, the hardness of Problem 5 guarantees the hardness of key-recovery attacks against binSIDH and terSIDH, while the hardness of Problem 6 and 7 protects $\mathsf{binSIDH}^{\mathsf{hyb}}$ and $\mathsf{terSIDH}^{\mathsf{hyb}}$ from key-recovery attacks. However, the security of the key-exchange protocols, as well as any other protocol built on those, depends on the hardness of a different problem, which we call the Artificially Oriented Computational Diffie-Hellman (AO-CDH) problem.

Problem 9 (AO-CDH). Let the notation be as in Fig. 1. Let $\phi_A : E_0 \to E_A$ be a $\mathfrak{A}$-isogeny, and $\phi_B : E_0 \to E_B$ be a $\mathfrak{B}$-isogeny. Given $(E_A, \phi_A(\mathfrak{B}))$ and $(E_B, \phi_B(\mathfrak{A}))$, compute $j(E_{AB})$, where E_{AB} is the codomain of the push-forward of ϕ_A under ϕ_B (or vice versa).

The problem, as stated, guarantees the security of terSIDH. We can easily obtain similar problems for the remaining protocols by requires that the isogenies have fixed degrees (binSIDH) or allowing one party to use unoriented isogenies (binSIDH$^{\mathsf{hyb}}$, terSIDH$^{\mathsf{hyb}}$). We can also consider a decisional variant of these problems, where given an additional j-invariant j', the problem asks to determine whether $j' = j(E_{AB})$. While the security of the proposed protocols does not depend on such decisional problems, advanced constructions based on these protocol might require such an assumption.

The relationship between these problems and those introduced in Section 3.2 is similar to that between the Computational Diffie-Hellman problem and the Discrete Logarithm problem, or between the Supersingular Computational Diffie-Hellman problem and the Computational Supersingular Isogeny problem [JD11]. While there is no reduction from the problems in Section 3.2 to Problem 9, it is likely that any attack that breaks the proposed protocols would need to efficiently solve the problems of Section 3.2.

Remark 10. In binSIDH and terSIDH, the two parties reveal the codomain of their secret isogenies, together with only the images of two disjoint cyclic subgroups. This is, in some sense, optimal, as it is the minimum amount of information needed for the other party to compute the push-forwards. Thus, if any major cryptanalytic breakthrough managed to break binSIDH and terSIDH, it seems likely that any possible SIDH-like construction would equally be broken, including the existing countermeasures against the SIDH attacks [FMP23].

Lastly, we analyze the choice of starting curve E_0 and its effects on the security of the protocols. In unbalanced variants of SIDH, the curve E_0 could be backdoored to enable key-recovery attacks [dQKL+21], while the M-SIDH and MD-SIDH [FMP23] are vulnerable to the SIDH attacks when the starting curve has small endomorphisms. In the key-exchange variant, this can be solved through a trusted setup [BCC+23]. However, none of these attacks seem to apply to our protocols. In binSIDH and terSIDH, the lack of torsion information prevents both the small-endomorphism attack from [FMP23] and the torsion-point attacks that were used in the backdoor attack, which means that the starting curve E_0 can be easily generated and can, for instance, be chosen to be the curve satisfying $j(E_0) = 1728$. In the case of binSIDH$^{\mathsf{hyb}}$ and terSIDH$^{\mathsf{hyb}}$, one party does reveal exact torsion information. However, the degree of the secret isogeny is much larger than the order of the points whose image is revealed, which prevents both types of attacks. Hence, even in binSIDH$^{\mathsf{hyb}}$ and terSIDH$^{\mathsf{hyb}}$, it is secure to start from a curve of known endomorphism ring, such as in the case when $j(E_0) = 1728$.

6.1 Adaptive security

SIDH has been known to be vulnerable to active adaptive attacks [GPST16,FP22], i.e. attacks where the target has a long-term static key and the attacker is a participant of the key exchange. In this section, we show how the proposed protocols are unfortunately similarly vulnerable to adaptive attacks.

In binSIDH$^{\mathsf{hyb}}$ and terSIDH$^{\mathsf{hyb}}$, one party computes SIDH-like isogenies. As such, they are vulnerable to exactly the same attacks that SIDH is. We can thus focus on active attacks against oriented isogenies, which covers the remaining cases.

Let us assume Alice is the target party, while Bob plays the role of the attacker; For simplicity, let us also assume we are in the case of binSIDH, where Alice's secret is the binary vector $\mathbf{a} \in \{1,2\}^t$. The case of terSIDH follows similarly.

Bob can use potentially malicious public keys and check whether both parties obtained the same shared secret. In other words, the attacker has access to the following oracle:

$$\mathcal{O}(E, \mathfrak{A}, j') = \begin{cases} \mathit{true} & \text{if } j(E/\langle \mathbf{a}, \mathfrak{A} \rangle) = j', \\ \mathit{false} & \text{otherwise.} \end{cases}$$

Write A, the order of the artificial orientation $\mathfrak{A}$, as $A = \prod_{i=1}^{t} p_i$. To target the i-th bit of the secret key $\mathbf{a}_i$, the attacker can honestly compute the curve E_B and the image orientation $\mathfrak{A} = (G_1, G_2)$ and write $G_j = H_j^1 \oplus \ldots \oplus H_j^t$ for $j \in \{1,2\}$, where each H_j^k has coprime order p_j. Then, if I_1^i is any cyclic subgroup of order p_i such that $I_1^i \cap H_1^i = \emptyset$, the attacker can define $\mathfrak{A}' = (G_1', G_2)$, where G' is the same subgroup as G with H_1^i replaced by I_1^i, i.e. $G_1' = H_1^1 \oplus I_1^i \oplus \ldots \oplus H_1^t$. The attacker can also obtain the j-invariant j_{AB} corresponding to the shared secret of an honest exchange and query the oracle $\mathcal{O}(E_B, \mathfrak{A}', j_{AB})$. If

the oracle returns *true*, the shared secret is unchanged: this means that modified subgroup did not affect the computations, and thus $\mathbf{a}_i = 2$. Otherwise, the modified subgroup did change the shared secret, and thus $\mathbf{a}_i = 1$.

The active attack against the proposed protocols is slightly more powerful than the GPST attack. It does not involve carefully crafted torsion points, and it allows to target any bit of the secret key without necessarily proceeding in order. In the PKE setting, one party can achieve long-term security with the use of the Fujsaki-Okamoto transform [FO97], while in the key exchange setting, it is possible to obtain active security for both parties, thus obtaining a non-interactive key exchange, by introducing a proof of public key correctness. For artifically oriented curves, this can be achieved by adapting the zero-knowledge proof of masked public keys from [Bas23] to work with independently scaled points.

7 Implementation

7.1 Parameter selection

Following the security analysis of Section 3.2, we generated parameter sets for the four proposed protocols at security levels $\lambda \in \{128, 192, 256\}$.

In binSIDH and terSIDH, both parties rely on oriented isogenies, and thus the degrees corresponding to both isogenies need to be quite large: in the case of binSIDH, at least the product of λ distinct primes. This is reduced to $\lambda/\log_3(2)$ for terSIDH, since each prime provides $\log_3(2)$ bits of security. To obtain a balanced trade-off between the two parties, we assign consecutive primes to different parties; in other words, the degree of Alice's isogenies is the product of t even-index primes, while Bob's isogenies degree is the product of t odd-index primes. Moreover, the isogeny degrees need to be coprime for the key exchange to be commutative, and thus the underlying prime necessarily needs to be larger than the product of the first 2λ in binSIDH ($2\lambda/\log_3(2)$ in terSIDH). The resulting parameter sets for binSIDH and terSIDH are summarized in Table 1, where we also list the corresponding public key sizes.

Remark 11 (Public-key compression). As in SIDH, public keys can be compressed by expressing the the torsion points with respect to a deterministically generated basis [CJL+17]. This requires three coefficients in SIDH since both points can be scaled by the same value without affecting the SIDH computations, which means that one of the four coefficients can be fixed to one. In our case, however, the two points that generate artificial orientations can be scaled independently: this means that the public keys of the proposed protocols can be compressed to only two coefficients.

The size of the primes and public keys of binSIDH and terSIDH is a stark improvement over those of the existing countermeasures M-SIDH and MD-SIDH [FMP23]. For instance, at $\lambda = 128$, the primes of binSIDH and terSIDH are 2.5× and 8.8× smaller than those in M-SIDH and MD-SIDH, respectively.[4] While terSIDH$^{\text{hyb}}$ requires larger parameters than terSIDH, binSIDH$^{\text{hyb}}$ manages to be smaller and more efficient than binSIDH. When compared to M-SIDH, the underlying prime in binSIDH$^{\text{hyb}}$ is 2.9× smaller.

7.2 Implementation results

We developed a proof-of-concept implementation of all four protocols in SageMath [The23], based on the Kummer Line library [Pop23] to estimate the running times of the proposed protocols[5]. We report the average running times on an Apple M1 PRO CPU in Table 2.

The results of Table 2 show that the ternary variants significantly outperforms the binary ones, especially at higher security levels. This is because binSIDH uses larger prime fields and larger-degree isogenies than terSIDH.[6] Moreover, terSIDH does not need to compute full-degree isogenies due to its

[4] Interestingly, in the terSIDH case, the variable-degree isogenies allow us to achieve smaller parameters, while in MD-SIDH, the variable-degree isogenies require larger parameters because of the information leakage due to pairing computations.

[5] The source code is available at https://github.com/binary-ternarySIDH/bin-terSIDH-SageMath

[6] The specific SageMath implementation of VeluSqrt [BDFLS20] that we rely on does not outperform Velu's formulae [Vél71] until the isogeny degree is extremely large. We thus expect a low-level implementation to significantly improve the computation times of high-degree isogenies, more so than for lower-degree ones.

			Alice				Bob			
	λ	$\log p$	t	$\mathcal{B}$	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{pk}_{\mathsf{cmp}}\rvert$	t	B	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{pk}_{\mathsf{cmp}}\rvert$
binSIDH	128	2421	134	2^{11}	1816	907	134	2^{11}	1816	909
	192	3710	192	2^{12}	2783	1390	192	2^{12}	2783	1392
	256	5201	256	2^{12}	3901	1949	256	2^{12}	3901	1950
terSIDH	128	1568	93	2^{11}	1176	587	93	2^{11}	1176	588
	192	2295	128	2^{11}	1722	860	128	2^{11}	1722	861
	256	3035	162	2^{12}	2277	1137	162	2^{12}	2277	1139
$\mathsf{binSIDH}^{\mathsf{hyb}}$	128	2004	1	2	1503	937	203	2^{11}	1503	565
	192	3126	1	2	2345	1465	296	2^{11}	2345	878
	256	4267	1	2	3201	2004	387	2^{12}	3201	1195
$\mathsf{terSIDH}^{\mathsf{hyb}}$	128	1532	1	2	1149	701	156	2^{10}	1149	447
	192	2373	1	2	1780	1089	226	2^{11}	1780	690
	256	3216	1	2	2412	1479	293	2^{11}	2412	932

Table 1: Parameters for binSIDH and terSIDH. The coloumn t reports the number of distinct primes dividing the degrees of Alice's and Bob's isogenies, while their smoothness bound is reported in the $\mathcal{B}$ column. The columns $\lvert\mathsf{pk}\rvert$ and $\lvert\mathsf{pk}_{\mathsf{cmp}}\rvert$ reports the size of the public keys of both parties, respectively uncompressed and compressed.

varying-degree nature: it is thus likely that the benefits of this are reduced in a constant-time optimization. Nonetheless, the results of terSIDH are encouraging. At security $\lambda = 128$, the SharedKey computations take around 1.4 seconds, while key generation (which is run less often) requires about twos seconds. The current implementation is only a proof of concept in a high-level language: we can thus expect it to be several times faster once optimally implemented in a low-level language. Despite the lack of optimizations, the current implementation already outperforms optimized implementations of CSIDH with parameters sufficiently large to guarantee post-quantum security [CSCDJRH22], which require between 2.8 and 5.8 seconds to compute a group action at security level one.[7] Similarly, our proof-of-concept implementation outperforms the PoC implementation of FESTA [BMP23], which is based on the same SageMath library and takes 3.5 seconds to encrypt and 10.1 seconds to decrypt. It is thus mostly likely that terSIDH provides the most efficient key exchange and encryption protocol among all isogeny-based protocols.

Moreover, the results of the hybrid variants show that it is possible to have very low running times for one party, at the cost of a slight increase in the running times of the other party. The hybrid variants significantly reduce the overall running time of a complete key exchange.

8 Conclusion

In this work, we introduced artificial orientations, and proposed two new protocols, binSIDH and terSIDH, that translate the SIDH key exchange to the artificially oriented isogeny setting. This allows us to develop two protocols that are resistant against the SIDH attacks, while also achieving significantly smaller parameters than the previously proposed countermeasures. We also proposed $\mathsf{binSIDH}^{\mathsf{hyb}}$ and $\mathsf{terSIDH}^{\mathsf{hyb}}$, hybrid variants of binSIDH and terSIDH respectively, that allow one party to have very short and efficient isogenies. To validate the concrete efficiency of the protocols, we developed a proof-of-concept implementation. Despite being far from optimal, it already outperforms existing implementations of other isogeny-based encryption protocols, which suggests that optimized implementations of terSIDH and its hybrid variant might have practical running times.

In future work, we are interested in developing efficient and optimized implementations of binSIDH and terSIDH to accurately measure their running times. Moreover, this work opens up new possibilities that were previously closed by the SIDH attacks. In particular, it is interesting to assess the impact of

[7] Note, however, that these computations are constat-time, and that CSIDH does not require the Fujisaki-Okamoto [FO99] to obtain IND-CCA security.

			Timings (s)			
	λ	$\log p$	KeyGen_A	KeyGen_B	$\mathsf{SharedKey}_A$	$\mathsf{SharedKey}_B$
binSIDH	128	2421	13.69	13.86	9.40	9.46
	192	3710	48.69	49.36	27.39	27.81
	256	5201	140.79	140.57	94.13	95.67
terSIDH	128	1570	2.09	2.01	1.41	1.34
	192	2297	6.84	6.83	4.50	4.39
	256	3039	15.68	16.03	10.00	10.35
$\mathsf{binSIDH}^{\mathsf{hyb}}$	128	2004	0.23	14.33	0.22	10.66
	192	3126	0.62	56.77	0.61	42.85
	256	4267	1.41	157.58	1.34	117.07
$\mathsf{terSIDH}^{\mathsf{hyb}}$	128	1532	0.16	3.21	0.16	1.96
	192	2373	0.47	13.44	0.44	10.01
	256	3216	0.94	34.66	0.90	23.57

Table 2: Execution times in seconds of the SageMath proof-of-concept implementation. Since it is a PoC in a high-level language, we expect an optimized implementation of the same protocols to be several times more efficient.

the proposed protocols on the SIDH-based constructions, such as the round-optimal OPRF construction by Basso [Bas23], where we expect binSIDH and terSIDH to have a significant impact in reducing prime size and computational costs.

Acknowlegements. The first author has been supported in part by EPSRC via grant EP/R012288/1, under the RISE (http://www.ukrise.org) programme.

References

ACC+19. Gora Adj, Daniel Cervantes-Vázquez, Jesús-Javier Chi-Domínguez, Alfred Menezes, and Francisco Rodríguez-Henríquez. On the cost of computing isogenies between supersingular elliptic curves. In Carlos Cid and Michael J. Jacobson Jr:, editors, *SAC 2018*, volume 11349 of *LNCS*, pages 322–343. Springer, Heidelberg, August 2019. doi:10.1007/978-3-030-10970-7_15.

Bas23. Andrea Basso. A post-quantum round-optimal oblivious PRF from isogenies. Cryptology ePrint Archive, Report 2023/225, 2023. https://eprint.iacr.org/2023/225.

BBC+21. Gustavo Banegas, Daniel J. Bernstein, Fabio Campos, Tung Chou, Tanja Lange, Michael Meyer, Benjamin Smith, and Jana Sotáková. CTIDH: faster constant-time CSIDH. *IACR TCHES*, 2021(4):351–387, 2021. https://tches.iacr.org/index.php/TCHES/article/view/9069. doi:10.46586/tches.v2021.i4.351-387.

BCC+23. Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 405–437, Cham, 2023. Springer Nature Switzerland.

BDFLS20. Daniel J Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020. doi:10.2140/obs.2020.4.39.

BFGP23. Ward Beullens, Luca De Feo, Steven D. Galbraith, and Christophe Petit. Proving knowledge of isogenies – a survey. Cryptology ePrint Archive, Paper 2023/671, 2023. https://eprint.iacr.org/2023/671. URL: https://eprint.iacr.org/2023/671.

BKM+21. Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit, and Antonio Sanso. Cryptanalysis of an oblivious PRF from supersingular isogenies. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 160–184. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92062-3_6.

BMP23. Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: Fast Encryption from Supersingular Torsion Attacks. Cryptology ePrint Archive, Paper 2023/660, 2023. https://eprint.iacr.org/2023/660. URL: https://eprint.iacr.org/2023/660.

CD23. Wouter Castryck and Thomas Decru. An Efficient Key Recovery Attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 423–447, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-30589-4_15.

CHM+23. Wouter Castryck, Marc Houben, Simon-Philipp Merz, Marzio Mula, Sam van Buuren, and Frederik Vercauteren. Weak instances of class group action based cryptography via self-pairings. Cryptology ePrint Archive, Paper 2023/549, 2023. https://eprint.iacr.org/2023/549. URL: https://eprint.iacr.org/2023/549.

CJL+17. Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 679–706. Springer, Heidelberg, April / May 2017. doi:10.1007/978-3-319-56620-7_24.

CJS14. Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Math. Cryptol.*, 8(1):1–29, 2014. doi:10.1515/jmc-2012-0016.

CK20. Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. Cryptology ePrint Archive, Report 2020/985, 2020. https://eprint.iacr.org/2020/985.

CLL23. Kelong Cong, Yi-Fu Lai, and Shai Levin. Efficient isogeny proofs using generic techniques. Cryptology ePrint Archive, Report 2023/037, 2023. https://eprint.iacr.org/2023/037.

CLM+18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018. doi:10.1007/978-3-030-03332-3_15.

Cou06. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291.

CSCDJRH22. Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, 12(3):349–368, September 2022. doi:10.1007/s13389-021-00271-w.

CV23. Wouter Castryck and Frederik Vercauteren. A polynomial time attack on instances of M-SIDH and FESTA. private communication, 2023.

DD21. Pierrick Dartois and Luca De Feo. On the security of OSIDH. Cryptology ePrint Archive, Report 2021/1681, 2021. https://eprint.iacr.org/2021/1681.

DdF+21. Luca De Feo, Cyprien de Saint Guilhem, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Christophe Petit, Javier Silva, and Benjamin Wesolowski. Séta: Supersingular encryption from torsion attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 249–278. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92068-5_9.

DDGZ22. Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 310–339. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22966-4_11.

DFFK+23. Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. Scallop: Scaling the csi-fish. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography – PKC 2023*, pages 345–375, Cham, 2023. Springer Nature Switzerland.

DG16. Christina Delfs and Steven D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, Feb 2016. doi:10.1007/s10623-014-0010-1.

DKL+20. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 64–93. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64837-4_3.

dQKL+21. Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Improved torsion-point attacks on SIDH variants. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 432–470, Virtual Event, August 2021. Springer, Heidelberg. doi:10.1007/978-3-030-84252-9_15.

FMP23. Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: Countering SIDH Attacks by Masking Information. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 282–309, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-30589-4_10.

FO97. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 16–30. Springer, Heidelberg, August 1997. doi:10.1007/BFb0052225.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Heidelberg, August 1999. doi:10.1007/3-540-48405-1_34.

FP22. Tako Boris Fouotsa and Christophe Petit. A new adaptive attack on SIDH. In Steven D. Galbraith, editor, *CT-RSA 2022*, volume 13161 of *LNCS*, pages 322–344. Springer, Heidelberg, March 2022. doi:10.1007/978-3-030-95312-6_14.

GPST16. Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_3.

JAC+20. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. SIKE. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

JD11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011. doi:10.1007/978-3-642-25405-5_2.

Kan97. Ernst Kani. The number of curves of genus two with elliptic differentials., 1997. URL: https://doi.org/10.1515/crll.1997.485.93.

Ler22. Antonin Leroux. A new isogeny representation and applications to cryptography. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 3–35. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22966-4_1.

MMP+23. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A Direct Key Recovery Attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 448–471, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-30589-4_16.

Pei20. Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 463–492. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_16.

Pet17. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 330–353. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70697-9_12.

Pop23. Giacomo Pope. Kummer Isogeny SageMath Library. https://github.com/jack4818/KummerIsogeny, 2023.

Rob23. Damien Robert. Breaking SIDH in Polynomial Time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 472–503, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-30589-4_17.

Sil09. Joseph H Silverman. *The Arithmetic of Elliptic Curves*, volume 106. Springer Science & Business Media, 2009.

The23. The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.8)*, 2023. URL: https://www.sagemath.org.

Vél71. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris, Séries A*, 273:305–347, 1971.

vW99. Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, January 1999. doi:10.1007/PL00003816.