Generating Supersingular Elliptic Curves over $\mathbb{F}_p$ with Unknown Endomorphism Ring

Youcef Mokrani, David Jao

Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
{ymokrani,djao}@math.uwaterloo.ca

Abstract. A number of supersingular isogeny based cryptographic protocols require the endomorphism ring of the initial elliptic curve to be either unknown or random in order to be secure. To instantiate these protocols, Basso et al. recently proposed a secure multiparty protocol that generates supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ of unknown endomorphism ring as long as at least one party acts honestly. However, there are many protocols that specifically require curves defined over $\mathbb{F}_p$, for which the Basso et al. protocol cannot be used. Also, the simple solution of using a signature scheme such as CSI-FiSh or SeaSign for proof of knowledge either requires extensive precomputation of large ideal class groups or is too slow for everyday applications.
In this paper, we present CSIDH-SCG, a new multiparty protocol that generates curves of unknown endomorphism ring defined over $\mathbb{F}_p$. This protocol relies on CSIDH-ROIP, a new CSIDH based proof of knowledge. We also present CSIDH-CR, a multiparty algorithm that be used in conjunction with CSIDH-SCG to generate a random curve over $\mathbb{F}_p$ while still keeping the endomorphism ring unknown.

Keywords: Elliptic curves · Supersingular curves · CSIDH · Multiparty computation

1 Introduction

Recent attacks on SIDH by Castryck, Decru, Maino, Martindale and Robert [6,14,18] have shown that torsion point information can be enough to find an isogeny between two supersingular elliptic curves. It follows that, for an isogeny based scheme to be secure, it must avoid giving too much information about its elliptic curves and isogenies.

One such piece of information is the endomorphism ring of the starting elliptic curve. In fact, the first break by Wouter Castryck and Thomas Decru [6] exploits this knowledge. We also note that Petit's torsion point attacks on SIDH [17] also need a known endomorphism ring. Since many current attacks on SIDH all make use of the endomorphism ring, it stands to reason that, unless necessary, a cryptographic protocol should avoid working on elliptic curves with a known endomorphism ring if possible. In addition, a number of existing schemes require

a supersingular curve of unknown endomorphishm ring. To solve this issue, Basso et al. [3] proposed a multiparty protocol that generates a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ as long as at least one participant acts honestly.

Although the Basso et al. protocol solves the problem in general, there remain a number of schemes that explicitly require a supersingular curve of unknown endomorphism ring defined over $\mathbb{F}_p$. As mentioned in [3], some examples of such protocols include CSIDH-based Verifiable Delay Functions [10], as well as Delay Encryption algorithms [5] that need to start with a random curve over $\mathbb{F}_p$. Such curves are also required for some Oblivious Transfer protocols [13] and dual mode PKE [1]. For such curves, the protocol found in [3] cannot be directly applied, as random walks in the supersingular isogeny graph have a negligible probability of ending on a curve defined over $\mathbb{F}_p$. Basso et al. [3] mention possible solutions, but they all come with important issues as they either leak too much information, require specific parameter sets or are too inefficient for everyday use. Another possible solution was proposed by Moriya, Takashima and Tagaki [15]. However, its security proof only deals with honest but curious participants (prover and verifier), and does not take into account the case where malicious adversaries send malformed data. Finally, a recent paper by Atapoor et al. [2] presents a distributed key generation protocol for CSIDH, but this situation differs from our scenario in that we are not trying to retain collective knowledge of any associated secret key.

In this paper, we present CSIDH-SCG, a new multiparty protocol that generates supersingular elliptic curves defined over $\mathbb{F}_p$ of unknown endomorphism ring as long as at least one of the participating parties is honest. CSIDH-SCG does not require the knowledge of any ideal class group, is efficient even for large groups, and resists active adversaries. We also present CSIDH-CR, a multiparty protocol taking the secure curve outputted by CSIDH-SCG and using it to generate a random supersingular elliptic curve of unknown endomorphism ring.

Section 2 presents the basic definitions and assumptions used in this paper. Past results related to this problem can be found in Section 3. We present two new CSIDH based zero-knowledge proofs in Section 4. Section 5 presents CSIDH-SCG, a new multiparty protocol generating curves of unknown endomorphism ring over $\mathbb{F}_p$, as well as CSIDH-ASCG, a variation avoiding the use of a random oracle function at the cost of an additional round of interaction. In Section 6, we present CSIDH-CR, which is used in addition to CSIDH-SCG or CSIDH-ASCG for cases where the desired curve needs to be random. Finally, Section 7 contains a brief summary of the results and possible avenues of further work.

2 Definitions and Assumptions

In this section, we present the various definitions and assumptions that are used at different points in this paper, as well as heuristics to justify said assumptions.

Since the protocols in this paper work with elliptic curves defined over $\mathbb{F}_p$ when the associated ideal class group is unknown, we start by presenting the necessary definitions for the sampling method used in CSIDH [7].

Notation 1 *Let p be a prime number and let E be a known supersingular elliptic curve defined over $\mathbb{F}_p$. We denote by $\mathcal{C}$ the ideal class group of the endomorphism ring of E.*

Notation 2 *In cases where $\mathcal{C}$ is unknown, let $\mathcal{S} = \{\mathfrak{l}_1, \ldots, \mathfrak{l}_t\} \subseteq C$ denote a generating set of $\mathcal{C}$ consisting of prime ideals of small (relatively prime) norm.*

Definition 3. *Let B be a positive integer. Define* CSIDHSample$(\mathcal{S}, B)$ *to be the procedure which outputs a random element of $\mathcal{C}$ using the following algorithm:*

CSIDHSample$(\mathcal{S}, B)$

$(e_1, \ldots, e_t) \leftarrow\!\!\$\ [-B, B]^t$

$\mathfrak{a} \leftarrow \prod_{i=1}^{t} \mathfrak{l}_i^{e_i}$

return $\mathfrak{a}$

We also need a basic notation for a set of nonces.

Notation 4 *Let N denote a known large set of nonces.*

Similarly to Basso et al. [3], we use a chain of secret isogenies to obtain an elliptic curve of unknown endomorphism ring. This idea is based on Wesolowski's theorem.

Theorem 1 ([19,20]). *Let IsogenyPath be the problem where, given two supersingular elliptic curves E and F, one must compute an isogeny $\phi : E \to F$. Let EndRing be the problem of computing the endomorphism ring of a supersingular elliptic curve E. Then IsogenyPath and EndRing can be polynomially reduced to each other.*

Since the goal of this paper is to generate elliptic curves of unknown endomorphism ring, we have to assume that computing such a ring is hard. With the above theorem, we will often use the problems of computing an isogeny and the problem of computing an endomorphism ring interchangeably and therefore also assume that computing an isogeny between two curves is hard.

To be more precise, the isogeny problem on which we base our protocol requires a stronger assumption.

Assumption 1 ([7]) *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$ of unknown endomorphism ring. Given $\mathfrak{a} \star E$ and $\mathfrak{b} \star E$ with unknown $\mathfrak{a}, \mathfrak{b} \in \mathcal{C}$, the CCISDH problem is to compute $\mathfrak{a} \star \mathfrak{b} \star E$. We assume that this problem is hard.*

The above assumption is required for (and equivalent to) the one-way security of CSIDH, and is therefore already widely accepted for many protocols that work with isogenies over $\mathbb{F}_p$. We also note that Assumption 1 implies that computing isogenies between two curves over $\mathbb{F}_p$ is hard.

Our second assumption is an adaptation of the Knowledge-of-Exponent Assumption (KEA) that was first described by Damgård [8]. This assumption was then used to create a classical Diffie-Hellman protocol by Wu and Stinson [21].

Assumption 2 *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$. For any probabilistic polynomial-time (PPT) $\boldsymbol{A}$ that takes as input E and $\mathfrak{b} \star E$, where $\mathfrak{b}$ is sampled using* $\text{CSIDHSAMPLE}(\mathcal{S}, B)$*, and which produces as output a pair of supersingular elliptic curves over $\mathbb{F}_p$ (F, F'), there exists probabilistic polynomial-time extractor $\boldsymbol{E}$ which takes the same input and outputs the same pair (F, F'), along with $\mathfrak{a}$, such the probability that $F' = \mathfrak{b} \star F$ and $\mathfrak{a} \star E \neq F$ is negligible.*

In other words, we assume that there is no pair of supersingular elliptic curves over $\mathbb{F}_p$ such that computing an isogeny between them is hard but computing a random CSIDH exchange involving them is easy. The original KEA is used in the context of classical Diffie-Hellman, while this new version is applied to CSIDH. While this assumption is new in the context of post-quantum cryptography, there is currently no known way to attack it.

Our third assumption simply states that we have access to a function H with some strong security properties. This same assumption is used by Basso et al. [3] for generating a multiparty protocol for secure curves over $\mathbb{F}_{p^2}$.

Assumption 3 ([3]) *We assume the existence of a function H which is a statistically hiding and computationally binding commitment scheme on the set of binary strings. Denote by $\mathcal{H}$ the codomain of H.*

The above three assumptions are enough to obtain an efficient protocol generating supersingular elliptic curves of unknown endomorphism rings. However, we can obtain a more efficient protocol using a random oracle function.

Assumption 4 *Assume that W is a random oracle function. Let $\mathcal{W}$ be the codomain of W. We also assume that $\mathcal{W}$ is large enough for collisions to be unfeasible to find and that the mapping of any input by W can be efficiently computed.*

To be more precise, for this paper, we require that, for any function f with domain X, the problem of distinguishing between $(f(x), W(x))$ for a random $x \in X$ and $(f(x), r)$, where r is a uniformly random element of $\mathcal{H}$ is equivalent to the problem of computing $x \in X$ when given $f(x)$.

In cases where we use H or W on arbitrary data, we implicitly assume that this data is encoded in the form of a binary string using a suitable encoding scheme.

The above assumptions are required for most of our protocols. These assumptions are enough for all our protocols except CSIDH-CR. Hence, if we only need to generate supersingular elliptic curves with unknown endomorphism rings over $\mathbb{F}_p$, we do not need to invoke Assumption 5 below. On the other hand, for our results on random curve generation (Section 6), we require the following assumption, which was considered in the SeaSign paper [9, p.766].

Assumption 5 *The output distribution of* $\text{CSIDHSAMPLE}(\mathcal{S}, B)$ *is indistinguishable from the uniform distribution on the ideal class group* $\mathcal{C}$ *of* $\text{End}(E)$.

3 Existing Solutions

In this section, we present a brief overview of possible ways to generate supersingular elliptic curves of unknown endomorphism ring that were either presented or mentioned in previous papers.

3.1 Signature Schemes

In the model used by Basso et al. [3], we have n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ whose goal is to generate a supersingular elliptic curve of unknown endomorphism ring. A starting elliptic curve E_0 is provided.

Their idea is to have each party $\mathcal{P}_i$ in turn compute a random isogeny $\phi_i : E_{i-1} \to E_i$ and publish E_i. If all parties act honestly and do not share their secret isogenies, then the endomorphism ring is unknown to them all by Theorem 1.

Of course, the above is not sufficient against dishonest adversaries, as nothing stops $\mathcal{P}_n$ from choosing the curve of their choice as E_n and lying about their isogeny. To solve this issue, [3] proposed having each party prove their knowledge of their claimed isogeny by publishing a Fiat-Shamir signature of a zero-knowledge proof.

Using a signature as a proof of knowledge has the advantage of keeping the number of required interactions to a minimum. The one proposed in [3], in particular, is fast enough for the desired application over $\mathbb{F}_{p^2}$.

However, the zero-knowledge proof proposed in [3] reveals the degree of the secret isogeny. While it does not create issues when working over $\mathbb{F}_{p^2}$, this leaks too much information when dealing with isogenies defined over $\mathbb{F}_p$. This is because isogenies defined over $\mathbb{F}_p$ are usually sampled using CSIDHSAMPLE. Therefore, the isogeny degree can be used to efficiently compute $|e_i|$ and this knowledge massively reduces the possible key space. Because of this, a different signature scheme would need to be used to prove the knowledge of the claimed isogenies.

Basso et al. mention the possible use of either SeaSign [9] or CSI-FiSh [4] as possible replacement signatures. While both work in theory, they each come with issues limiting their practical applications. While SeaSign is a zero-knowledge signature, its current computation times are way too long to be used for everyday applications. However, it is worth noting that, in cases where a single secure

curve needs to be generated by parties that can afford to wait multiple hours, for example when generating secure parameters for a scheme, using SeaSign is a possible solution.

On the other hand, CSI-FiSh can potentially be both zero-knowledge and efficient. However, it requires full knowledge of the ideal class group associated with the chosen parameter set. Currently, the parameter sets for which the ideal class group is known are pretty limited and, as discussed by Panny [16], computing new ones would require an extensive amount of computation even with access to a quantum computer. While recent results presented in the SCALLOP paper [11] have expanded the number of parameter sets that can be used today, the complexity of finding new ones is still super-polynomial.

3.2 Multiparty Key Generation

Two other possible ideas to generate secure curves defined over $\mathbb{F}_p$ were proposed by Moriya, Takashima and Tagaki [15]. However, the adversarial model in that paper is honest but curious, and this creates issues when trying to adapt their techniques when dealing with active adversaries.

The core idea of the protocol, given n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$, is to have n chains of isogenies that all loop over the same set of ideal class group elements so that, for each party, there is a chain where they are the last to apply their group action. This then implies that each party can trust the security of one chain and that, since the commutativity of the ideal class group implies that all chains end at the same curve, they can all trust the security of the final curve.

While the security of the above scheme is not proven against active adversaries in [15], such a proof might be possible. However, another issue is that the number of required interactions grows quadratically in proportion to the number of parties, making the scheme inefficient when working with a large number of parties.

4 A New Zero-Knowledge Proof

As mentioned in the previous session, there is currently no known fast CSIDH based signature scheme that works with any parameter set without heavy precomputation or the use of a quantum computer.

To get around this issue, we propose replacing the signature part of the multiparty protocol with an interactive zero-knowledge proof.

The following proposal, CSIDH Random Oracle Interactive Proof (CSIDH-ROIP), requires Assumptions 1, 4 and 2 for its security proof. Note that, in contrast to Moriya et al. [15], we allow for malicious parties.

Definition 5 (CSIDH-ROIP). *Let a prover P know a secret $\mathfrak{a} \in \mathcal{C}$ with associated public data $(E, E_{\mathcal{P}} := \mathfrak{a} \star E)$.*

The goal of the following protocol is for $\mathcal{P}$ to prove their knowledge of $\mathfrak{a}$ to a verifier $\mathcal{V}$ without leaking any extra information.

CSIDH-ROIP consists of the followings steps:

1. *Challenge: $\mathcal{V}$ sends the challenge curve $E_{\mathcal{V}}$.*
2. *Response: $\mathcal{P}$ computes $\mathfrak{a} \star E_{\mathcal{V}}$ and publishes a masked version of it using a random oracle function.*
3. *Verification: $\mathcal{V}$ verifies $\mathcal{P}$'s answer.*

Challenge($\mathcal{P}$,$\mathcal{V}$,E,$E_{\mathcal{P}}$)	*Response($\mathcal{P}$,$\mathcal{V}$,E,$E_{\mathcal{P}}$,$\mathfrak{a}$,$E_{\mathcal{V}}$)*
$\mathfrak{b} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$	$E_{\mathcal{P},\mathcal{V}} \leftarrow \mathfrak{a} \star E_{\mathcal{V}}$
$E_{\mathcal{V}} \leftarrow \mathfrak{b} \star E$	$M_{\mathcal{P},\mathcal{V}} \leftarrow W(\mathcal{P}, E_{\mathcal{P},\mathcal{V}})$
return $E_{\mathcal{V}}$	**return** $E_{\mathcal{P},\mathcal{V}}$

Verification($\mathcal{P}$,$\mathcal{V}$,E,$E_{\mathcal{P}}$,$E_{\mathcal{V}}$,$\mathfrak{b}$,$M_{\mathcal{P},\mathcal{V}}$)

$M'_{\mathcal{P},\mathcal{V}} \leftarrow W(\mathcal{P}, \mathfrak{b} \star E_{\mathcal{P}})$
if $M'_{\mathcal{P},\mathcal{V}} = M_{\mathcal{P},\mathcal{V}}$: **return true**
else : **return false**

The correctness of CSIDH-ROIP comes from the fact that $\mathfrak{a} \star (\mathfrak{b} \star E) = \mathfrak{b} \star (\mathfrak{a} \star E)$. Its soundness and zero-knowledge properties come from the following theorems.

Theorem 2. *Given Assumption 4, CSIDH-ROIP is zero-knowledge.*

Proof. We can separate the adversarial strategies as the prover in two cases. Either they send a challenge $E_{\mathcal{V}}$ for which they know the associated $\mathfrak{b}$ or they choose a curve for whose associated group element is unknown and not feasibly computable.

In the former case, for any chosen $\mathfrak{b}$, the associated simulator is simple as an honest verifier knows the correct answer without needing to interact with the prover.

In the latter case, we use the fact that W is a random oracle function to simulate transcripts indistinguishable from honest ones. This time, the adversary can use any method they desire to sample $E_{\mathcal{V}}$, but they do not gain any usable information from the answer as it is masked by W.

As we have two cases, we present two simulators. SIMULATOR1 represents the first case while SIMULATOR2 represents the second. The simulator returns a challenge-response pair with the same distribution as that of an honest exchange.

Simulator1($\mathcal{P}$,E,$E_{\mathcal{P}}$, $\mathfrak{b}$)	Simulator2($\mathcal{P}$,E,$E_{\mathcal{P}}$, $E_{\mathcal{V}}$)
$E_{\mathcal{V}} \leftarrow \mathfrak{b} \star E$	Challenge $\leftarrow E_{\mathcal{V}}$
Challenge $\leftarrow E_{\mathcal{V}}$	$M_{\mathcal{P},\mathcal{V}} \leftarrow\!\!\$\ \mathcal{W}$
$M_{\mathcal{P},\mathcal{V}} \leftarrow W(\mathcal{P}, \mathfrak{b} \star E_{\mathcal{P}})$	Response $\leftarrow E_{\mathcal{P},\mathcal{V}}$
Response $\leftarrow E_{\mathcal{P},\mathcal{V}}$	**return** (Challenge, Response)
return (Challenge, Response)	

Theorem 3. *CSIDH-ROIP is computationally sound under Assumptions 1, 4 and 2.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) algorithm able to generate valid responses $M_{\mathcal{P},\mathcal{V}}$ for random challenges $E_{\mathcal{V}}$.

Since W is a random oracle function, computing a valid $M_{\mathcal{P},\mathcal{V}}$ is equivalent to computing the correct $E_{\mathcal{P},\mathcal{V}}$ associated with the given challenge. This implies the existence of a PPT algorithm $\mathcal{A}'$ capable of generating $E_{\mathcal{P},\mathcal{V}}$ when given $E_{\mathcal{V}}$.

By Assumption 2, this then implies the existence a PPT extractor capable of computing a valid witness $\mathfrak{a}$.

4.1 Avoiding the Random Oracle Model

As mentioned in Section 2, it is possible to generate supersingular elliptic curves over $\mathbb{F}_p$ with unknown endomorphism ring without having to rely on a random oracle function. The core idea of this trick is to remark that, without the use of W, CSIDH-ROIP is still honest verifier zero-knowledge. Therefore, we only need a way to deal with dishonest verifiers.

CSIDH-ROIP does so by publishing information unusable by dishonest verifiers, but this is not the only way to proceed. Another way is to have the verifiers prove that they were honest at the end of the proof. This can be done by requiring verifiers to publish a masked commitment for their challenge using a statistically hiding and computationally binding protocol H.

Of course, an adversary might still lie about their commitment and not care about being found cheating, as they still obtain information from the prover's answer. However, when it comes to using our proof scheme to generate secure curves, once the honest parties detect that someone cheated, the guilty party can then simply be removed from the protocol and the honest parties can then restart the protocol using new random values. As long as the new values are independent of the previous ones, the adversary gains no information by cheating.

Definition 6 (CSIDH-AIP). *Let a prover $\mathcal{P}$ know a secret $\mathfrak{a} \in \mathcal{C}$ with associated public data $(E, E_{\mathcal{P}} := \mathfrak{a} \star E)$.*

The goal of the following protocol is for $\mathcal{P}$ to prove their knowledge of $\mathfrak{a}$ to a verifier $\mathcal{V}$ without leaking any extra information.

CSIDH-AIP consists of the followings steps:

1. *Challenge: $\mathcal{V}$ sends the challenge curve $E_{\mathcal{V}}$ and commits the associated $\mathfrak{b}$ using H.*
2. *Response: $\mathcal{P}$ computes $\mathfrak{a} \star E_{\mathcal{V}}$ and publishes it.*
3. *Verification1: $\mathcal{V}$ verifies $\mathcal{P}$'s answer and, if it is valid, publishes $\mathfrak{b}$.*
4. *Verification2: $\mathcal{P}$ verifies that $\mathcal{V}$'s commitment coincides with the challenge.*

The details of the protocol are as follows:

Challenge($\mathcal{V}$,E)	*Response($\mathcal{P}$,$\mathfrak{a}$,$E_{\mathcal{V}}$)*
$\mathfrak{b} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$	$E_{\mathcal{P},\mathcal{V}} \leftarrow \mathfrak{a} \star E_{\mathcal{V}}$
$E_{\mathcal{V}} \leftarrow \mathfrak{b} \star E$	**return** $E_{\mathcal{P},\mathcal{V}}$
$r \leftarrow\$ N$	
$C \leftarrow H(\mathfrak{b}, r)$	
return $(E_{\mathcal{V}}, C)$	

Verification1($\mathcal{V}$,$E_{\mathcal{P}}$,$\mathfrak{b}$,r,$E_{\mathcal{P},\mathcal{V}}$)	*Verification2*($\mathcal{P}$,E,$E_{\mathcal{V}}$,$\mathfrak{b}$,r,C)
$E'_{\mathcal{P},\mathcal{V}} \leftarrow \mathfrak{b} \star E_{\mathcal{P}}$	**if** $C \neq H(\mathfrak{b}, r)$: **return false**
if $E'_{\mathcal{P},\mathcal{V}} = E_{\mathcal{P},\mathcal{V}}$: **return** $(\mathfrak{b}, r)$	$E'_{\mathcal{V}} \leftarrow \mathfrak{b} \star E$
else : **return false**	**if** $E'_{\mathcal{V}} \neq E_{\mathcal{V}}$: **return false**
	return true

The proof fails if either Verification1 or Verification2 returns **false** *and succeeds otherwise.*

The correctness of CSIDH-AIP holds for the same reason as CSIDH-ROIP's. It is also sound and zero-knowledge, as the following theorems show.

Theorem 4. *Given Assumption 3, if it does not abort, CSIDH-AIP is zero-knowledge.*

Proof. Since H is computationally binding, $\mathcal{V}$ must compute $E_{\mathcal{V}}$ by first choosing a valid $\mathfrak{b}$. By the same reasoning, they must also choose their nonce r in advance.

The following simulator returns a challenge-response-verification1 triple for any challenge constructed using $(\mathfrak{b}, r)$.

Simulator(E,$E_{\mathcal{P}}$, $\mathfrak{b}$, r)

$E_{\mathcal{V}} \leftarrow \mathfrak{b} \star E$

$C \leftarrow H(\mathfrak{b}, r)$

Challenge $\leftarrow (E_{\mathcal{V}}, C)$

$E_{\mathcal{P},\mathcal{V}} \leftarrow \mathfrak{b} \star E_{\mathcal{P}}$

Response $\leftarrow E_{\mathcal{P},\mathcal{V}}$

verification1 $\leftarrow (\mathfrak{b}, r)$

return (Challenge, Response, verification1)

Theorem 5. *CSIDH-AIP is computationally sound under Assumptions 1, 2 and 3.*

Proof. Since H is statistically hiding, from $\mathcal{P}$'s point of view C can be replaced with a random value and, therefore, gives no advantage when it comes to beating the soundness property. Without being able to make use of this extra information, successfully generating a valid CSIDH-AIP proof becomes equivalent to generating $\mathfrak{a} \star \mathfrak{b} \star E$ when given $\mathfrak{b} \star E$ and $\mathfrak{a} \star E$.

By Assumption 2, if there was a PPT algorithm capable of doing so, then there would be another PPT algorithm capable of solving the CCISDH problem, which we assume is hard.

5 Secure Curve Generation

With the help of CSIDH-ROIP, we can now present our multiparty protocol for generating supersingular elliptic curves defined over $\mathbb{F}_p$ with unknown endomorphism ring. As its security mostly relies on CSIDH-ROIP, this new protocol, which we call CSIDH Secure Curve Generator (CSIDH-SCG), requires Assumptions 1 2 and 4, but not Assumption 5.

Definition 7 (CSIDH-SCG). *Let $\mathcal{P}_1, \ldots, \mathcal{P}_n$ be n parties that want to generate a supersingular elliptic curve over $\mathbb{F}_p$ with unknown endomorphism ring. Let E_0 be a known supersingular curve over the same field. CSIDH-SCG consists of the following steps.*

- ***CurveGen:*** *For i from 1 to n, party $\mathcal{P}_i$ computes an ideal class group element $\mathfrak{a}_i$, saves it, and publishes $E_i := \mathfrak{a}_i \star E_{i-1}$.*
- ***Challenge:*** *For each $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ sends a CSIDH-ROIP challenge $E_{i,j}$ to $\mathcal{P}_j$ and saves the associated $\mathfrak{b}_{i,j}$.*
- ***Response:*** *After received a challenge from every $\mathcal{P}_j$ such that $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ publishes a CSIDH-ROIP response $M'_{i,j}$ for each of them.*
- ***Verification:*** *After receiving the response to all their challenges, $\mathcal{P}_i$ checks their validity. If every response is correct, $\mathcal{P}_i$ published* **true** *and then accepts E_n as the final curve if every other party also publishes* **true***. Otherwise, $\mathcal{P}_i$ publishes* **false** *and aborts the entire protocol.*
- ***Abort:*** *If at any point, a party published* **false***, the protocol is aborted and every party must publish all their computed values. Any dishonest parties are thereby revealed.*

The algorithms for each step are as follows:

CurveGen$(\mathcal{P}_i, E_{i-1})$	*Challenge*$(\mathcal{P}_i, \mathcal{P}_j, E_{j-1})$
$\mathfrak{a}_i \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$	$\mathfrak{b}_{i,j} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$
$E_i \leftarrow \mathfrak{a}_i \star E_{i-1}$	$E_{i,j} \leftarrow \mathfrak{b}_{i,j} \star E_{j-1}$
return E_i	**return** $E_{i,j}$
Response$(\mathcal{P}_i, \mathcal{P}_j, \mathfrak{a}_i, E_{j,i})$	*Verification*$(\mathcal{P}_i, \mathcal{P}_j, M'_{i,j}, E_j)$
$M'_{j,i} \leftarrow W(\mathcal{P}_i, \mathfrak{a}_i \star E_{j,i})$	$M''_{i,j} \leftarrow W(\mathcal{P}_j, \mathfrak{b}_{i,j} \star E_j)$
return $M'_{j,i}$	**if** $M''_{i,j} = M'_{i,j}$: **return true**
	else : **return false**

Notation 8 (Secure Curve Generation Adversary) *Given a secure curve generation multiparty protocol with n parties, the adversary is denoted $\mathcal{A}_{SCG}$.*

The goal of $\mathcal{A}_{SCG}$ is to compute the endomorphism ring of the final curve E_n.

$\mathcal{A}_{SCG}$ is able to take control of all parties but one, say $\mathbb{P}_i$. They can try to be dishonest during the multiparty protocol. However, they fail if the protocol is aborted.

Theorem 6. *Given Assumptions 1, 2, and 4, CSIDH-SCG is secure against $\mathcal{A}_{SCG}$ adversaries.*

Proof. During CSIDH-SCG, every party must prove knowledge of their $\mathfrak{a}_i$ using $n-1$ parallel CSIDH-ROIP proofs.

By Theorem 2, $\mathcal{A}_{SCG}$ gains no information about $\mathfrak{a}_i$.

By Theorem 3, $\mathcal{A}_{SCG}$ cannot lie about any of their $\mathfrak{a}_j$.

Since $\mathcal{A}_{SCG}$ knows every $\mathfrak{a}_j$ expect for $\mathfrak{a}_i$, computing the endomorphism ring of E_n is equivalent to computing the endomorphism ring of E_i. However, by Theorem 1, doing so is equivalent to computing $\mathfrak{a}_i$, which is hard given Assumption 1.

In addition to being secure, CSIDH-SCG is also efficient, even when considering large groups. This is true for both the computation time and the number of interactions.

In practice, the CSIDH parameters are chosen so that both sampling and group actions are computed efficiently. W is also assumed to be efficiently computable. In CSIDH-SCG, the number of times each party must compute a group action grows linearly in terms of the number of participants. The same is true for the number of times each party must call the functions W and CSIDHSample.

When it comes to the number of operations, CSIDH-SCG is constructed in a way that every step of the proof can be done in parallel with every other party. Because of this, each party is only required to publish once for each of the four steps of CSIDH-SCG, making the total number of interactions linear in terms of the number of participants.

5.1 Generating Secure Curves Without a Random Oracle

In cases where we want to avoid needing a random oracle, we can use CSIDH-AIP instead of CSIDH-ROIP.

Definition 9 (CSIDH-ASCG). *Let $\mathcal{P}_1, \ldots, \mathcal{P}_n$ be n parties that want to generate a supersingular elliptic curve over $\mathbb{F}_p$ with unknown endomorphism ring. Let E_0 be a known supersingular curve over the same field. CSIDH-ASCG consists of the following steps.*

- ***CurveGen:*** *For i from 1 to n, party $\mathcal{P}_i$ computes an ideal class group element $\mathfrak{a}_i$, saves it, and publishes $E_i := \mathfrak{a}_i \star E_{i-1}$.*
- ***Challenge:*** *For each $j \in [n] \backslash \{i\}$, $\mathcal{P}_i$ sends a CSIDH-AIP challenge $(E_{i,j}, C_{i,j})$ to $\mathcal{P}_j$ and saves the associated $(\mathfrak{b}_{i,j}, r_{i,j})$.*
- ***Response:*** *After received a challenge from every $\mathcal{P}_j$ such that $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ publishes a CSIDH-AIP response $E'_{i,j}$ for each of them.*
- ***Verification1:*** *After receiving the response to all their challenges, $\mathcal{P}_i$ checks their validity. If every response is correct, $\mathcal{P}_i$ publishes all their pairs $(\mathfrak{b}_{i,j}, r_{i,j})$. Otherwise, $\mathcal{P}_i$ publishes* **false** *and aborts the entire protocol.*
- ***Verification2:*** *After receiving every pair $(\mathfrak{b}_{j,i}, r_{j,i})$, $\mathcal{P}_i$ verifies that each pair agrees with their challenges. If that is the case for every pair, $\mathcal{P}_i$ publishes* **true**. *Otherwise, $\mathcal{P}_i$ publishes* **false** *and aborts the entire protocol. If every party publishes* **true**, *E_n is accepted as the curve of unknown endomorphism ring.*
- ***Abort:*** *If at any point, a party published* **false**, *the protocol is aborted and every party must publish all their computed values. Any dishonest parties are thereby revealed.*

The algorithms for each step are as follows:

CurveGen($\mathcal{P}_i$,E_{i-1})

$\mathfrak{a}_i \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$
$E_i \leftarrow \mathfrak{a}_i \star E_{i-1}$
return E_i

Challenge($\mathcal{P}_i$,$\mathcal{P}_j$,E_{j-1})

$\mathfrak{b}_{i,j} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$
$E_{i,j} \leftarrow \mathfrak{b}_{i,j} \star E_{j-1}$
$r_{i,j} \leftarrow\!\!\$\ N$
$C_{i,j} \leftarrow H(\mathfrak{b}_{i,j}, r_{i,j})$
return $(E_{i,j}, C_{i,j})$

Response($\mathcal{P}_i$,$\mathcal{P}_j$,$\mathfrak{a}_i$,$E_{j,i}$)

$E'_{j,i} \leftarrow \mathfrak{a}_i \star E_{j,i}$
return $E'_{j,i}$

Verification1($\mathcal{P}_i$,$\mathcal{P}_j$,$E'_{i,j}$,E_j,$\mathfrak{b}_{i,j}$,$r_{i,j}$)

$E''_{i,j} \leftarrow \mathfrak{b}_{i,j} \star E_j$
if $E''_{i,j} = E'_{i,j}$: **return** $(\mathfrak{b}_{i,j}, r_{i,j})$
else : **return false**

Verification2($\mathcal{P}_i$,$\mathcal{P}_j$,$C_{j,i}$,$\mathfrak{b}_{j,i}$,$r_{j,i}$,E_{i-1},$E_{j,i}$)

if $C_{j,i} \neq H(\mathfrak{b}_{j,i}, r_{j,i})$: **return false**
$E'''_{j,i} \leftarrow \mathfrak{b}_{j,i} \star E_{i-1}$
if $E'''_{j,i} \neq E_{j,i}$: **return false**
return true

Theorem 7. *Given Assumptions 1, 3, and 2, CSIDH-ASCG is secure against $\mathcal{A}_{SCG}$ adversaries.*

Proof. During CSIDH-ASCG, every party must prove knowledge of their $\mathfrak{a}_i$ using $n-1$ parallel CSIDH-AIP proofs.

By Assumption 3, since H is statistically hiding, $\mathcal{A}_{SCG}$ gains no information from $\mathcal{P}_i$ publishing $C_{i,j}$. This implies that $\mathcal{A}_{SCG}$ must prove knowledge of its $\mathfrak{a}_j$ using CSIDH-AIP without any extra information.

By Theorem 4, since the adversary looses if they cause an abort, $\mathcal{A}_{SCG}$ gains no information about $\mathfrak{a}_i$.

By Theorem 5, $\mathcal{A}_{SCG}$ cannot lie about any of their $\mathfrak{a}_j$.

Since $\mathcal{A}_{SCG}$ knows every $\mathfrak{a}_j$ expect for $\mathfrak{a}_i$, computing the endomorphism ring of E_n is equivalent to computing the endomorphism ring of E_i. However, by Theorem 1, doing so is equivalent to computing $\mathfrak{a}_i$, which is hard given Assumption 1.

While efficient and secure, CSIDH-ASCG comes with two disadvantages compared CSIDH-SCG. First, since there are two verification steps in CSIDH-AIP, CSIDH-ASCG requires one more round of interaction than CSIDH-SCG. The other issue is that CSIDH-AIP is only zero-knowledge if there are no aborts. Because of this, every time a dishonest party is found, the entire process needs to be restarted with new random values. In practice, needing a random oracle function assumption is usually worth it for the efficiency gains of CSIDH-SCG.

6 Curve Randomizer

CSIDH-SCG allows us to generate a supersingular elliptic curve of unknown endomorphism ring. However, E_n is not uniformly random, as $\mathcal{P}_n$ has some control

over what the final curve is. This limitation also appears in the protocol proposed by Basso et al. [3]. However, some protocols (for example Delay Encryption [5]) explicitly ask for a uniformly random (or at least nearly uniform) curve over $\mathbb{F}_p$.

The core idea of CSIDH Curve Randomizer (CSIDH-CR) is to use a multiparty commitment scheme to generate some random data and then convert that data into a random isogeny whose codomain is chosen as the random curve.

CSIDH-CR is the only protocol in this paper that requires Assumption 5. On the other hand, it does not require Assumption 1.

Definition 10 (CSIDH-CR). *Let $\mathcal{P}_1, \ldots, \mathcal{P}_n$ be n parties that want to generate a random supersingular elliptic curve over $\mathbb{F}_p$. Let E_n be supersingular elliptic curve of unknown endomorphism ring defined over the same field. CSIDH-SCG consists of the following steps.*

- ***RandomSample:*** *Each $\mathcal{P}_i$ samples a random $\mathfrak{a}'_i$ and nonce r'_i and commits $H(\mathfrak{a}'_i, r'_i)$.*
- ***CurveComp:*** *Once every commitment has been published, each party publishes their pair $(\mathfrak{a}'_i, r'_i)$. If every pair corresponds to their commitment, the final curve is chosen to be $(\prod_{i=1}^{n} \mathfrak{a}'_i) \star E_n$.*
- ***Abort:*** *If any of the pairs do not correspond to their commitments, the protocol aborts and any dishonest parties are thereby revealed.*

The algorithms for each step are as follows:

RandomSample($\mathcal{P}_i$)

$\mathfrak{a}'_i \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$

$r'_i \leftarrow\!\!\$\ N$

$C'_i \leftarrow H(\mathfrak{a}'_i, r'_i)$

return C'_i

CurveComp(E_n,$(C'_1, \ldots, C'_n)$,$((\mathfrak{a}'_1, r'_1), \ldots, (\mathfrak{a}'_1, r'_1))$)

$F \leftarrow E_n$

for $j \in [n]$:

  if $C'_j \neq H(\mathfrak{a}'_j, r'_j)$: **return false**

  $F \leftarrow \mathfrak{a}'_j \star F$

return F

Notation 11 (Curve Randomizer Adversary) *Given a curve randomizer multiparty protocol with n parties and whose initial curve E_n has an unknown endomorphism ring, the adversary is denoted $\mathcal{A}_{CR}$.*

The goal of $\mathcal{A}_{CR}$ is either to guess the final curve F before starting the scheme or to compute its endomorphism ring.

$\mathcal{A}_{CR}$ is able to take control of all parties but one, say $\mathbb{P}_i$. They can try to be dishonest during the multiparty protocol. However, they fail if the protocol is aborted.

Theorem 8. *Given Assumptions 3 and 5, CSIDH-CR is secure against $\mathcal{A}_{CR}$ adversaries.*

Proof. By Assumption 3, H is statistically hiding and $\mathcal{A}_{CR}$ gains no information from C'_i. By the same assumption, since H is a binding commitment scheme, $\mathcal{A}_{CR}$ must choose their $\mathfrak{a}'_j$ before knowing anything about $\mathfrak{a}'_i$.

Once the $\mathfrak{a}'_j$ have been chosen, set $F' := \left(\prod_{j\in[n]\setminus\{j\}} \mathfrak{a}'_j\right) \star E_n$.

By Assumption 5, $\mathfrak{a}'_i$ is indistinguishable from a uniformly random ideal class group element. Since $\mathcal{C}$ is an Abelian group, we have that $F = \mathfrak{a}'_i \star F'$. Therefore, $\mathfrak{a}'_i$ is indistinguishable from random, and so is F.

Also, since every $\mathfrak{a}'_i$ is revealed at the end of the protocol, computing the endomorphism ring of F is equivalent to computing the endomorphism ring of E_n, which is hard.

7 Conclusion

CSIDH-SCG enables efficient generation of supersingular elliptic curves defined over $\mathbb{F}_p$ with unknown endomorphism ring. In analogy to the work of Basso et al. for curves over $\mathbb{F}_{p^2}$ [3], the total number of interactions required for CSIDH-SCG grows linearly in terms of the number of participants. Currently, the greatest limitation of CSIDH-SCG is the need of a Knowledge of Exponent Assumption. While such an assumption is sometimes used in classical schemes, this new CSIDH variant requires further study on its security.

The other limitation of CSIDH-SCG is its need of a random oracle function. However, this can be dealt with by using CSIDH-ASCG. While this alternative is strictly less efficient, the increase in computation time is only a single additional round of interactions.

Given an additional assumption on the randomness of CSIDH samples, CSIDH-CR makes it possible for the generated curve to be random.

It is worth mentioning that the curve randomizer structure can also be adapted to generate random supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Using the Ramanujan property of supersingular isogeny graphs defined over $\mathbb{F}_{p^2}$, Jao, Miller and Venkatesan [12] showed that the codomain of a random isogeny of large enough degree is indistinguishable from random. It is therefore possible to use a multiparty protocol to generate random data, which can then be converted into a random isogeny in order to obtain a random supersingular elliptic curve. We leave the implementation of such a protocol for future work.

By itself, CSIDH-ROIP is a secure and efficient zero-knowledge proof that can be used with any CSIDH parameter sets. While its structure makes it so that it cannot be used in a signature scheme, its large challenge space makes it so that a single run of CSIDH-ROIP is enough to achieve levels of security comparable to CSIDH given a KEA assumption.

8 Acknowledgments

We would like to thank Andrea Basso for helpful comments on a draft version of this paper. This research was supported by NSERC Alliance Consortia Quantum Grant ALLRP 578463–2022.

References

1. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 411–439. Springer International Publishing, Cham (2020)
2. Atapoor, S., Baghery, K., Cozzo, D., Pedersen, R.: Practical robust dkg protocols for csidh. In: Tibouchi, M., Wang, X. (eds.) Applied Cryptography and Network Security. pp. 219–247. Springer Nature Switzerland, Cham (2023)
3. Basso, A., Codogni, G., Connolly, D., De Feo, L., Fouotsa, T.B., Lido, G.M., Morrison, T., Panny, L., Patranabis, S., Wesolowski, B.: Supersingular curves you can trust. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. pp. 405–437. Springer Nature Switzerland, Cham (2023)
4. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient isogeny based signatures through class group computations. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 227–247. Springer International Publishing, Cham (2019)
5. Burdges, J., De Feo, L.: Delay encryption. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 302–326. Springer International Publishing, Cham (2021)
6. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. pp. 423–447. Springer Nature Switzerland, Cham (2023)
7. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: Csidh: An efficient post-quantum commutative group action. In: Advances in Cryptology – ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III. p. 395–427. Springer-Verlag, Berlin, Heidelberg (2018). https://doi.org/10.1007/978-3-030-03332-3˙15
8. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) Advances in Cryptology — CRYPTO '91. pp. 445–456. Springer Berlin Heidelberg, Berlin, Heidelberg (1992)
9. De Feo, L., Galbraith, S.D.: SeaSign: Compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019. pp. 759–789. Springer International Publishing, Cham (2019)
10. De Feo, L., Masson, S., Petit, C., Sanso, A.: Verifiable delay functions from supersingular isogenies and pairings. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 248–277. Springer International Publishing, Cham (2019)
11. Feo, L.D., Fouotsa, T.B., Kutas, P., Leroux, A., Merz, S.P., Panny, L., Wesolowski, B.: Scallop: Scaling the csi-fish. In: Boldyreva, A., Kolesnikov, V. (eds.) Public-Key Cryptography – PKC 2023. pp. 345–375. Springer Nature Switzerland, Cham (2023)
12. Jao, D., Miller, S.D., Venkatesan, R.: Expander graphs based on GRH with an application to elliptic curve cryptography. Journal of Number Theory **129**(6), 1491–1504 (2009). https://doi.org/10.1016/j.jnt.2008.11.006
13. Lai, Y.F., Galbraith, S.D., Delpech de Saint Guilhem, C.: Compact, efficient and UC-secure isogeny-based oblivious transfer. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 213–241. Springer International Publishing, Cham (2021)

14. Maino, L., Martindale, C.: An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Paper 2022/1026 (2022)
15. Moriya, T., Takashima, K., Takagi, T.: Group key exchange from csidh and its application to trusted setup in supersingular isogeny cryptosystems. In: Liu, Z., Yung, M. (eds.) Information Security and Cryptology. pp. 86–98. Springer International Publishing, Cham (2020)
16. Panny, L.: Csi-fish really isn't polynomial-time, https://yx7.cc/blah/2023-04-14.html
17. Petit, C.: Faster algorithms for isogeny problems using torsion point images. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 330–353. Springer International Publishing, Cham (2017)
18. Robert, D.: Breaking sidh in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. pp. 472–503. Springer Nature Switzerland, Cham (2023)
19. Wesolowski, B.: The supersingular isogeny path and endomorphism ring problems are equivalent. 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS) pp. 1100–1111 (2021)
20. Wesolowski, B.: Orientations and the supersingular endomorphism ring problem. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 345–371. Springer International Publishing, Cham (2022)
21. Wu, J., Stinson, D.R.: An efficient identification protocol and the knowledge-of-exponent assumption. Cryptology ePrint Archive, Paper 2007/479 (2007), https://eprint.iacr.org/2007/479