Research Article

David Urbanik* and David Jao

New Techniques for SIDH-based NIKE

https://doi.org/???
Received ; accepted

Abstract: We consider the problem of producing an efficient, practical, quantum-resistant non-interactive key exchange (NIKE) protocol based on Supersingular Isogeny Diffie-Hellman (SIDH). An attack of Galbraith, Petit, Shani and Ti rules out the use of naïve forms of the SIDH construction for this application, as they showed that an adversary can recover private key information when supplying an honest party with malformed public keys. Subsequently, Azarderakhsh, Jao and Leonardi presented a method for overcoming this attack using multiple instances of the SIDH protocol, but which increases the costs associated with performing a key exchange by factors of up to several thousand at typical security levels. In this paper, we present two new techniques to reduce the cost of SIDH-based NIKE, with various possible tradeoffs between key size and computational cost.

Keywords: isogeny-based cryptography, SIDH, active attacks, NIKE, automorphisms

1 Introduction

The Supersingular Isogeny Diffie-Hellman (SIDH) protocol [10, 15] is a promising candidate for quantum-resistant key exchange. The protocol functions analogously to classical Diffie-Hellman, but using supersingular elliptic curves and cyclic subgroups instead of group elements and exponents. That is, one starts with a "base curve" E, Alice and Bob pick private cyclic subgroups $A \subset E$ and $B \subset E$, and they each compute the "quotient curves" E/A and E/B for use in their respective public keys. To facilitate computation of the shared secret, Alice and Bob's public keys also contain additional information about the quotient maps $\phi_A : E \to E/A$ and $\phi_B : E \to E/B$. Using this information, Alice and Bob then complete the protocol by computing a shared secret derived from an isomorphism invariant of the curve $E/(A + B)$. SIDH security is based on a special case of the supersingular isogeny problem, which was first proposed for use in cryptography in [6]; as explained in [6, §5.3.1], this problem in turn was first introduced in [13]. We refer to [8] for a discussion of these hardness assumptions and their historical context.

Given the similar dataflow to the ordinary Diffie-Hellman protocol, it was at one time hoped that the SIDH construction would be a promising candidate for a static-static or non-interactive key exchange (NIKE) protocol. However, Galbraith, Petit, Shani, and Ti [14] showed that it was possible to use the additional information about ϕ_A and ϕ_B provided in the public keys to perform an active attack capable of recovering Alice and Bob's private keys. Prior work of Azarderakhsh *et al.* [2] shows that one can prevent the GPST attack and obtain a NIKE from SIDH by applying an expensive generic transformation, as follows. Suppose that Alice generates α public keys and Bob generates β public keys, where α and β are positive integers. Then Alice and Bob may perform a total of $\alpha\beta$ key exchanges — one for each pair of public keys — and take their shared secret to be a hash of the concatenation of all of them. If a malicious attacker (say, Bob) presents an honest Alice with a malformed public key, then a total of α secret curves are potentially affected. To extract information about Alice's public keys from the hash computed by Alice, the attacker must know what input produced the hash, and so must search through all possible modifications of the α affected secret keys and try the possible

***Corresponding Author: David Urbanik:** Department of Combinatorics and Optimization University of Waterloo, ON, Canada; Email: dburbani@uwaterloo.ca
David Jao: Department of Combinatorics and Optimization University of Waterloo, ON, Canada; Email: djao@uwaterloo.ca

hash values until they obtain a collision. If each secret curve can take on r possible values (say all occurring with equal probability, for simplicity, although the situation in practice is in fact more complicated) then the attacker must search through a space of r^α possibilities, which requires exponential work if α is taken to be large enough. In [2], this construction is referred to as k-SIDH.

For 128-bit post-quantum security, Azarderakhsh *et al.* recommend $\alpha = 113$ and $\beta = 94$ for standard SIDH parameters (the asymmetry arises because ϕ_A and ϕ_B are different), resulting in a total of $113 \cdot 94 = 10622$ key exchanges. In general, key size is proportional to α and β and scales linearly with security level, and computational cost is proportional to $\alpha\beta$ and scales quadratically with security level.

In this paper, we significantly improve this state of affairs in two ways. The first approach is to modify the k-SIDH construction using extra automorphisms in a way that greatly increases the likelihood of obtaining malformed secret keys, allowing us to decrease the values of α and β. Using this approach, the computational cost remains quadratic, but with much smaller constants. The second approach is to devise new zero-knowledge proofs, based in part on our first improvement, to validate SIDH public keys and thus resist GPST-style attacks. Our second approach has linear cost overhead and hence is asymptotocally more cost-efficient, but requires larger (though still linearly scaling) key sizes.

We believe that our contributions likely have additional applications other than NIKE, although we do not pursue them here. Our first contribution, using non-trivial automorphisms to produce non-isomorphic isogenies between isomorphic curves, might be useful for performance improvements, similar to how some variants of GLV use extra low-degree endomorphisms to speed up point multiplication [17]. Our second contribution, on zero-knowledge proofs of validity for SIDH keys, may be useful for other authentication protocols such as digital signatures.

1.1 Related work

The recently proposed CSIDH protocol [5] is an alternative isogeny-based cryptosystem which seems to be especially well-suited to the NIKE setting. Under the original parameter choices and security analysis in [5], CSIDH-based NIKE is both faster and more compact than SIDH-based NIKE for a given security level, even with our improvements. However, subsequent analyses [3, 4] indicate that CSIDH may not be as secure as originally estimated. Hence, we believe our improvements are still worth proposing, since they could lead to further improvements which might make SIDH competitive in this setting. In any case, accurate information about the cost overhead of SIDH-based NIKE is necessary for a fair comparison of current state of the art NIKE protocols under SIDH vs. CSIDH.

We are not aware of any other papers containing an extended discussion of NIKE protocols in the post-quantum setting, though some protocols believed to be quantum-resistant have been analyzed in the classical setting [16, Theorem 1].

2 Extra Secrets from Automorphisms

In this section, we develop some mathematical preliminaries for changes we will make to the SIDH construction. These changes allow us, in certain situations, to agree on multiple non-isomorphic shared secret curves from a single public key pair. We believe these techniques are of independent interest, which is why we have isolated them in their own section.

We begin by recalling the SIDH construction. Let ℓ_A and ℓ_B be small primes, let e_A and e_B be exponents such that $\log(\ell_A^{e_A}) \approx \log(\ell_B^{e_B})$, and let f be a small cofactor such that $p = f\ell_A^{e_A}\ell_B^{e_B} \pm 1$ is prime. Set $n_A = \ell_A^{e_A}$ and $n_B = \ell_B^{e_B}$. Then it is possible to find a supersingular elliptic curve E such that $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(fn_An_B)\mathbb{Z}) \times (\mathbb{Z}/(fn_An_B)\mathbb{Z})$. In particular, the entire n_A and n_B torsion subgroups are defined over $\mathbb{F}_{p^2}$, and so Alice and Bob may select their respective secrets A and B to be cyclic subgroups of $E[n_A]$ and $E[n_B]$, respectively. They then take their public keys to be the information of $(E/A, \phi_A|_{E[n_B]})$ and $(E/B, \phi_B|_{E[n_A]})$. The shared secret is then

an isomorphism invariant of $E/(A+B)$. In order for Alice to compute this shared secret, she must compute the quotient $(E/B)/\phi_B(A) \cong E/(A+B)$, for which it suffices for her to know $\phi_B|_{E[n_A]}$. Since $\phi_B|_{E[n_A]}$ is a $\mathbb{Z}$-linear map from $E[n_A]$ to $(E/B)[n_A]$, it can be specified by giving its values $\phi_B(P_A)$ and $\phi_B(Q_A)$ on a basis P_A, Q_A for $E[n_A]$. Bob computes the shared secret similarly.

Consider now an elliptic curve E defined over a field of characteristic p not equal to 2 or 3. If $\eta\colon E \to E$ is an automorphism of E, that is, an invertible map of curves which is also a group homomorphism, then generically there are only two possibilities for η, as follows: either $\eta(P) = P$ is the identity map, or $\eta(P) = -P$ is the negation map. Two exceptional cases can occur when E is a curve isomorphic to $E_0 : y^2 = x^3 + 1$ or $E_{1728} : y^2 = x^3 + x$, that is, when its j-invariant is equal to either 0 or 1728. In the first case, one can have a nontrivial automorphism of order six given by $\eta_6\colon (x, y) \mapsto (\zeta_3 x, -y)$, where ζ_3 is a non-trivial third root of unity, and in the second case one can have a nontrivial automorphism of order four given by $\eta_4\colon (x, y) \mapsto (-x, iy)$.

The existence of these automorphisms has consequences for isogenies emanating from E. For instance, consider the case where $\eta_4\colon E_{1728} \to E_{1728}$ is a non-trivial automorphism of order four. If $G \subset E_{1728}$ is a subgroup, then one obtains a second subgroup $\eta_4(G)$ of E_{1728} which is usually distinct from G. (The cyclic subgroups of size N where it is not distinct correspond exactly to the ramification points of the classical modular curve $X_0(N)$ lying over $j = 1728$.) If $\phi_G\colon E_{1728} \to E_{1728}/G$ is an isogeny associated to the quotient E_{1728}/G, then the map $\phi_G \circ \eta_4^{-1}\colon E_{1728} \to E_{1728}/G$ has kernel $\eta_4(G)$, and hence its image E_{1728}/G is isomorphic to $E_{1728}/\eta_4(G)$.

If we consider this setup in the context of the SIDH construction with $E = E_{1728}$ and $A = G$, then we have that Alice's public key $(E/A,\ \phi_A|_{E[n_B]})$ is in a certain sense "degenerate," in the sense that there is an additional associated public key $(E/\eta_4(A),\ \phi_A|_{E[n_B]} \circ \eta_4^{-1})$ which has the same target curve (since $E/A \cong E/\eta_4(A)$), but as an isogeny is not isomorphic to ϕ_A. (For a detailed discussion of this unusual situation, in which two non-isomorphic isogenies have isomorphic domains and codomains, we refer to [1].) One may easily compute the associated torsion information for the other isogeny by precomposing $\phi_A|_{E[n_B]}$ with η_4^{-1}. This means that each public key generated from $j = 1728$ actually corresponds to *two* public keys (with isomorphic curves but different torsion point information), and so a public key pair can be thought of (naïvely) as determining the four secret curves $E/(A+B)$, $E/(\eta_4(A)+B)$, $E/(A+\eta_4(B))$ and $E/(\eta_4(A)+\eta_4(B))$. However, these four curves comprising the four shared secrets generically[1] only represent two distinct isomorphism classes. This fact follows because the quotient maps $E \to E/(A+B)$ and $E \to E/(\eta_4(A)+\eta_4(B))$ have kernels which differ by an application of η_4, and so are isomorphic by the preceding reasoning (take $G = A + B$). The analogous fact is is true for the other pair. Nevertheless, despite this degeneracy, one still obtains two secret curves (up to isomorphism) from a single public key pair using $E = E_{1728}$ as the base curve.

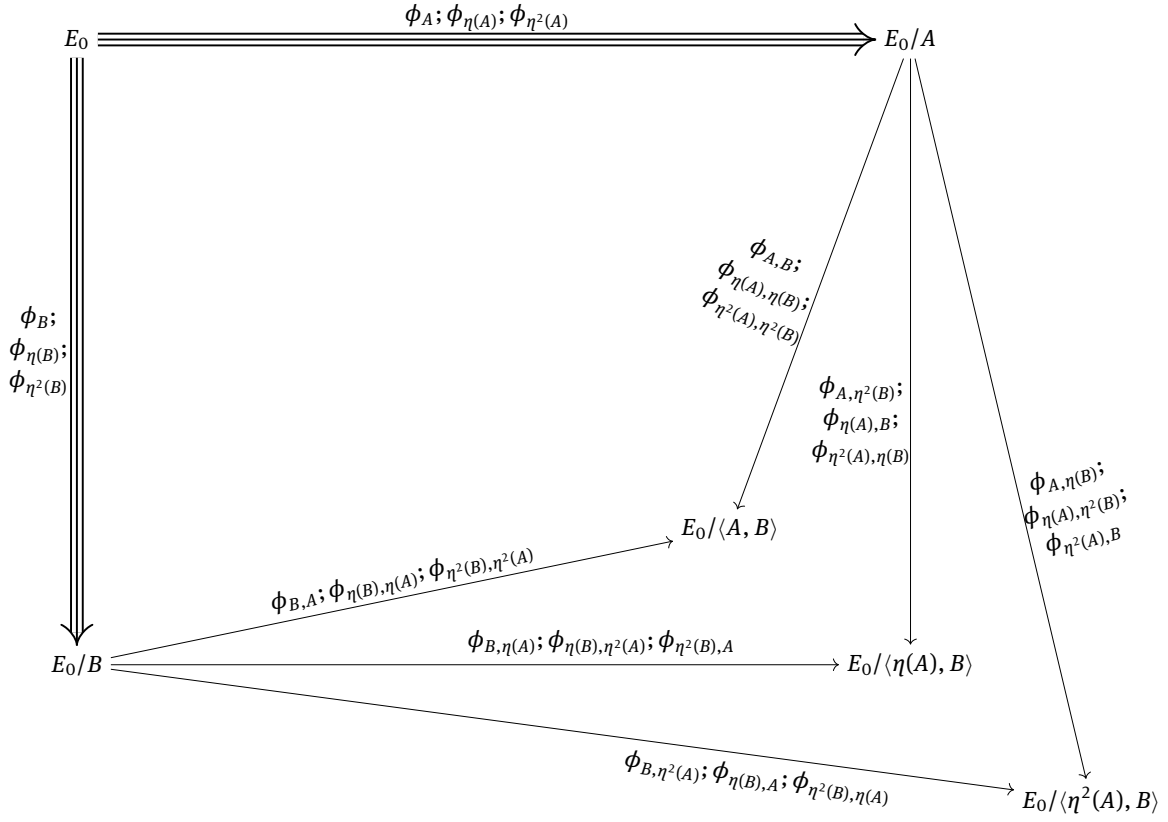
One can do even better by using $\eta_6\colon E_0 \to E_0$, of order six. This time, each public key is thrice-degenerate, resulting in a total of nine shared secrets which represent three generically distinct isomorphism classes, namely:

$$\begin{aligned} E/(A+B) &\cong E/(\eta_6(A)+\eta_6(B)) \cong E/(\eta_6^2(A)+\eta_6^2(B)) \\ E/(A+\eta_6(B)) &\cong E/(\eta_6(A)+\eta_6^2(B)) \cong E/(\eta_6^2(A)+B) \\ E/(A+\eta_6^2(B)) &\cong E/(\eta_6(A)+B) \cong E/(\eta_6^2(A)+\eta_6(B)). \end{aligned}$$

Since this case is the case of primary interest in what follows, we diagram it here. The subscripts on the initial arrows (leading out from the base curve) denote the kernel of the map, and the subscripts on the secondary arrows denote the isogeny obtained by quotienting out the second subscript after applying the isogeny determined by the first. The secondary arrows have multiple labels because the same isogeny arises

1 We say "generically distinct" because if the isogeny class is sufficiently small, these curves could all be isomorphic "by accident" (for instance, when $p < 11$, there is only one supersingular elliptic curve up to isomorphism). However, for exponentially large p this is not a problem, since the probability of collisions is heuristically $O(p^{-1})$ for each of $j = 0$ and 1728.

in multiple ways, and the triple arrows have multiple labels because there are actually multiple isogenies.



3 The Action of Automorphisms on Private Keys

The observations in the previous section allow us to develop new strategies to limit the effectiveness of GPST and similar active attacks. To understand how these strategies work, we provide a description of the GPST attack using a morphism-based framework. The GPST attack works by modifying the values of $\phi_B(P_A)$ and $\phi_B(Q_A)$ presented to Alice, and such a modification can be viewed as giving Alice the information of $L \circ \phi_B\big|_{E[n_A]}$, where L is a linear automorphism of $(E/B)[n_A]$ chosen by the attacker. When Alice computes her secret, she will then compute $(E/B)/L(\phi_B(A))$. The map L can be chosen so that the isomorphism class of $(E/B)/L(\phi_B(A))$ is always "close" to the isomorphism class of $E/(A+B)$ (in the sense of being isogenous to $E/(A+B)$ by an isogeny of degree ℓ_A), and by computing $E/(A+B)$ and finding the location of $(E/B)/L(\phi_B(A))$ relative to $E/(A+B)$, the attacker can find out information about A. Specifically, the attacker can exhaustively enumerate all of the $\ell_A + 1$ curves which are ℓ_A-isogenous to $E/(A+B)$, and try all of their j-invariants successively as the putative output of a shared secret computation with Alice. Depending on which of these guesses matches Alice's modified shared secret computation, the attacker then knows exactly which of the curves ℓ_A-isogenous to $E/(A+B)$ lies on the ℓ_A-isogeny path of length e_A between E/B and $E/(A+B)$, and this partial information about the isogeny path corresponds directly to partial information about Alice's secret key.

We now suppose $\eta_6 : E_0 \to E_0$ is a non-trivial automorphism of order six. The idea is that if the attacker gives false information for the map $\phi_A\big|_{E_0[n_B]}$ in the public key $(E_0/A,\ \phi_A\big|_{E_0[n_B]})$, then this modification not only affects the computation of the secret $E_0/(A+B)$ but also that of the associated secrets $E_0/(A+\eta_6(B))$ and $E_0/(A+\eta_6^2(B))$. One can show that it is possible to choose private keys which guarantee that at least two (and typically three) of these computations will fail under GPST-type attacks. This line of defense increases the size of the attacker's search space, since the attacker now essentially has to guess the result of three modified shared secret computations simultaneously instead of just one. The increase in attack difficulty in

turn yields an improvement in performance for a non-interactive exchange at the same security level. The same observation also leads to a natural non-interactive proof mechanism for validating SIDH public keys (cf. Section 5).

Lemma 3.1. *Suppose that $P, Q \in E[\ell^e]$ are points of order ℓ^e, where ℓ^e is one of $\ell_A^{e_A}$ or $\ell_B^{e_B}$. Then P and Q are independent if and only if $[\ell^{e-1}]P$ and $[\ell^{e-1}]Q$ are independent.*

Note that $\eta_6^3 = -1$ (as automorphisms), and $\eta_6^2 = \eta_6 - 1$. For any positive integer n, we will say that two points $P, Q \in E[n]$ are *independent* if $\langle P\rangle \cap \langle Q\rangle = \langle \mathcal{O}_E\rangle$ (that is, the intersection of the subgroups they generate is trivial).

Proof. If $[\ell^{e-1}]P$ and $[\ell^{e-1}]Q$ are not independent, then $\langle[\ell^{e-1}]P\rangle = \langle[\ell^{e-1}]Q\rangle$, and this subgroup is a non-trivial subgroup of $\langle P\rangle$ and $\langle Q\rangle$. It follows that P and Q are not independent. Hence, if P and Q are independent, then $[\ell^{e-1}]P$ and $[\ell^{e-1}]Q$ must be independent.

Conversely, suppose P and Q are dependent. We know $\langle P\rangle$ and $\langle Q\rangle$ are cyclic groups of prime power order, and hence their lattice of subgroups under inclusion forms a single chain. The group $\langle P\rangle \cap \langle Q\rangle$ is non-trivial, and hence contains the ℓ-order subgroups of both $\langle P\rangle$ and $\langle Q\rangle$, which are precisely the subgroups $\langle[\ell^{e-1}]P\rangle$ and $\langle[\ell^{e-1}]Q\rangle$. But since $\langle P\rangle \cap \langle Q\rangle$ is prime-power cyclic, it can only contain one ℓ-order subgroup, and so $\langle[\ell^{e-1}]P\rangle = \langle[\ell^{e-1}]Q\rangle$, which completes the proof. □

Lemma 3.2. *Suppose that ℓ is not equal to the characteristic p of the field of definition of E_0, and $P \in E_0[\ell^e]$ is a random full order point. With probability at least $1 - (2\ell - 2)/(\ell^2 - 1)$, every pair of elements from the set $\{P, \eta_6(P), \eta_6^2(P)\}$ is independent.*

Proof. Applying the previous lemma, it suffices to determine when pairs of elements in the set $\{[\ell^{e-1}]P, [\ell^{e-1}]\eta_6(P), [\ell^{e-1}]\eta_6^2(P)\}$ are independent. Any pair of elements from this set is independent precisely when one element is not a scalar multiple of the other. In particular, if this property holds for one pair, then it holds for all of them by the linearity of η_6. So it suffices to determine the probability that P is an eigenvector of η_6. Since ℓ is not equal to the characteristic of the field of definition of E_0, Deuring's lifting theorem [11, p. 203] implies that η_6 does not restrict to a scalar multiplication, so it has two distinct eigenvalues. Hence each one-dimensional eigenspace contains at most $\ell - 1$ non-zero elements, so the probability of P not being an eigenvector of η_6 is at least $1 - 2(\ell - 1)/(\ell^2 - 1) = \frac{\ell-1}{\ell+1}$. □

4 Using multiple secrets in key exchange

We recall how the GPST attack works. Suppose Alice is an honest participant with public key $(E/A, \phi_A|_{E[n_B]})$, where A has order $n_A = \ell_A^{e_A}$, which for brevity we abbreviate ℓ^e. Bob generates an honest public key $(E/B, \phi_B|_{E[n_A]})$ and then alters $\phi_B|_{E[n_A]}$ by pre-composing this linear map with a matrix such as $\left[\begin{smallmatrix} 1 & \ell^{e-1} \\ 0 & 1 \end{smallmatrix}\right]$ (with respect to a basis $\{P_A, Q_A\}$ of $E[n_A]$). This alteration has the effect of changing Alice's shared secret computation if and only if a certain ℓ-torsion point lies in A. We assume that Bob can interact with Alice to distinguish failed key exchanges from correct key exchanges. By repeating this process with different matrices, Bob can determine which ℓ-torsion points lie in A, and then iteratively do the same for ℓ^2-torsion, ℓ^3-torsion, etc. until Bob knows A.

The k-SIDH proposal [2] thwarts the GPST attack by having Alice and Bob instantiate α and β public keys respectively and performing $\alpha\beta$ key exchanges. The main idea is that Alice's α different secret keys will not have any ℓ-torsion point in common. Therefore, any GPST-style alteration that Bob makes will cause at least one of the $\alpha\beta$ key exchanges to fail, yielding no information about Alice's secret key. Indeed, even in the case $\alpha = \beta = 2$, one can already arrange for Alice's two secret keys to be linearly disjoint, so that any alterations by Bob will cause one or more of the four shared secret computations to fail. However, k-SIDH with $\alpha = \beta = 2$ is not enough to defend against a more sophisticated attack, in which Bob guesses which incorrect shared

secrets Alice will compute, and then forges his own shared secret computation to match what he guesses Alice will compute. As shown in [2], the probability of a successful guess is $1/(\ell(\ell+1))$; briefly speaking, Bob must compute the correct ℓ^e-isogeny, backtrack by one ℓ-isogeny ($\ell+1$ possibilities), and then move forward by one ℓ-isogeny (ℓ possibilities, since we exclude the one ℓ-isogeny that would undo the previous backtrack). Although SIDH is typically instantiated using $\ell = 2$ or $\ell = 3$ for efficiency, larger values of ℓ provide better defense against this type of attack. Our improvements below benefit even more from larger ℓ, and accordingly in what follows we propose the use of $\ell = 11$ or $\ell = 13$ as a good compromise between performance and security.

We now explain how to use multiple secrets to help k-SIDH better defend against the GPST attack. Suppose we use E_0 with j-invariant $j = 0$ for our base curve. For simplicity we assume $E[n_A]$ has basis $\{P, \eta_6(P)\}$ and that Alice's secret key is of the form $Q = \gamma P + \eta_6(P)$ (we remark that most published implementations of SIDH, such as [7], use keys of this form). Each round of the key exchange then produces three secret keys. These keys are related: if the kernel of Alice's original secret isogeny is generated by $Q = \gamma P + \eta_6(P)$, then the other two kernels will be generated by $\eta_6(Q) = -P + (\gamma+1)\eta_6(P)$ and $\eta_6^2(Q) = -(\gamma+1)P + \gamma\eta_6(P)$. Applying Lemma 3.2 to Q, we find that the elements $\{Q, \eta_6(Q), \eta_6^2(Q)\}$ are pairwise independent with probability $\frac{\ell-1}{\ell+1}$, and of course Alice could simply choose Q so that this property holds. Assuming it does, any GPST-style attack matrix will cause at least two of the resulting shared secret computations to be wrong, since a GPST matrix M is upper-triangular with one eigenvector, which can only overlap one of $\{Q, \eta_6(Q), \eta_6^2(Q)\}$; any element of this set which does not lie in an eigenspace of M will generate a kernel which is perturbed by M, resulting in an incorrect shared secret computation. Furthermore, with high probability (namely, $\frac{\ell+1-3}{\ell+1} = \frac{\ell-2}{\ell+1}$), all three shared secret computations will be wrong; we find this probability by observing that $\{Q, \eta_6(Q), \eta_6^2(Q)\}$ defines three lines in $E[n_A]$ and that the eigenvector of the GPST matrix avoids all three with probability $\frac{\ell+1-3}{\ell+1}$. This refinement therefore prevents the simple version of the GPST attack in which the adversary submits altered public keys and probes for correctness in the shared secret computation.

Consider now the "sophisticated" version of the GPST attack in which the adversary tries to guess which incorrect shared secrets Alice will compute. Under a naive estimate, typically three of the shared secrets will be wrong, and the number of possible wrong answers for each shared secret is $\ell(\ell+1)$. The attacker then has to search through a space of $\Omega((\ell(\ell+1))^3)$ possibilities. If Alice has α public keys, the cost is therefore $\Omega((\ell(\ell+1))^{3\alpha}) \approx \ell^{6\alpha}$, and so setting $256 \approx \lg(\ell^{3\alpha}(\ell+1)^{3\alpha})$ (where 256 is required to resist Grover's algorithm, but 128 can be chosen for security against classical attacks), we get $\alpha \approx 12$ for the prime $\ell = 11$.

Unfortunately, the naïve estimate above overestimates security. The reason is that the "incorrectness" of the three shared secrets is not independent: the errors are correlated, and the attacker can exploit this correlation. Specifically, an attacker can start from E_0/A and compute all of the $\ell+1$ possible ℓ-isogenies starting from E_0/A. Of these, exactly one ℓ-isogeny will have codomain equal to the correct curve, namely the elliptic curve lying along the ℓ^e-isogeny path from E_0 to E_0/A. The attacker does not know which curve is correct, but can guess the correct curve with probability $1/(\ell+1)$. Having guessed the correct curve E', the attacker can now compute the images B_1, B_2, B_3 of $B, \eta_6(B), \eta_6^2(B)$ in E' under the isogeny $E_0 \to E'$, and then the three curves E'/B_i, for $i = 1, 2, 3$. Each of these three curves now admits $\ell+1$ possible ℓ-isogenies, of which one will land in the correct curve $E/\langle A, B\rangle$, and the others will correspond to possible incorrect secrets that Alice might compute. The probability of guessing all three incorrect secrets successfully is thus $1/(\ell+1)^4$, or alternatively $1/(\ell^3 \cdot (\ell+1))$ if we assume that none of the three is computed correctly by Alice. As far as we know, there is no better way to guess, although we can only prove optimality by introducing an additional assumption contrived exactly for this purpose. If we assume that there is no better way, then the actual cost of blindly searching for Alice's incorrect shared secret values is $\Omega(\ell^{3\alpha}(\ell+1)^\alpha) \approx \ell^{4\alpha}$, which increases the requirements for α by a factor of $3/2$. For 256-bit security and $\ell \approx 11$, we need $\alpha \approx 18$ in order to obtain $256 \approx \lg(\ell^{3\alpha}(\ell+1)^\alpha)$.

5 NIZK-based SIDH key validation

A second approach to key validation is to have the two parties run an additional zero-knowledge proof protocol to validate the SIDH key. In this section we present a new isogeny-based zero-knowledge identification protocol which, unlike previous such protocols, validates all elements of an SIDH key. By itself, our protocol has non-negligible soundness error. Since we require negligible soundness error for key validation purposes, we must repeat this protocol many times. We refer to Section 6 for a discussion of efficiency considerations. One can apply a generic transformation such as the Fiat-Shamir [19] or Unruh transformation [18] in order to convert the resulting interactive protocol into a non-interactive transcript.

In the original De Feo-Jao-Plût zero-knowledge identification scheme [10], a prover publishes $(E, E/A, \phi_A|_{E[n_B]})$ and wishes to prove knowledge of A. The prover chooses a commitment $\psi\colon E \to E/B$ and divulges E/B and $E/\langle A, B\rangle$ (see Figure 1). The verifier sends a challenge bit $b \in \{0, 1\}$ and the prover responds with $B = \ker\psi$ or $\ker\phi'$ depending on whether $b = 0$ or $b = 1$. In the $b = 0$ case, the verifier checks that $\ker\psi$ yields E/B and $E/\langle A, B\rangle$, and in the $b = 1$ case the verifier checks that $\ker\phi'$ yields $E/\langle A, B\rangle$. The scheme is secure under the Computational Supersingular Isogeny (CSSI) and Decisional Supersingularity Problem (DSSP) assumptions [10].

Our main contribution of this section is a new zero-knowledge proof which validates not only knowledge of A but also correctness of the auxiliary data $\phi_A|_{E[n_B]}$. We first observe that a new proof is in fact needed: the above proof does not always detect an invalid $\phi_A|_{E[n_B]}$. As explained in Section 4, a dishonest prover (Alice, in this case) can modify $\phi_A|_{E[n_B]}$ using a GPST matrix in such a way that the shared secret computation is unchanged if and only if a certain ℓ-torsion point lies in B. The prover, who also chooses B, can choose B so that the requisite ℓ-torsion point indeed lies in B, in which case the corresponding value of $E/\langle A, B\rangle$ is equal to the correct value. For such B, no choice of response $b \in \{0, 1\}$ by the verifier will detect this modification. In cases where the proof is repeated many times, it may be possible for a verifier to detect the resulting bias in B and flag the prover as a likely cheater, but this technique is more complicated than a simple Σ-protocol, and we do not pursue it here. Instead, we propose to exploit the availability of multiple secrets from degenerate keys in order to validate $\phi_A|_{E[n_B]}$, using a modified Σ-protocol.

Our new zero-knowledge proof proceeds as follows. We use the base curve $E = E_0$ with j-invariant 0. In the commitment phase, the prover publishes E/B and the three shared secrets $E_1 = E/\langle\eta_6(A), B\rangle$, $E_2 = E/\langle\eta_6^2(A), B\rangle$, and $E_3 = E/\langle\eta_6^3(A), B\rangle$. The verifier choose a challenge $b \in \{0, 1, 2, 3\}$. In the $b = 0$ case, the prover responds with B, and the verifier computes $\psi_i\colon E \to E/\eta_6^i(B)$ and $\psi_i'\colon E/A \to E/\langle A, \eta_6^i(B)\rangle$ for $i = 1, 2, 3$ as in the SIDH protocol, and verifies that the isogenies $(\psi_3, \psi_1', \psi_2', \psi_3')$ have codomains $(E/B, E_1, E_2, E_3)$ respectively. The verifier also checks that $\{B, \eta_6(B), \eta_6^2(B)\}$ are pairwise independent, so that the results of Section 4 apply. In the other cases, the prover responds with the kernel of the isogeny $E/B \to E_b$, and the verifier computes the isogeny using this kernel and verifies that its codomain matches the commitment E_b.

Correctness of our protocol is immediate. Zero-knowledge follows easily from the proof of [10, Theorem 6.3], as follows: If the simulator guesses $b = 0$, then it chooses B and produces the commitment data $(E/B, E_1, E_2, E_3)$ from its knowledge of B, and responds as the honest prover would respond to the challenge $b = 0$. If the simulator guesses $b = 1, 2, 3$, then it chooses E/B and the isogenies $E/B \to E_b$ randomly of degree ℓ^e, and responds with the kernels of these isogenies to the challenge $b = 1, 2, 3$. These responses are indistinguishable from an honest prover under DSSP. Revealing these extra (codomains of) maps does not create any extra insecurity, since a simulator (who, in the $b = 0$ case, knows B) can (in the $b = 0$ case) generate all these maps on their own anyway.

To prove soundness, the proof of [10, Theorem 6.3] shows that E/A is a valid curve, so we only need to prove the correctness of the auxiliary data. Recall that the verifier checks in the $b = 0$ case that $\{B, \eta_6(B), \eta_6^2(B)\}$ are pairwise independent. Assuming this is the case, the results of Section 4 imply that any GPST-style manipulation of $\phi_A|_{E[n_B]}$ will cause the computation of at least two of the curves E_1, E_2, E_3 in the $b = 0$ case (a computation which depends on the value of $\phi_A|_{E[n_B]}$) to fail, in the sense that these curves E_i admit no isogeny $E/B \to E_i$ of degree ℓ^e. Hence if the verifier chooses $b = 0$ with probability $2/5$, and each of $b = 1, 2, 3$ with

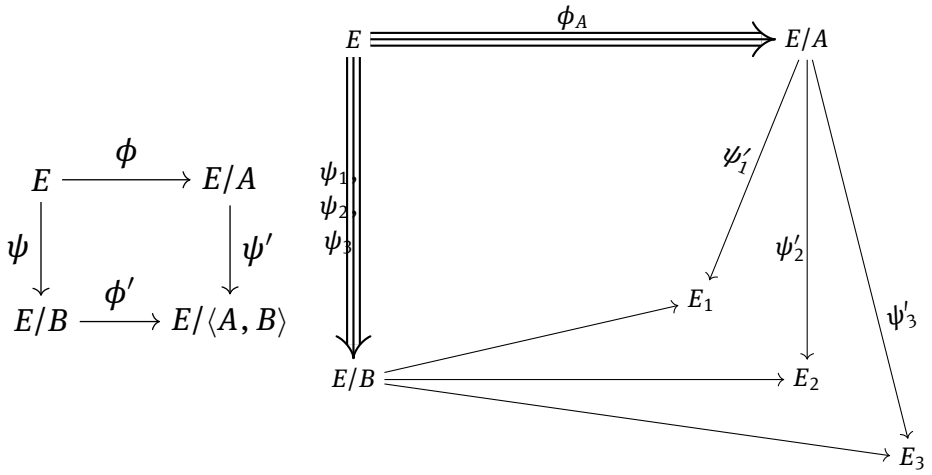


Figure 1: Original De Feo-Jao-Plût protocol (left) and our new protocol (right).

probability $1/5$, then the failure probability for a cheating prover is at least $2/5$: either the $b = 0$ response is flawed, which the verifier will detect whenever the verifier chooses the $b = 0$ value (40%probability), or else at least two of the responses out of $b \in \{1, 2, 3\}$ case are flawed, which the verifier will detect whenever the verifier chooses one of these two values (40%probability).

One may try to optimize our zero-knowledge proof by having the prover publish the auxiliary data $\phi_B|_{E[n_A]}$ for the commitment E/B and then using this auxiliary data to derive (say) all three of the kernels $E/B \to E_i$ from one of them. However, this approach is insecure, since the kernel of $E/B \to E_3$ is equal to $\phi_B(A)$, and knowledge of both $\phi_B|_{E[n_A]}$ and $\phi_B(A)$ trivially exposes the original secret A. Another idea is to reveal more than one of the maps $E/B \to E_b$ at once. While this strategy may work in practice, we cannot prove it to be zero-knowledge, since a simulator cannot accurately simulate two related maps simultaneously.

6 Efficiency

We compare the efficiency of our two methods, using the 256-bit classical / 128-bit quantum security level (which is the only security level treated in [2]). For our first method, using the primes $\ell_A = 11$ and $\ell_B = 13$, the results of Section 4 show that we need $\alpha = 18$ and $\beta = 17$ respectively in order to implement our variant of the k-SIDH NIKE protocol with this security level. The public keys are 18 (respectively 17) times larger than in SIDH, and each party computes $3 \cdot 18 \cdot 17 = 918$ shared secrets. As with "standard" SIDH using $\ell_A = 2$ and $\ell_B = 3$, there is no difficulty in finding primes p of the appropriate size. Costello and Hisil [9, Fig. 2] indicate that such primes are about 3 to 4 times slower than standard primes.

Our second NIKE proposal, using explicit key validation via zero-knowledge proofs, requires approximately 347 proof iterations for 256-bit security (since $(3/5)^{347} \approx 2^{-256}$). Relative to an SIDH iteration, each zero-knowledge proof iteration is also larger (since there is more commitment data) and slower (since multiple isogenies potentially need to be verified) by a small constant factor. Comparing our two methods, the public keys for the second method are larger, and the computational cost of the two is approximately the same at the 256-bit security level. Our second scheme scales better in computational cost with increasing security, since the computational cost grows only linearly in the security level instead of quadratically. However, our first scheme has smaller public keys, and validates both keys at once, whereas the second scheme needs to be repeated by each party in order to validate both keys.

7 Implementation

We implemented the automorphism-based multi-secret SIDH protocol described in this paper, using Doliskani's publicly available SIDH reference implementation [12] as a base. Our implementation uses $p = 2 \cdot 13^{102} \cdot 11^{111} + 1$ and $E : y^2 = x^3 + (32 + \sqrt{-1})$. It can be found at [20]. Our implementation is intended as a proof-of-concept to validate the correctness of the construction, and as an aid to non-specialists who may benefit more from working code than a detailed technical description.

Acknowledgement: This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund, CryptoWorks21, Public Works and Government Services Canada, and the Royal Bank of Canada.

References

[1] Gora Adj, Omran Ahmadi and Alfred Menezes, *On isogeny graphs of supersingular elliptic curves over finite fields*, Cryptology ePrint Archive, Report 2018/132, 2018, https://eprint.iacr.org/2018/132.

[2] Reza Azarderakhsh, David Jao and Christopher Leonardi, Post-Quantum Static-Static Key Agreement Using Multiple Protocol Instances, in: *Selected Areas in Cryptography — SAC 2017* (Carlisle Adams and Jan Camenisch, eds.), pp. 45–63, Springer International Publishing, Cham, 2018.

[3] Jean-François Biasse, Annamaria Iezzi and Jr Michael J. Jacobson, *A note on the security of CSIDH*, 2018.

[4] Xavier Bonnetain and André Schrottenloher, *Quantum Security Analysis of CSIDH and Ordinary Isogeny-based Schemes*, Cryptology ePrint Archive, Report 2018/537, 2018, https://eprint.iacr.org/2018/537.

[5] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny and Joost Renes, *CSIDH: An Efficient Post-Quantum Commutative Group Action*, Cryptology ePrint Archive, Report 2018/383, 2018, https://eprint.iacr.org/2018/383.

[6] Denis X. Charles, Kristin E. Lauter and Eyal Z. Goren, Cryptographic Hash Functions from Expander Graphs, *Journal of Cryptology* **22** (2009), 93–113.

[7] Microsoft Corporation, *SIDH v3.0*, https://github.com/Microsoft/PQCrypto-SIDH, 2018.

[8] Anamaria Costache, Brooke Feigon, Kristin Lauter, Maike Massierer and Anna Puskas, *Ramanujan graphs in cryptography*, Cryptology ePrint Archive, Report 2018/593, 2018, https://eprint.iacr.org/2018/593.

[9] Craig Costello and Huseyin Hisil, A Simple and Compact Algorithm for SIDH with Arbitrary Degree Isogenies, in: *Advances in Cryptology – ASIACRYPT 2017* (Tsuyoshi Takagi and Thomas Peyrin, eds.), pp. 303–329, Springer International Publishing, Cham, 2017.

[10] Luca De Feo, David Jao and Jérôme Plût, Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, *J. Math. Cryptol.* **8** (2014), 209–247.

[11] Max Deuring, Die Typen der Multiplikatorenringe elliptischer Funktionenkörper, *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* **14** (1941), 197–272.

[12] Javad Doliskani, *SIDH C Reference*, https://github.com/sidh-crypto/sidh-c-reference, 2017.

[13] Steven D. Galbraith, Constructing Isogenies between Elliptic Curves Over Finite Fields, *LMS Journal of Computation and Mathematics* **2** (1999), 118–138.

[14] Steven D. Galbraith, Christophe Petit, Barak Shani and Yan Bo Ti, On the Security of Supersingular Isogeny Cryptosy stems, in: *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pp. 63–91, 2016.

[15] David Jao and Luca De Feo, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, Post-quantum cryptography, Lecture Notes in Comput. Sci. 7071, Springer, Heidelberg, 2011, pp. 19–34.

[16] Richard Lindner and Chris Peikert, Better Key Sizes (and Attacks) for LWE-Based Encryption, in: *Topics in Cryptology – CT-RSA 2011* (Aggelos Kiayias, ed.), pp. 319–339, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[17] Patrick Longa and Francesco Sica, Four-Dimensional Gallant–Lambert–Vanstone Scalar Multiplication, *Journal of Cryptology* **27** (2014), 248–283.

[18] Dominique Unruh, Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model, in: *Advances in Cryptology — EUROCRYPT 2015* (Elisabeth Oswald and Marc Fischlin, eds.), pp. 755–784, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[19] Dominique Unruh, Post-quantum Security of Fiat-Shamir, in: *Advances in Cryptology — ASIACRYPT 2017* (Tsuyoshi Takagi and Thomas Peyrin, eds.), pp. 65–95, Springer International Publishing, Cham, 2017.

[20] David Urbanik, *Multikey SIDH*, http://csclub.uwaterloo.ca/~dburbani/work/multikey-sidh.zip, 2018.