A Note on the Static-Static Key Agreement Protocol from Supersingular Isogenies

Selçuk Kayacan

TÜBİTAK BİLGEM
41470 Gebze, Kocaeli, Turkey.
e-mail: selcuk.kayacan@tubitak.gov.tr

Abstract

The basic Supersingular Isogeny Diffie-Hellman (SIDH) key agreement protocol is insecure due to an attack described by Galbraith, Petit, Shani and Ti. In this note we present two variants of SIDH that are immune to this attack.

Keywords: key agreement, post-quantum cryptography, supersingular isogeny Diffie-Hellman

1 Introduction

Despite its relatively young age isogeny-based cryptography has become an attractive candidate for building quantum resistant cryptosystems among the other branches of post-quantum cryptography. This is partly due to the very small key sizes it offers. The Supersingular Isogeny Diffie-Hellman (SIDH) key exchange protocol was introduced by Jao and De Feo [3] as a promising quantum secure cryptographic primitive and subsequently improved by various people [4, 1, 2]. Besides offering the smallest key sizes in post-quantum cryptography, SIDH features a Diffie-Hellman style message flow. Alice and Bob exchanges their public curves together with the images of the pairs of torsion points under their secret isogenies. This message flow enables them to arrive at isomorphic curves so that they can use the common j-invariant as the shared secret. Unfortunately, the static variant of SIDH is not secure due to an attack described by Galbraith et al. [5]: Bob maliciously sends to Alice malformed points so that algorithm success or failure reveals some bits of the Alice's private key. Therefore, it is possible for an attacker to obtain the victim's long-term private key over multiple sessions.

Previously, Kirkwood et al. identified the key leakage problem in the isogeny-based key agreement schemes [6]. The usual way to solve this problem is to validate public keys of the participants as a requirement of the protocol specification. However, unlike classical public key cryptosystems, in supersingular isogeny setting validating public keys directly, i.e., performing a check on the key itself, is a non-trivial problem. To circumvent this issue Kirkwood et al. proposed the so-called indirect key validation that can be performed using a Fujisaki-Okamoto type transformation. This solution requires one of the parties to disclose its private key by the end of the session. Therefore, Kirkwood et al.'s approach is not satisfactory if static-static key agreement is desired. As a side note

it seems static-static key agreement (with ephemeral values) is possible if both sides encapsulates the other side's public key (compare with AKE-SIDH protocol in [10]).

The problem of direct key validation was studied by Costello et al. [1, Section 9] in the SIDH setting, whereby they show how to check that the two torsion points are of full order and independent within their algorithm suite. As is shown by Galbraith et al. [5] an attacker can modify the torsion points in a way that passes these checks and still learn the secret bits of the receiver's private key. Moreover, the known direct key validation methods are not enough to prevent Galbraith et al.'s attack. A solution to this problem would be validating the existence of a corresponding isogeny which is the private key of the sender. In [11] Urbanik and Jao has shown that determining the existence of such an isogeny (key validation problem) is equivalent to the supersingular isogeny problem in which the security of SIDH lies. An alternative route would be to employ a zero-knowledge proof of knowledge protocol to this end, such as the one described by Jao, De Feo and Plût [4, Section 3.1], possibly with the aid of a certification authority [9, Section 7.2].

In [8] Azarderakhsh, Jao and Leonardi introduced a generic transformation to make the key agreement protocols satisfying certain security properties immune to the attacks using specialized public keys to leak information about the private key such as the Galbraith et al.'s attack. This transformation requires the use of k different keys for each party and to establish the shared secret k^2 key exchange must be performed for each of the k^2 combinations of those keys. It has shown that SIDH satisfies the necessary security properties and $k = 92$ is enough to guarantee 128-bit quantum security level.

Our Contributions. In this note we present two variants of SIDH to achieve static-static key agreement both featuring a Diffie-Hellman style message flow. Two attack models can be defined in terms of access to an oracle in this setting. In the first model given the (possibly malformed) public key of the attacker oracle answers whether the victim and the attacker would compute the same shared secret; and in the second model oracle outputs the victim's shared secret.

- *Protocol A:* Each party uses two "complementary" SIDH private keys and to establish the shared secret four instances of SIDH protocol must be performed. This protocol is secure against Galbraith et al.'s attack in the first model, but not secure in the second model.
- *Protocol B:* Each party uses a single SIDH private key and computes the j-invariants of two curves which are distant away from each other by specific paths in the isogeny graph. This protocol is secure against Galbraith et al.'s attack in the second model.

In Section 2 we review the background material and fix a notation. Section 3 briefly introduces two attack models that is relevant to this work. In Section 4 and Section 5 we describe and discuss two possible solutions, Protocol A and Protocol B, respectively to the problem of static-static key exchange from supersingular isogenies. Finally, in Section 6 we conclude the paper.

2 Preliminaries

Elliptic Curves and Isogenies. For general background on elliptic curves the reader may refer to [12]. Let E and E' be two elliptic curves defined over the finite field $\mathbb{F}_q$. An

isogeny $\phi\colon E \to E'$ is a rational map (i.e., quotients of polynomials with coefficients from $\mathbb{F}_q$) which is also a group homomorphism. For a separable isogeny its degree (as a rational map) is equal to the order of its kernel (as a homomorphism). The map ϕ is called an isomorphism provided that only the identity $\mathcal{O}_E$ lies in the kernel. For a positive integer m, the map $[m]$ taking the point P on E to the point mP (scalar multiplication by m) is an example of an isogeny from E to itself. Each isogeny $\phi\colon E \to E'$ has a dual isogeny $\hat{\phi}\colon E' \to E$ satisfying $\phi \circ \hat{\phi} = [m]$ and $\hat{\phi} \circ \phi = [m]$, where m is the degree of ϕ. We say ϕ is a m-isogeny if its degree is equal to m.

The set of isogenies from E to itself over the algebraic closure $\bar{\mathbb{F}}_q$ of $\mathbb{F}_q$ together with the zero map taking any point on E to the identity element form a ring, where addition is pointwise addition and multiplication is function composition. This ring, denoted $\mathrm{End}(E)$, is called the endomorphism ring of E. The curve E is said to be a supersingular curve if $\mathrm{End}(E)$ is an order in a quaternion algebra. Otherwise, we say E is ordinary. If E is a supersingular elliptic curve, any curve which is isogenous to E is also supersingular.

Any two elliptic curves E and E' over $\mathbb{F}_q$ share the same j-invariant if and only if they are isomorphic over $\bar{\mathbb{F}}_q$. Thus, the isomorphism class of E over $\bar{\mathbb{F}}_q$ can be represented by its j-invariant $j(E)$. If the curve is given in Weierstrass form $E\colon y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_q$, then the associated j-invariant is

$$j(E) = 1728\frac{4a^3}{4a^3 + 27b^2} \in \mathbb{F}_q.$$

Since any isogeny has a dual, being isogenous is an equivalence relation over the set of isomorphism classes of elliptic curves over $\bar{\mathbb{F}}_q$. If E is a supersingular curve, then its j-invariant lies in $\mathbb{F}_{p^2}$, where p is the characteristic of the field $\mathbb{F}_q$. Therefore, there are only finitely many isomorphism classes of supersingular elliptic curves. In fact this number is approximately $p/12$ [12, V.4.1].

Up to isomorphism an isogeny is determined uniquely by its kernel. That is, if $\phi\colon E \to E'$ and $\psi\colon E \to E''$ are two isogenies having the same kernel K, then there exists an isomorphism $\iota\colon E' \to E''$ such that $\iota \circ \phi = \psi$. Given a finite subgroup K of E, the corresponding isogeny ϕ can be calculated by using Vélu's formulas [13]. We shall denote the image curve $\phi(E)$ by E/K.

For a positive integer m, the m-torsion subgroup $E[m]$ is defined as the kernel of $[m]$, the multiplication by m mapping. Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, where p is the characteristic of the field; and let ℓ be a positive integer which is not divisible by p. Then the ℓ-torsion subgroup $E[\ell]$ is isomorphic to $\mathbb{Z}/\ell\mathbb{Z} \oplus \mathbb{Z}/\ell\mathbb{Z}$ and, consequently, up to isomorphism there are $\ell + 1$ distinct isogenies whose domain is E and degree is ℓ. Let L be a set of primes. The isogeny graph $X(\mathbb{F}_q, L)$ is defined as the directed graph whose vertices are isomorphism classes of supersingular curves over $\mathbb{F}_q$ labeled with the associated j-invariants and edges are equivalence classes of ℓ-isogenies where $\ell \in L$.

SIDH Key Exchange. SIDH requires a prime of special form $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$, where ℓ_A and ℓ_B are small prime numbers and f is a cofactor. Normally, we choose $\ell_a^{e_A} \approx \ell_B^{e_B} \approx 2^\lambda$, where λ is the security parameter. Construct a supersingular elliptic curve E over $\mathbb{F}_{p^2}$ so that the number of points of $E(\mathbb{F}_{p^2})$ is $(\ell_A^{e_A}\ell_B^{e_B}f)^2$. To motivate the construction of SIDH, let us consider the following commutative diagram, where T_A and

Table 1: SIDH Key Exchange Protocol

Public parameters: E, P_A, Q_A, P_B, Q_B	
Alice	***Bob***
$\alpha \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$	$\beta \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$
$\phi_A \colon E \to E/\langle P_A + \alpha Q_A \rangle$	$\phi_B \colon E \to E/\langle P_B + \beta Q_B \rangle$
$E_A := \phi_A(E)$	$E_B := \phi_B(E)$
$R_B := \phi_A(P_B), S_B := \phi_A(Q_B)$	$R_A := \phi_B(P_A), S_A := \phi_B(Q_A)$
$\xrightarrow{(E_A, R_B, S_B)}$	
$\xleftarrow{(E_B, R_A, S_A)}$	
$\psi_A \colon E_B \to E_B/\langle R_A + \alpha S_A \rangle$	$\psi_B \colon E_A \to E_A/\langle R_B + \beta S_B \rangle$
$E_{BA} := \psi_A(E_B)$	$E_{AB} := \psi_B(E_A)$
$j(E_{BA})$	$j(E_{AB})$

T_B are some points of order $\ell_A^{e_A}$ and $\ell_B^{e_B}$ respectively:

$$\begin{array}{ccc} E & \xrightarrow{\phi} & E/\langle T_A \rangle \\ \downarrow{\scriptstyle \psi} & & \downarrow \\ E/\langle T_B \rangle & \longrightarrow & E/\langle T_A, T_B \rangle \end{array}$$

Notice that the isogenies ϕ and ψ determines two paths consisting of ℓ_A- and ℓ_B-isogenies respectively in the isogeny graph $X(\mathbb{F}_{p^2}, \{\ell_A, \ell_B\})$. Using Vélu's formulas directly to compute ϕ or ψ requires exponentially large space in terms of the security parameter λ. However, we can write them as the composition of ℓ_A- or ℓ_B-isogenies so that ϕ or ψ would be computed by applying Vélu's formulas to those low degree isogenies. To make this diagram commutative one should also compute the isogenies $\phi' \colon E/\langle T_B \rangle \to E/\langle T_A, T_B \rangle$ and $\psi' \colon E/\langle T_A \rangle \to \langle T_A, T_B \rangle$ taking $\langle \psi(T_A) \rangle$ and $\langle \phi(T_B) \rangle$ respectively to the identity element in the image curve. Given E, $E/\langle T_A \rangle$ and $E/\langle T_B \rangle$ it is supposed to be difficult to find $E/\langle T_A, T_B \rangle$.

With those remarks, SIDH key exchange proceeds as follows. Take a pair of basis points P_A, Q_A generating the torsion subgroup $E[\ell_A^{e_A}]$ and a pair of basis points P_B, Q_B generating $E[\ell_B^{e_B}]$. Alice chooses an integer α uniformly at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and computes the isogeny $\phi_A \colon E \to E_A = E/\langle P_A + \alpha Q_A \rangle$. Alice's private key is α and her public key is the tuple $(E_A, \phi_A(P_B), \phi_A(Q_B))$. Similarly, Bob chooses an integer $\beta \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ and form his public key $(E_B, \phi_B(P_A), \phi_B(Q_A))$, where $\phi_B \colon E \to E_B = E/\langle P_B + \beta Q_B \rangle$. Upon receiving each other's public key Alice and Bob compute the j-invariant of $E/\langle P_A + \alpha Q_A, P_B + \beta Q_B \rangle$ as their shared secret (see Table 1 for details).

Notice that to arrive at the same isomorphism class of a curve, parties must share with each other the image of some torsion points under their secret isogenies. The SIDH Problem can be formulated as follows.

Problem S (SIDH Problem)**.** *With the notation of Table 1, given E, P_A, Q_A, P_B, Q_B, E_A, R_A, S_A, E_B, R_B, S_B to determine $j(E_{AB})$.*

Galbraith et al.'s Attack. If T is a point on E of order ℓ^e, then the isogeny $\psi \colon E \to$

$E/\langle T\rangle$ can be factored into ℓ-isogenies

$$E = E_1 \xrightarrow{\psi_1} E_2 \xrightarrow{\psi_2} \cdots \xrightarrow{\psi_e} E_{e+1} = E/\langle T\rangle,$$

where $T_1 = T, T_{i+1} = \psi_i(T_i)$ and the kernel of ψ_1 is $\langle \ell^{e-1}T\rangle$, the kernel of ψ_i, for $i > 1$, is $\langle \ell^{e-i}\psi_{i-1}(T_{i-1})\rangle$. With the notation of Table 1, put $E := E_B$, $\ell^e := \ell_A^{e_A}$, $T := R_A + \alpha S_A$ and $\psi := \psi_A$. Write $\alpha = \alpha'\ell_A + \alpha_0$ where $\alpha_0 \in \{0, 1, \ldots, \ell_A - 1\}$. Since $\ell^{e-1}T = \ell_A^{e_A-1}R_A + \alpha_0\ell_A^{e_A-1}S_A$, the value of α_0 determines which neighbor of E_B in the isogeny graph $X(\mathbb{F}_{p^2}, \{\ell_A\})$ would be $\psi_1(E_B)$.

Suppose Bob acts maliciously to learn about the secret bits of Alice's private key α. To be more precise, Bob makes a guess α_0^* on the value of α_0 and accordingly modifies the torsion points R_A and S_A as

$$R_A^* = R_A - \alpha_0^*\ell_A^{e_A-1}S_A, \quad S_A^* = (1 + \ell_A^{e_A-1})S_A.$$

Upon receiving the tuple (E_B, R_A^*, S_A^*) Alice would compute $E_{BA}^* = E_B/\langle R_A^* + \alpha S_A^*\rangle$ which is isomorphic to $E_{AB} = E_A/\langle R_B + \beta S_B\rangle$ if $\alpha_0^* = \alpha_0$. In other words, Bob learns some of the least significant bits of Alice's private key after a successful run of the protocol. Actually, even the failure of key agreement reveals one bit of information. Moreover, algorithm failure implies that the curve $E_{BA}^* = E_B/\langle R_A^* + \alpha S_A^*\rangle$ would be ℓ_A^2-isogenous to $E_{BA} = E_B/\langle R_A + \alpha S_A\rangle$. To see this observe that the subgroups $\langle R_A^* + \alpha S_A^*\rangle$ and $\langle R_A + \alpha S_A\rangle$ of E_B intersects at a subgroup of order $\ell_A^{e_A-1}$. In other words, even in the case of failure, Bob knows that there are at most $\ell_A(\ell_A + 1)$ possibilities for the value of $j(E_{BA}^*)$ computed by Alice.

This attack can be turned into an adaptive attack in which the attacker learns the private key of the victim over multiple sessions. Let α_i be the coefficient of ℓ_A^i in the base ℓ_A expansion of α and write

$$\alpha = \alpha'\ell_A^i + \kappa_i \quad \text{with } \kappa_i < \ell_A^i.$$

Suppose Bob knows κ_i. Then he makes a guess α_i^* on the value of α_i and modifies his public key by setting

$$R_A^* = R_A - (\alpha_i^*\ell_A^i + \kappa_i)\ell_A^{e_A-i-1}S_A, \quad S_A^* = (1 + \ell_A^{e_A-i-1})S_A.$$

It can be easily verified that success or failure of the key agreement reveals at least one bit of information as in the previous case. We shall remark that the values of R_A^* and S_A^* can be further fine tuned by a scaling so that the known key validation methods would not detect them.

3 Attack Models

Following [5], two attack models can be defined in terms of access to an oracle. Let E_1, E_2 be two elliptic curves isogenous to E and let T_1, T_2 be two torsion points in $E_1[\ell^e]$. We assume Bob may act dishonestly and use specially designed public key in a key agreement.

- $\mathsf{Oracle}_1(E_1, T_1, T_2, E_2) = \begin{cases} 1 \text{ if } j(E_2) = j(E_1/\langle T_1 + \alpha T_2\rangle) \\ 0 \text{ otherwise} \end{cases}$

  Normally, Bob feeds this oracle with $E_1 = E_B$, $T_1 = R_A^*$, $T_2 = S_A^*$, $E_2 = E_A/\langle R_B + \beta S_B\rangle$ and sees whether the key agreement would be successful if Alice computes the shared secret with (E_B, R_A^*, S_A^*).

- $\mathsf{Oracle}_2(E_1, T_1, T_2) = E_1/\langle T_1 + \alpha T_2\rangle$
  This oracle outputs the shared secret computed by Alice if the provided public key is (E_1, T_1, T_2). Recall that, in a SIDH instance, even if the key agreement results in failure, Bob would still guess the shared secret computed by Alice with high probability. By calling this oracle Bob may verify his guess.

4 Protocol A

Description. We choose a prime of special form $p = \ell_A^{e_A}\ell_B^{e_B} f \pm 1$ and construct a supersingular elliptic curve E so that the number of elements in $E(\mathbb{F}_{p^2})$ is $(\ell_A^{e_A}\ell_B^{e_B} f)^2$. As in the case of SIDH key exchange protocol, we take a pair of basis points P_A, Q_A generating the torsion subgroup $E[\ell_A^{e_A}]$ and a pair of basis points P_B, Q_B generating $E[\ell_B^{e_B}]$. Alice chooses an integer α uniformly at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$. Let $\bar{\alpha}$ be the bitwise complement of α. Alice computes two isogenies

$$\phi_A \colon E \to E_A = E/\langle P_A + \alpha Q_A\rangle \quad \text{and} \quad \phi_{\overline{A}} \colon E \to E_{\overline{A}} = E/\langle P_A + \bar{\alpha} Q_A\rangle.$$

Alice's private key is α and her public key is $(E_A, \phi_A(P_B), \phi_A(Q_B), E_{\overline{A}}, \phi_{\overline{A}}(P_B), \phi_{\overline{A}}(Q_B))$. Similarly, Bob chooses an integer $\beta \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, computes isogenies

$$\phi_B \colon E \to E_B = E/\langle P_B + \beta Q_B\rangle \quad \text{and} \quad \phi_{\overline{B}} \colon E \to E_{\overline{B}} = E/\langle P_B + \bar{\beta} Q_B\rangle$$

and form his public key $(E_B, \phi_B(P_A), \phi_B(Q_A), E_{\overline{B}}, \phi_{\overline{B}}(P_A), \phi_{\overline{B}}(Q_A))$, where $\bar{\beta}$ is the bitwise complement of β. Notice that each party uses a public key which is equivalent to the concatenation of two SIDH public keys. Upon receiving each other's public key, Alice and Bob computes the j-invariants of the curves obtained by completing all possible four SIDH key exchange instance on their side. Let H be a preimage resistant hash function. The shared secret is the value of the function H computed on the concatenation of four j-invariants (see Table 2 for details).

Discussion. The correctness of Protocol A follows from the correctness of SIDH key exchange protocol. This protocol is secure against Galbraith et al.'s attack if the attacker has access to Oracle_1. Suppose Bob made a guess on some bits of Alice's private key α and altered the torsion points R_A and S_A into R_A^* and S_A^*. Observe that whether Bob is right on his guess or not, the session would end-up with algorithm failure, since the equalities

$$j(E_{BA}) = j(E_{BA}^*) \quad \text{and} \quad j(E_{B\overline{A}}) = j(E_{B\overline{A}}^*)$$

does not hold simultaneously.

Protocol A is not secure against Galbraith et al.'s attack if the attacker has access to Oracle_2. Recall that even if key agreement results in failure, Bob can compute a small set of possible values for the shared secret computed by Alice. By making a query to Oracle_2 Bob compares Alice's shared secret with the possible values and so can decide whether his guess on the bits of α was right or wrong. Thus, Bob would be able to proceed with the attack.

Protocol A is very similar to the generic solution presented in [8] to make SIDH key exchange protocol secure against Galbraith et al.'s type attacks. This transformation requires the use of k different SIDH key pair by each party and k^2 SIDH instance must be performed to compute the shared secret. Since private keys are independent, there is always a possibility for the attacker to successfully guess secret key bits. However, by taking k large enough this possibility can be made negligible. Further, even the

Table 2: Protocol A

Public parameters: E, P_A, Q_A, P_B, Q_B, H		
Alice		***Bob***
$\alpha \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$		$\beta \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$
$\phi_A \colon E \to E/\langle P_A + \alpha Q_A \rangle$		$\phi_B \colon E \to E/\langle P_B + \beta Q_B \rangle$
$E_A := \phi_A(E)$		$E_B := \phi_B(E)$
$R_B := \phi_A(P_B), S_B := \phi_A(Q_B)$		$R_A := \phi_B(P_A), S_A := \phi_B(Q_A)$
$\phi_{\overline{A}} \colon E \to E/\langle P_A + \bar{\alpha} Q_A \rangle$		$\phi_{\overline{B}} \colon E \to E/\langle P_B + \bar{\beta} Q_B \rangle$
$E_{\overline{A}} := \phi_{\overline{A}}(E)$		$E_{\overline{B}} := \phi_{\overline{B}}(E)$
$\overline{R}_B := \phi_{\overline{A}}(P_B), \overline{S}_B := \phi_{\overline{A}}(Q_B)$		$\overline{R}_A := \phi_{\overline{B}}(P_A), \overline{S}_A := \phi_{\overline{B}}(Q_A)$
$\alpha,\ (E_A, R_B, S_B, E_{\overline{A}}, \overline{R}_B, \overline{S}_B)$	**static parameters**	$\beta,\ (E_B, R_A, S_A, E_{\overline{B}}, \overline{R}_A, \overline{S}_A)$
	$\xrightarrow{(E_A, R_B, S_B, E_{\overline{A}}, \overline{R}_B, \overline{S}_B)}$	
	$\xleftarrow{(E_B, R_A, S_A, E_{\overline{B}}, \overline{R}_A, \overline{S}_A)}$	
$\psi_A \colon E_B \to E_B/\langle R_A + \alpha S_A \rangle$		$\psi_B \colon E_A \to E_A/\langle R_B + \beta S_B \rangle$
$E_{BA} := \psi_A(E_B), j_{BA} := j(E_{BA})$		$E_{AB} := \psi_B(E_A), j_{AB} := j(E_{AB})$
$\psi_{\overline{A}} \colon E_B \to E_B/\langle R_A + \bar{\alpha} S_A \rangle$		$\psi_{\overline{B}} \colon E_A \to E_A/\langle R_B + \bar{\beta} S_B \rangle$
$E_{B\overline{A}} := \psi_{\overline{A}}(E_B), j_{B\overline{A}} := j(E_{B\overline{A}})$		$E_{A\overline{B}} := \psi_{\overline{B}}(E_A), j_{A\overline{B}} := j(E_{A\overline{B}})$
$\bar{\psi}_A \colon E_{\overline{B}} \to E_{\overline{B}}/\langle \overline{R}_A + \alpha \overline{S}_A \rangle$		$\bar{\psi}_B \colon E_{\overline{A}} \to E_{\overline{A}}/\langle \overline{R}_B + \beta \overline{S}_B \rangle$
$E_{\overline{B}A} := \bar{\psi}_A(E_{\overline{B}}), j_{\overline{B}A} := j(E_{\overline{B}A})$		$E_{\overline{A}B} := \bar{\psi}_B(E_{\overline{A}}), j_{\overline{A}B} := j(E_{\overline{A}B})$
$\bar{\psi}_{\overline{A}} \colon E_{\overline{B}} \to E_{\overline{B}}/\langle \overline{R}_A + \bar{\alpha} \overline{S}_A \rangle$		$\bar{\psi}_B \colon E_{\overline{A}} \to E_{\overline{A}}/\langle \overline{R}_B + \bar{\beta} \overline{S}_B \rangle$
$E_{\overline{B}\overline{A}} := \bar{\psi}_{\overline{A}}(E_{\overline{B}}), j_{\overline{B}\overline{A}} := j(E_{\overline{B}\overline{A}})$		$E_{\overline{A}\overline{B}} := \bar{\psi}_{\overline{B}}(E_{\overline{A}}), j_{\overline{A}\overline{B}} := j(E_{\overline{A}\overline{B}})$
$H(j_{BA}, j_{B\overline{A}}, j_{\overline{B}A}, j_{\overline{B}\overline{A}})$		$H(j_{AB}, j_{A\overline{B}}, j_{\overline{A}B}, j_{\overline{A}\overline{B}})$

queries to Oracle_2 might become useless as the number of possibilities for the shared secret computed by Alice would be to much.

In Protocol A parties use a secret key and its bitwise complement in two different SIDH instance with the same public key. This might be a concern.

Problem A. *With the notation of Table 2, given* E, P_A, Q_A, P_B, Q_B, E_A, R_A, S_A, $E_{\overline{A}}$, $\overline{R}_A$, $\overline{S}_A$, E_B, R_B, S_B, $E_{\overline{B}}$, $\overline{R}_B$, $\overline{S}_B$ *to determine* $H(j_{AB}, j_{A\overline{B}}, j_{\overline{A}B}, j_{\overline{A}\overline{B}})$.

5 Protocol B

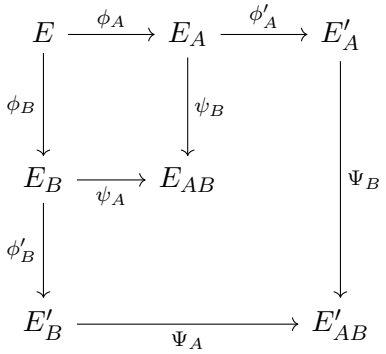
Description. We choose a prime of special form $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ and construct a supersingular elliptic curve E so that the number of elements in $E(\mathbb{F}_{p^2})$ is $(\ell_A^{e_A}\ell_B^{e_B}f)^2$. As in the case of SIDH key exchange protocol, we take a pair of basis points P_A, Q_A generating the torsion subgroup $E[\ell_A^{e_A}]$ and a pair of basis points P_B, Q_B generating $E[\ell_B^{e_B}]$. Let $e_A = f_A + g_A$ and $e_B = f_B + g_B$. Typically, we require $\ell_A^{f_A} \approx \ell_A^{g_A} \approx \ell_B^{f_B} \approx \ell_B^{g_B} \approx 2^\lambda$, where λ is the security parameter. Alice chooses an integer α uniformly at random from $\mathbb{Z}/\ell_A^{f_A}\mathbb{Z}$ and computes the isogeny

$$\phi_A \colon E \to E_A = E/\langle \ell_A^{g_A}(P_A + \alpha Q_A)\rangle.$$

Let $K_A := \phi_A(P_A + (\ell_A^{e_A-1} + \alpha)Q_A)$, $R_B := \phi_A(P_B)$ and $S_B := \phi_A(Q_B)$. Alice's private key is α and her public key is the tuple (E_A, K_A, R_B, S_B). Similarly, Bob chooses an integer $\beta \in_R \mathbb{Z}/\ell_B^{f_B}\mathbb{Z}$ and computes the isogeny

$$\phi_B \colon E \to E_B = E/\langle \ell_B^{g_B}(P_B + \beta Q_B)\rangle.$$

Let $K_B := \phi_B(P_B + (\ell_B^{e_B-1} + \beta)Q_B)$, $R_A := \phi_B(P_A)$ and $S_A := \phi_B(Q_A)$. Bob's private key is β and his public key is the tuple (E_B, K_B, R_A, S_A). Consider the following diagram:



Upon receiving each other's public key, Alice and Bob computes the isogenies ϕ'_B, ψ_A, Ψ_A and ϕ'_A, ψ_B, Ψ_B respectively on their side so that the diagram would be commutative. Let H be a preimage resistant hash function. The shared secret is the value of the function H computed on the concatenation of $j(E_{AB})$ and $j(E'_{AB})$ (see Table 3 for details).

Discussion. The correctness of Protocol B follows from the commutativity of the diagram on page 8. This protocol is secure against Galbraith et al.'s attack if the attacker has access to Oracle_2. Suppose Bob made a guess on some bits of Alice's private key α and altered the torsion points R_A and S_A into R_A^* and S_A^*. Additionally,

Table 3: Protocol B

Public parameters: E, P_A, Q_A, P_B, Q_B, f_A, f_B, H		
Alice		***Bob***
$\alpha \in_R \mathbb{Z}/\ell_A^{f_A}\mathbb{Z}$		$\beta \in_R \mathbb{Z}/\ell_B^{f_B}\mathbb{Z}$
$\phi_A \colon E \to E/\langle \ell_A^{g_A}(P_A + \alpha Q_A)\rangle$		$\phi_B \colon E \to E/\langle \ell_B^{g_B}(P_B + \beta Q_B)\rangle$
$E_A := \phi_A(E)$		$E_B := \phi_B(E)$
$K_A := \phi_A(P_A + (\ell_A^{e_A-1} + \alpha)Q_A)$		$K_B := \phi_B(P_B + (\ell_B^{e_B-1} + \beta)Q_B)$
$R_B := \phi_A(P_B)$, $S_B := \phi_A(Q_B)$		$R_A := \phi_B(P_A)$, $S_A := \phi_B(Q_A)$
α, (E_A, K_A, R_B, S_B)	**static parameters**	β, (E_B, K_B, R_A, S_A)
	$\xrightarrow{(E_A, K_A, R_B, S_B)}$	
	$\xleftarrow{(E_B, K_B, R_A, S_A)}$	
$\phi'_B \colon E_B \to E_B/\langle K_B\rangle$		$\phi'_A \colon E_A \to E_A/\langle K_A\rangle$
$E'_B := \phi'_B(E_B)$		$E'_A := \phi'_A(E_A)$
$\psi_A \colon E_B \to E_B/\langle \ell_A^{g_A}(R_A + \alpha S_A)\rangle$		$\psi_B \colon E_A \to E_A/\langle \ell_B^{g_B}(R_B + \beta S_B)\rangle$
$E_{BA} := \psi_A(E_B)$		$E_{AB} := \psi_B(E_A)$
$U_A := \phi'_B(R_A)$, $V_A := \phi'_B(S_A)$		$U_B := \phi'_A(R_B)$, $V_B := \phi'_A(S_B)$
$\Psi_A \colon E'_B \to E'_B/\langle U_A + (\ell_A^{e_A-1} + \alpha)V_A\rangle$		$\Psi_B \colon E'_A \to E'_A/\langle U_B + (\ell_B^{e_B-1} + \beta)V_B\rangle$
$E'_{BA} := \Psi_A(E'_B)$		$E'_{AB} := \Psi_B(E'_A)$
$H(j(E_{BA}), j(E'_{BA}))$		$H(j(E_{AB}), j(E'_{AB}))$

Bob would also modify the kernel point K_B into K_B^*. Then, Alice would compute $E_B'^*$, U_A^*, V_A^* and get the value of $j(E_{BA}'^*) = j(E_B'^*/\langle U_A^* + (\ell_A^{e_A-1} + \alpha)V_B^*)$. The success of Bob's attack depends on whether he could pick K_B^*, R_A^* and S_A^* in such way that the possible values for $j(E_{BA}'^*)$ is sufficiently few. The security parameter λ should be large enough to prevent this.

Comparing to SIDH in Protocol B the bit-length of the chosen prime should at least be doubled as intermediate values are exposed. Moreover, the secret isogeny ϕ_A or ϕ_B are not calculated on full torsion points and the values of K_A or K_B are revealed during protocol instance which might raise concern on the computational hardness of computing ϕ_A or ϕ_B.

Problem B. *With the notation of Table 3, given E, P_A, Q_A, P_B, Q_B, E_A, K_A, R_A, S_A, E_B, K_B, R_B, S_B to determine $H(j(E_{AB}), j(E'_{AB}))$.*

6 Conclusion

In this note we present two different solutions to the static-static key exchange problem from supersingular isogenies featuring Diffie-Hellman style message flow. If these solutions are viable, Problem A and Problem B should be studied more extensively.

References

[1] Craig Costello, Patrick Longa, and Michael Naehrig. *Efficient algorithms for supersingular isogeny Diffie-Hellman*. In M. Robshaw and J. Katz (eds.), CRYPTO 2016, Part I, Springer LNCS 9814 (2016) 572–601.

[2] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. *Efficient compression of SIDH public keys*. In J.-S. Coron and J. B. Nielsen (eds.), EUROCRYPT 2017, Part I, Springer LNCS 10210 (2017) 679–706.

[3] Luca De Feo and David Jao. *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*. In Bo-Yin Yang, editor, PQCrypto 2011, Springer LNCS 7071 (2011) 19–34.

[4] Luca De Feo, David Jao, and Jérôme Plût. *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*. J. Mathematical Cryptology, **8(3)** (2014) 209–247.

[5] Steven. D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. *On the security of supersingular isogeny cryptosystems*. In J. H. Cheon and T. Takagi (eds.), ASIACRYPT 2016, Springer LNCS 10031 (2016) 63–91.

[6] Daniel Kirkwood, Bradley C. Lackey, John McVey, Mark Motley, Jerome A. Solinas, and David Tuller. *Failure is not an option: Standardization issues for post-quantum key agreement*, 2015, Workshop on Cybersecurity in a Post-Quantum World.

[7] Joseph H. Silverman. *The arithmetic of elliptic curves*, Second edition. Graduate Texts in Mathematics, 106. Springer, 2009.

[8] Reza Azarderakhsh, David Jao, and Christopher Leonardi. *Post-Quantum Static-Static Key Agreement Using Multiple Protocol Instances*. In Selected Areas in Cryptography – SAC 2017, Springer International Publishing (2018) 45–63.

[9] Steven D. Galbraith. *Authenticated key exchange for SIDH*. eprint 2018/266.

[10] Patrick Longa. *A Note on Post-Quantum Authenticated Key Exchange from Supersingular Isogenies*. eprint 2018/267.

[11] David Urbanik and David Jao. *SoK: The Problem Landscape of SIDH*. AsiaPKC 2018, (2018) 53–60.

[12] Joseph Silverman. *The Arithmetic of Elliptic Curves*.

[13] Jacques Vélu. *Isogénies entre courbes elliptiques*. C. R. Acad. Sci. Paris Sér. A-B 273 (1971), A238–A241.