# Oblivious Pseudorandom Functions from Isogenies

Dan Boneh[1], Dmitry Kogan[1], and Katharine Woo[1,2]

[1] Stanford University, Stanford, CA, USA
{dabo,dkogan}@cs.stanford.edu
[2] Princeton University, Princeton, NJ, USA
khwoo@princeton.edu

**Abstract.** An oblivious PRF, or OPRF, is a protocol between a client and a server, where the server has a key $k$ for a secure pseudorandom function $F$, and the client has an input $x$ for the function. At the end of the protocol the client learns $F(k, x)$, and nothing else, and the server learns nothing. An OPRF is verifiable if the client is convinced that the server has evaluated the PRF correctly with respect to a prior commitment to $k$. OPRFs and verifiable OPRFs have numerous applications, such as private-set-intersection protocols, password-based key-exchange protocols, and defense against denial-of-service attacks. Existing OPRF constructions use RSA-, Diffie-Hellman-, and lattice-type assumptions. The first two are not post-quantum secure.

In this paper we construct OPRFs and verifiable OPRFs from isogenies. Our main construction uses isogenies of supersingular elliptic curves over $\mathbb{F}_{p^2}$ and tries to adapt the Diffie-Hellman OPRF to that setting. However, a recent attack on supersingular-isogeny systems due to Galbraith et al. [ASIACRYPT 2016] makes this approach difficult to secure. To overcome this attack, and to validate the server's response, we develop two new zero-knowledge protocols that convince each party that its peer has sent valid messages. With these protocols in place, we obtain an OPRF in the SIDH setting and prove its security in the UC framework.

Our second construction is an adaptation of the Naor-Reingold PRF to commutative group actions. Combining it with recent constructions of oblivious transfer from isogenies, we obtain an OPRF in the CSIDH setting.

## 1 Introduction

Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure pseudorandom function (PRF) [30]. An *oblivious PRF*, or *OPRF*, is a protocol between a client who has an input $x \in \mathcal{X}$, and a server who has a key $k \in \mathcal{K}$. At the end of the protocol the client learns $F(k, x)$ and nothing else, and the server learns nothing at all [24, 54]. Intuitively, an OPRF needs to be secure against a malicious client who is trying to learn more information about the server's key $k$, and a malicious server who is trying to learn more information about the client's input $x$. Earlier works [24, 41] defined

an OPRF as the secure computation of the above two-party functionality, and Jarecki et al. [36, 37] later gave strong but flexible security definitions for an OPRF in the UC framework [13].

An OPRF is said to be *verifiable* if the server commits to its key $k$ by publishing some public parameters derived from $k$. At the end of the OPRF protocol, the client should be convinced that the obtained value $y \in \mathcal{Y}$ satisfies $y = F(k, x)$ with respect to the server's committed key $k$. One benefit of verifiability is that it allows a group of clients to verify that the values they each obtain are all consistent with the same PRF key. Without verifiability, in applications where a client later reveals the obtained value to the server, a malicious server can link values with previous evaluations by using a different key for each evaluation.

Oblivious PRFs have many real-world applications. They are used in private-set-intersection protocols [41, 46, 47, 58–60], in password-management systems [23, 37], in adaptive oblivious transfer [41], in de-duplication systems [44], in password-authenticated key exchange [40], and are deployed at Cloudflare to defend against Denial of Service attacks [21]. As a result, there is an ongoing effort to standardize OPRFs at the Crypto Forum Research Group [20].

An OPRF can be built from general secure two-party computation. A much simpler and widely used OPRF, called DH-OPRF, is built from a PRF whose security is based on the Decisional Diffie-Hellman (DDH) assumption in the random-oracle model. Let $\mathbb{G}$ be a cyclic group of prime order $q$, and let $H : \mathcal{X} \to \mathbb{G}$ be a hash function. For $k \in \mathbb{Z}_q$ and $x \in \mathcal{X}$, the PRF is defined as $F(k, x) = H(x)^k$. This PRF is secure, assuming DDH holds in $\mathbb{G}$ and $H$ is a random oracle [53]. This PRF then supports the following OPRF protocol: a client computes $H(x)$, blinds it as $u \leftarrow H(x)^r$ for a random $r \xleftarrow{R} \mathbb{Z}_q$, and sends $u$ to the server. The server responds with $v \leftarrow u^k$. The client then computes the unblinded PRF value $y \leftarrow v^{1/r} = H(x)^k$. Appropriate modifications can make this OPRF verifiable. Security of the resulting OPRF relies on the one-more discrete-log assumption [7]. Jarecki et al. [36, 37] showed this OPRF is secure in the Universally Composable framework [13].

Another simple verifiable OPRF in the random-oracle model, called RSA-OPRF, is derived directly from RSA blind signatures [7, 17]. Since there are quantum-polynomial-time algorithms for the DDH and RSA problems, neither of these OPRFs is post-quantum secure.

Building an efficient post-quantum secure OPRF is more challenging. One solution is to use a generic post-quantum secure two-party-computation protocol to evaluate a PRF. For example, instantiating Yao's garbled-circuits protocol with a post-quantum-secure oblivious transfer results in a post-quantum-secure two-party computation protocol [11] that can then be used to obliviously evaluate an AES circuit. The downside is that the communication in generic protocols is proportional to the circuit size, which motivates the search for efficient special-purpose OPRF protocols from post-quantum primitives. Albrecht et al. [3] recently proposed an OPRF based on the ring learning-with-errors problem and the short-integer-solution problem in one dimension.

**Our contributions.** In this paper we give another path towards a simple post-quantum secure OPRF by constructing several OPRFs from hard problems on isogenies of elliptic curves, in the random-oracle model.

Our first set of constructions operates on supersingular elliptic-curve isogenies over a field $\mathbb{F}_{p^2}$. Starting with a simple idea for an OPRF in the honest-but-curious setting, based on the SIDH key-exchange protocol of De Feo, Jao, and Plût [22], we then show how to elevate this OPRF to the setting of a malicious client and malicious server, and to make the OPRF verifiable. Our security proofs are set in the UC framework [13] in the random-oracle model. We describe our construction using an abstraction we call an *augmentable commitment*, defined in Section 2. These commitments abstract away many of the complexities of working with supersingular-curves isogenies, and they may be of independent interest.

To ensure that our OPRF is secure against a malicious client, we construct a zero-knowledge proof of knowledge for proving that the first message the client sends to the server is well formed. Here, a well formed message should contain an elliptic curve, obtained by correctly applying an isogeny to some base curve, together with points on that curve, obtained by applying that same isogeny to predetermined points on the base curve. To secure against a malicious server and obtain a verifiable OPRF, we construct an additional zero-knowledge proof of knowledge for proving that four elliptic curves $(E, E_a, E_b, E_{ab})$ form an isogeny DDH tuple, where the prover only knows the isogenies $\phi_a\colon E \to E_a$ and $\phi'_a\colon E_b \to E_{ab}$, whereas the isogeny $\phi_a\colon E \to E_a$ is private to the client. Our complete protocol requires up to 2MB of communication for 128-bit security, with the main bottleneck being the cut-and-choose repetitions in our zero-knowledge proofs of knowledge. We describe this protocol, using the language of augmentable commitments, in Section 6.

Our second class of OPRF protocols, presented in Section 8, builds an OPRF from a commutative group action, such as the one obtained from isogenies of ordinary elliptic curves [19, 61] or from isogenies of supersingular curves over $\mathbb{F}_p$ as in CSIDH [14]. Commutative group actions give rise to a generalized Diffie-Hellman problem, yet a construction similar to the DH-OPRF is not currently possible. The reason is that there is no known way to construct a hash function that maps its inputs to uniformly sampled elements in an isogeny class, without learning additional information about the output elements. This additional information would allow the client to evaluate the PRF at any point of its choice from just a single response from the server, breaking the security requirement. Therefore, an OPRF from commutative group actions requires a very different approach.

Our construction makes use of two observations. First, we adapt the Naor-Reingold PRF [54] to the setting of a commutative group action. This requires a new proof of security because the original proof of security in [54] relies on the DDH assumption and its random self-reduction. The difficulty is that the DDH problem for a commutative group action does not have the required random self-reduction. We nevertheless prove PRF security based on the DDH assumption

for such group actions; however the security reduction is not as efficient as for DDH over groups. Second, we observe that, similarly to the original PRF construction [54], this group-action variant admits an oblivious evaluation. The resulting OPRF scheme makes use of a 1-out-of-2 oblivious-transfer protocol, but such protocols are already known from isogeny problems [6, 51, 63, 69]. We thus obtain an OPRF from a commutative group action.

Between the two constructions, the supersingular construction is asymptotically more efficient, in the sense that it requires asymptotically less communication between the client and the server. The reason is a sub-exponential quantum algorithm for the discrete-log problem for a commutative group action due to Kuperberg [48, 49]. Kuperberg's attack applies to commutative group actions, which underpin our second construction, yet it does not apply to the non-commutative structure of supersingular isogenies over $\mathbb{F}_{p^2}$, which underpin our first construction. As a result, the first construction allows using smaller fields, which results in less communication asymptotically (in the security parameter). Its exponential security also makes it more robust to improvements in attacks. However, the second construction has better (i.e., smaller) constants, and as a result, the second construction is more efficient concretely: 424KB of communication vs. 2MB for the first construction.

### 1.1 Background and notation

Let $E$ be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Recall that every separable degree-$d$ isogeny $\phi\colon E \to E'$ has a kernel $G = \ker(\phi)$ which is a subgroup of order $d$ of $E(\overline{\mathbb{F}}_p)$. In the special case when $G$ is a cyclic subgroup of $E(\mathbb{F}_{p^2})$, we can succinctly represent $G$ by specifying a generator $K \in E(\mathbb{F}_{p^2})$, where $K$ is an element of the $d$-torsion of $E(\mathbb{F}_{p^2})$.

We follow de Saint Guilhem, Orsini, Petit, and Smart [63] and use the following notation to represent degree-$d$ isogenies. Recall that the *projective line* $\mathbb{P}_d$ is the set of all equivalence classes $[x\colon y]$, where $x, y \in \mathbb{Z}/d\mathbb{Z}$, and the ideal generated by $x$ and $y$ is all of $\mathbb{Z}/d\mathbb{Z}$. We specify an isogeny of degree $d$ using an element $k \in \mathbb{P}_d$. For $k = [k_p\colon k_q] \in \mathbb{P}_d$, and generators $P_d, Q_d$ of the $d$-torsion $E[d]$, the notation $\langle k \cdot (P_d, Q_d)\rangle$ refers to the order-$d$ cyclic group generated by $k_p P_d + k_q Q_d \in E[d]$.

### 1.2 Overview of our techniques

Our main result is an OPRF from isogenies on supersingular elliptic curves. We briefly summarize the main technical ideas, and refer to Sections 2–7 for the details.

Let $E/\mathbb{F}_{p^2}$ be a fixed supersingular elliptic curve, and let $N_{\mathsf{K}}, N_{\mathsf{M}}, N_{\mathsf{R}}$ be positive integers such that $E[N_{\mathsf{K}} \times N_{\mathsf{M}} \times N_{\mathsf{R}}]$ is contained in $E(\mathbb{F}_{p^2})$, where $p, N_{\mathsf{K}}, N_{\mathsf{M}}, N_{\mathsf{R}}$ are pairwise relatively prime. Let us derive a PRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ from the SIDH key-exchange protocol of [22]. The PRF makes use of two hash functions $H_1 : \mathcal{X} \to \mathbb{P}_{N_{\mathsf{M}}}$ and $H_2 : \mathcal{X} \times \mathbb{F}_{p^2} \to \mathcal{Y}$, and works as follows:

- The domain is $\mathcal{X}$. For each $x \in \mathcal{X}$ we obtain $m = H_1(x) \in \mathbb{P}_{N_{\mathsf{M}}}$, for which there is a corresponding degree-$N_{\mathsf{M}}$ isogeny $\phi_m : E \to E_m$;

– The key space is $\mathcal{K} = \mathbb{P}_{N_\mathsf{K}}$. For each $k \in \mathbb{P}_{N_\mathsf{K}}$ there is a corresponding degree-$N_\mathsf{K}$ isogeny $\phi_k : E \to E_k$;
– Let $\phi : E \to E_{m,k}$ be an isogeny with kernel $\ker(\phi_m) \times \ker(\phi_k)$.
Define $F(k, x) = H_2(x, j(E_{m,k}))$.

When $H_1$ and $H_2$ are modeled as random oracles, and assuming $N_\mathsf{K}$ is sufficiently large (i.e., superpolynomial in the security parameter), this function $F$ is a secure PRF.

To make this PRF into an oblivious PRF between a client and a server, it is tempting to try the following blinding approach (also used in [62, 65] in an attempt to construct a blinded version of an earlier undeniable-signature scheme [35]):

– The client has $x \in \mathcal{X}$. It computes $m = H_1(x) \in \mathbb{P}_{N_\mathsf{M}}$ which defines the degree-$N_\mathsf{M}$ isogeny $\phi_m : E \to E_m$ above. The client chooses a random $r \in \mathbb{P}_{N_\mathsf{R}}$, and computes the corresponding degree-$N_\mathsf{R}$ isogeny $\phi_r : E \to E_r$. Next, the client constructs an isogeny $\phi_{r,m} : E \to E_{r,m}$ whose kernel is $\ker(\phi_r) \times \ker(\phi_m)$. It sends $E_{r,m}$ to the server, along with four additional points on $E_{r,m}$, as specified in Section 3. Two of these four points are computed as $P'_\mathsf{K} = \phi_{r,m}(P_\mathsf{K})$ and $Q'_\mathsf{K} = \phi_{r,m}(Q_\mathsf{K})$, where $P_\mathsf{K}, Q_\mathsf{K} \in E$ are some fixed generators of $E[N_\mathsf{K}]$.
– The server has the secret key $k \in \mathbb{P}_{N_\mathsf{K}}$, and the corresponding isogeny $\phi_k : E \to \mathbb{E}_k$. It uses $P'_\mathsf{K}, Q'_\mathsf{K}$ to construct the curve $E_{r,m,k}$, which is the target of an isogeny acting on $E$ and whose kernel is $\ker(\phi_r) \times \ker(\phi_m) \times \ker(\phi_k)$. It sends $E_{r,m,k}$ back to the client, along with two additional points in $E[N_\mathsf{R}]$.
– The client uses its knowledge of $\phi_r$ to recover the required $E_{m,k}$ using an appropriate dual isogeny $\hat{\phi}' : E_{r,m,k} \to E_{m,k}$. Once the client has $E_{m,k}$, it can obtain the required PRF value $F(k, x)$ since $F(k, x) = H_2(x, j(E_{m,k}))$.

While this is a natural construction for an OPRF, it is unfortunately completely insecure. It is vulnerable to a clever active attack due to Galbraith et al. [27], which was originally used to attack SIDH key exchange where one of the parties uses a static key. In our setting, the attack lets a malicious client send carefully crafted points $P'_\mathsf{K}, Q'_\mathsf{K} \in E_{r,m}$ that are *not* the images of the fixed points $P_\mathsf{K}, Q_\mathsf{K} \in E$ under the isogeny $\phi_{r,m} : E \to E_{r,m}$. The client can then learn information about the PRF key $k$ from the server's response. With enough such queries, the client can extract $k$ from the server, thus fully breaking the OPRF.

In the SIDH key-exchange setting, there are several countermeasures against this active attack. Kirkwood et al. [45] suggest an approach, based on the Fujisaki-Okamoto [25] transformation, where the client sends encrypted information to the server. The server decrypts and uses the information from the client to validate the request. However, this approach cannot be used in an OPRF protocol because the information sent from the client reveals $m$ to the server, which violates the OPRF privacy requirement.

Our solution is to have the client prove to the server that the points $P'_\mathsf{K}$ and $Q'_\mathsf{K}$ are generated correctly without leaking any information about $m$ or $r$ to the server. To do so, we present in Section 5 a special-purpose zero-knowledge protocol that allows the client to prove the correctness of the points it sends. Our

protocol develops an idea sketched by Galbraith [26, Section 7.2], and builds on the isogeny-based identification protocol of De Feo et al. [22].

We obtain an OPRF that is secure against a malicious client. To further secure the OPRF against a malicious server, the server needs to somehow prove to the client that its response $E_{r,m,k}$ is consistent with its commitment $E_k$ to the secret key $k \in \mathbb{P}_{N_K}$. In other words, the server needs to prove that $(E, E_{r,m}, E_k, E_{r,m,k})$ form an isogeny DDH tuple, where the server only knows $\phi_k : E \to E_k$ and $\phi'_k : E_{r,m} \to E_{r,m,k}$. A similar protocol is needed in the constructions of [35,62,65] for the purpose of online signature confirmation. However, we cannot use their protocol because they assume the server knows both $\phi_k$ and $\phi_{r,m} : E \to E_{r,m}$. For us, this would break the OPRF privacy requirement because $\ker(\phi_{r,m})$ reveals information about $m \in \mathbb{P}_{N_M}$.

To address this, we develop in Section 6 a zero-knowledge proof of equality that lets the server prove the consistency of its response to the client. A key challenge is to ensure security of the OPRF, meaning that we must prevent the client from abusing the consistency check for extracting information about the key $k$. The result is a new private-coin protocol, that jointly meets the security requirements of both parties, and is quite different from the [22]-style public-coin protocol.

Our complete verifiable OPRF appears in Protocol 15.

**Security assumptions.** Our OPRF construction is based on the hardness of isogeny problems on supersingular curves over a field $\mathbb{F}_{p^2}$ for a prime $p$ of the form $p = f \cdot N_1 \cdot \ldots \cdot N_n - 1$, for relatively prime $N_i$. Specifically, for our verifiable OPRF, we use $n = 5$ prime powers.

The privacy of the client in our protocol relies on the hardness the Decisional SIDH Isogeny Problem [22, 29] adjusted from the standard SIDH setting of $n = 2$ prime powers to our setting of $n = 5$ (similarly to [35, 63, 65]). The security of the server in our protocol relies on a one-more Diffie-Hellman-type assumption in the SIDH setting. Recently, Merz, Minko, and Petit [52] presented a polynomial-time attack on certain "one-more" SIDH assumptions, introduced in [35, 65]. In Section 3, we present a new type of one-more SIDH assumption and discuss why it is not susceptible to this attack. Finally, our zero-knowledge proof, designed to prevent the active attack of [27], relies on the hardness of a variant of the Decisional Supersingular Product problem [22]. We discuss the security assumptions in more detail in Sections 3 and 5.

### 1.3 Additional related work

**OPRF from oblivious-transfer extension.** An efficient oblivious PRF can be constructed from oblivious-transfer extension [33]. The first works to do so [47,59,60] constructed a one-time OPRF, namely one where the client can only issue a single query to the server. Subsequent work [58] constructs a many-time OPRF from oblivious-transfer extension, but the client must choose all the query points before the OPRF key is generated. These non-adaptive OPRF schemes

are sufficient for protocols for private set intersection, and can be post-quantum secure if the underlying 1-out-of-2 oblivious transfer is post-quantum secure. The constructions in this paper give an OPRF which allows the client to select the query points adaptively, at any time after the OPRF key is generated, and supports an exponential size domain.

**Blind signatures.** Verifiable OPRFs share resemblance with blind signatures [17]. Both primitives allow a server holding a secret key to provide the client with a "certified" value on blinded input. However, unlike an OPRF, a blind signature does not have to be deterministic, yet it has to be publicly verifiable. Indeed, Jarecki and Liu [41] observed that earlier constructions [12] of oblivious-transfer protocols from unique blind signatures [7, 8, 17] and, similarly, from blind IBE schemes [31], give rise to OPRFs. None of these constructions are post-quantum secure. Recent works [62, 65] constructed variants of blind signatures from supersingular isogenies. As discussed above, the online verification protocols in these schemes require unblinding the message.

## 2 Augmentable commitments

In this section we introduce a primitive, called *augmentable commitments*, that makes it easier to describe the OPRF construction and prove its security. This abstraction makes it possible to describe the scheme without cluttering the description with many elliptic curve points.

An augmentable commitment is a commitment scheme where one can commit to a value $x_1 \in \mathcal{X}_1$ to obtain a commitment $\mathsf{com}$. Later, someone else can append $x_2 \in \mathcal{X}_2$ to the commitment $\mathsf{com}$ to obtain a new commitment $\mathsf{com}'$ to $(x_1, x_2)$. One can also obtain $\mathsf{com}'$ by committing in the reverse order, by first committing to $x_2 \in \mathcal{X}_2$, and then appending $x_1 \in \mathcal{X}_1$. We will refer to $\mathsf{com}'$ as $[\![x_1, x_2]\!]$. Regular values are append-only, in the sense that, given $[\![x_1, x_2]\!]$, it should be computationally unfeasible to compute $[\![x_2]\!]$ or $[\![x_1', x_2]\!]$. Looking ahead, this "non-malleability" property will provide privacy *for the server* in our OPRF protocol. It prevents the client from learning the value of the OPRF at one point given its evaluation at another.

To hide the contents of the commitment, its creator may include in it a special type of value $r \in \mathcal{R}$, called a *blind*. Such a blinded commitment $[\![r, x_1, x_2]\!]$ can later be *unblinded* to obtain $[\![x_1, x_2]\!]$, which is a binding commitment to $x_1$ and $x_2$, but may not be hiding. The blinding property will provide privacy *for the client* in our OPRF protocol, as it will prevent the server from learning the point where the OPRF is being evaluated.

We next define augmentable commitments more precisely and more generally. In the next sections we show how to use augmentable commitments to construct an OPRF scheme and how to construct them from supersingular isogenies.

**Definition 1 (Augmentable Commitment Scheme).** An augmentable commitment scheme $\mathcal{G}$ with an *input space* $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_{n-1}$, a blinding space $\mathcal{R} \coloneqq \mathcal{X}_n$, a commitment space $\mathcal{C}$, and a space of representatives $\mathcal{J}$, consists of five algorithms

- $\mathsf{Setup}(1^\lambda) \to \mathsf{com}_0 \in \mathcal{C}$. The algorithm takes as input the security parameter and outputs the "empty" commitment $\mathsf{com}_0$.
- $\mathsf{Blind}(\mathsf{com}_0 \in \mathcal{C}, r \in \mathcal{R}) \to \mathsf{com} \in \mathcal{C}$. The algorithm takes as input the empty commitment and a blind value $r$, and creates an initial blinded commitment.
- $\mathsf{Append}\,(\mathsf{com} \in \mathcal{C}, i \in [n-1], x \in \mathcal{X}_i) \to \mathsf{com}' \in \mathcal{C}$. The algorithm takes as input a commitment $\mathsf{com}$, an index of an input space, and an input from that space, and outputs a new commitment. The input commitment $\mathsf{com}$ can be the empty commitment $\mathsf{com}_0$, a blinded commitment output by $\mathsf{Blind}$, or a commitment obtained from a previous call to $\mathsf{Append}$.
- $\mathsf{Unblind}\,(\mathsf{com} \in \mathcal{C}, r \in \mathcal{R}) \to \mathsf{com}' \in \mathcal{C}$. The algorithm takes as input a commitment previously blinded with $r$ together with the same blind value $r$ used for blinding, and outputs an unblinded commitment.
- $\mathsf{Invariant}\,(\mathsf{com} \in \mathcal{C}) \to j \in \mathcal{J}$ returns the invariant of a commitment.

For simplicity, we avoid including explicit public parameters in the syntax of the scheme. If the scheme requires the $\mathsf{Setup}$ algorithm to set some public parameters, we assume without the loss of generality that they are included in the empty commitment $\mathsf{com}_0$ and in all subsequent commitments.

Note that the $\mathsf{Blind}$ step is the only time when an element $r \in \mathcal{R}$ of the blinding space may be committed to.

For brevity, we use the notation $[\![x_1, \ldots, x_t]\!]$ to refer to a commitment to a sequence of elements $x_1 \in \mathcal{X}_{i_1}, \ldots, x_t \in \mathcal{X}_{i_t}$. Specifically, if none of the distinct indices $i_1, \ldots, i_t \in [n-1]$ is the blinding index, we define $\mathsf{com}_j \leftarrow \mathsf{Append}(\mathsf{com}_{j-1}, i_j, x_j)$, and set $[\![x_1, \ldots, x_t]\!] \coloneqq \mathsf{com}_t$. Similarly, if $i_1 = n$ is the index of the blinding space $\mathcal{R} = \mathcal{X}_n$, we define $\mathsf{com}_1 \leftarrow \mathsf{Blind}(\mathsf{com}_0, x_1)$, and for $j \in [2, t]$ we define $\mathsf{com}_j \leftarrow \mathsf{Append}(\mathsf{com}_{j-1}, x_j)$, and set $[\![x_1, \ldots, x_t]\!] \coloneqq \mathsf{com}_t$.

For two commitments $c, c' \in \mathcal{C}$, we write $c \sim c'$ if and only if $\mathsf{Invariant}(c) = \mathsf{Invariant}(c')$.

The commitment scheme must satisfy the following correctness property, which states that (i) commitments to the same set of elements in a different order are equivalent; and (ii) unblinding results in an a commitment to the remaining elements.

**Correctness.** For every $t \in [n-1]$, every set of *distinct* indices $i_1, \ldots, i_t \in [n-1]$, every set of values $x_j \in \mathcal{X}_{i_j}$, and every $r \in \mathcal{R}$, we require the following.

1. $\mathsf{Invariant}([\![x_1, \ldots, x_t]\!])$ is independent of the ordering of $x_1, \ldots, x_t$. Similarly, $\mathsf{Invariant}([\![r, x_1, \ldots, x_t]\!])$ is independent of the ordering of $x_1, \ldots, x_t$.
2. $\mathsf{Unblind}([\![r, x_1, \ldots, x_t]\!], r) \sim [\![x_1, \ldots, x_t]\!]$.

An augmentable commitment must satisfy the following three security requirements: hiding, weak binding, and one-more unpredictability. We give formal game-based definitions of those properties in the full version of this work.

**Hiding.** The hiding property requires that a random committed element, be it an input or a blind, computationally hides all other committed elements. More specifically, an adversary should not be able to distinguish between a commitment to a set of random values and a commitment to a set of values of his choice,

provided that the commitment includes at least one additional random element, that the adversary does not know. This additional element can either be an input element or a blind, i.e., the hiding property holds with respect to both inputs and blinds, with the only difference being that blinds can also be unblinded.

**Weak binding.** The binding requirement asks that no efficient adversary can produce a collision between two commitments. We actually only need a weak form of binding, in the sense that the adversary needs to produce a pair of distinct elements that create a collision with noticeable probability over a random choice of a sequence of appended elements.

**One-more unpredictability.** In an augmentable commitment scheme, the result of augmenting a secret value to one randomly chosen value should not reveal the result of augmenting that same secret value to other random values. Specifically, consider a game between a challenger and adversary. The challenger chooses a secret input value $k$ and gives the adversary $t+1$ challenges $m_1, \ldots, m_{t+1}$, each of which is a random input value to the commitment. The solution to the $i$th challenge is the $\mathsf{Invariant}(\llbracket m_i, k \rrbracket)$ of a commitment to both the challenge value and the challenger's secret value. Finally, the adversary may issue queries to the challenger. Each query consists of an input value $m$ of the adversary's choice, to which the challenger responds with $\mathsf{Invariant}(\llbracket m, k \rrbracket)$, where $k$ is the challenger's secret value. The one-more unpredictability property requires that after issuing at most $t$ queries the adversary should not be able to produce the solution to all $t+1$ challenges.

*Remark 2.* de Saint Guilhem et al. [63] introduced an abstraction called *semi-commutative masking structure* that captures both commutative group actions and isogenies on supersingular elliptic curves. Our abstraction of augmentable commitments draws inspiration from theirs and shares some technical similarities with it. One difference is that our abstraction separates regular values, that are append-only, from blinds, that can be removed.

## 3 Augmentable commitments from supersingular isogenies

In this section we show how to construct an augmentable commitment scheme from supersingular isogenies. We refer to this scheme as $\mathcal{G}_{\mathsf{si}}$. We begin by defining a parameterization algorithm, which we use throughout our construction and our security assumptions.

**Definition 3 (Parameterization $p(\lambda, n)$).** We define the following deterministic algorithm. On input a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathbb{N}$, compute the first $n$ primes $\ell_1, \ldots, \ell_n$ and choose $e_1, \ldots, e_n$ to be positive integers such that for all $i \in [n]$, $N_i \coloneqq \ell_i^{e_i} \approx 2^{2\lambda}$. Choose $f \in \mathbb{N}$ to be a cofactor such that $p = f \cdot N_1 \cdot \ldots \cdot N_n - 1$ is a prime. Output $p(\lambda, n) \coloneqq p$.

For $\lambda \in \mathbb{N}$, and $p(\lambda, n+1) = f \cdot N_1 \cdot \ldots \cdot N_{n+1} - 1$, the input space of the commitment are the projective lines $\mathbb{P}_{N_i}$ for $i \in [n-1]$, and the blinding space is the projective line $\mathbb{P}_{N_n}$. For now, we do not explicitly use the $N_{n+1}$ torsion, and in particular, $\mathbb{P}_{N_{n+1}}$ is not part of the commitment input/blinding spaces. In Section 5, we will use this extra torsion to construct zero knowledge proofs on our commitment scheme.

**Setup.** The input to the setup routine is a security parameter $\lambda \in \mathbb{N}$. It computes $p = p(\lambda, n+1) = f \cdot N_1 \cdot \ldots N_{n+1} - 1$, then chooses $E_0$ to be a random supersingular elliptic curve over $\mathbb{F}_{p^2}$ such that $E_0(\mathbb{F}_{p^2}) \cong \mathbb{Z}_{N_1}^2 \times \ldots \times \mathbb{Z}_{N_{n+1}}^2 \times \mathbb{Z}_f^2$. Finally, for $i \in [n]$, the setup routine chooses $P_i^0, Q_i^0$ generators of $E_0[N_i] \cong \mathbb{Z}_{N_i}^2$ and outputs the empty commitment that consists of the curve $E_0$ and the generators $(P_i^0, Q_i^0)_{i \in [n-1]}$.

Our augmentable commitments take the form $(E, (P_i, Q_i)_{i \in I})$, where $I \subseteq [n]$, representing the curve $E$ by its $j$-invariant $j(E) \in \mathbb{F}_{p^2}$ using $2 \log p$ bits. (All logarithms in this work have base two.) This defines the curve up to isomorphism, and a canonical curve in that isomorphism class can be efficiently computed. Therefore, before outputting a commitment, each of the algorithms in our construction first computes an isomorphism from the curve it has computed to the canonical curve of the same isomorphism class. It also computes the images of the points in the commitment under this isomorphism [5, 28, 63]. Thus, any published points are always on the canonical curve. Similarly to SIDH public-key compression [5, 18, 34], each basis can be represented using $3 \log N_i$ bits. Overall, the size of the commitment is at most $5 \log p$ bits.

**Blinding.** The Blind algorithm blinds the empty commitment with a blind $r \in \mathbb{P}_{N_n}$ as follows. First, compute a degree $N_n$ isogeny $\phi_r \colon E_0 \to E_r$ where $E_r = E_0 / \langle r \cdot (P_n^0, Q_n^0) \rangle$ and $P_n^0, Q_n^0$ is a canonical basis for $E_0[N_n]$. Then compute a canonical basis $P_n, Q_n$ for $E_r[N_n]$. This basis, together with the knowledge of the kernel of the dual isogeny $\hat{\phi}_r$ is what enables to later unblind the commitment. Finally output the commitment

$$[\![r]\!] := \Big( E_r,\ (\phi_r(P_j^0),\ \phi_r(Q_j^0))_{j \in [n-1]},\ P_n,\ Q_n \Big).$$

**Appending.** To append a value $x_t \in \mathbb{P}_{N_j}$ to a commitment $[\![r, x_1, \ldots, x_{t-1}]\!] = (E, (P_i, Q_i)_{i \in I})$ for some $j \in I \cap [n-1]$, the algorithm Append computes the isogeny $\phi' \colon E \to E'$ with kernel $\langle x_t \cdot (P_j, Q_j) \rangle$. The new commitment is then

$$[\![r, x_1, \ldots, x_t]\!] = \Big( E',\ (\phi'(P_i),\ \phi'(Q_i))_{i \in I \setminus \{j\}} \Big).$$

As values are added to the commitment, the Append algorithm drops the bases of the corresponding torsion groups from the commitment. However, the commitment tracks the basis for the blinding space throughout, and the Unblind algorithm uses them to remove the blind $r$.

**Unblinding.** Algorithm Unblind removes $r \in \mathbb{P}_{N_n}$ from a blinded commitment $[\![r, x_1, \ldots, x_t]\!] = (E', (P_i', Q_i')_{i \in I})$ by first computing the isogeny $\phi_r \colon E_0 \to E_r$ for

$E_r = E_0/\langle r \cdot (P_n^0, Q_n^0)\rangle$ together with the canonical basis $P_n, Q_n \in E_r[N_n]$ as in the $\mathsf{Blind}$ algorithm above. It then computes a representative $\hat{r} \in \mathbb{P}_{N_n}$ of the kernel $\langle \hat{r} \cdot (P_n, Q_n)\rangle$ for the dual isogeny $\hat{\phi}_r \colon E_r \to E_0$. Finally, it computes the unblinding isogeny $\phi \colon E' \to E$ where $E = E'/\langle \hat{r} \cdot (P'_n, Q'_n)\rangle$, and outputs $(E)$—a curve isomorphic to the curve of $[\![x_1, \ldots, x_t]\!]$.

The $\mathsf{Invariant}$ of a commitment $(E, (P_i, Q_i)_{i\in I})$ is the $j$-invariant $j(E) \in \mathbb{F}_{p^2}$.

The full specification of our augmentable-commitment construction $\mathcal{G}_{\mathsf{si}}$ appears in the full version of this work. We also prove there that $\mathcal{G}_{\mathsf{si}}$ meets the correctness requirement of Definition 1. We now turn to discussing its security.

**Hiding.** The hiding property of our construction relies on the following variant of the Decisional Supersingular Isogeny problem.

**Problem 4 (Decisional SIDH Isogeny problem).** Let $p = p(\lambda, n) = f \cdot N_1 \cdot N_2 \cdot \ldots \cdot N_n - 1$ be as in Definition 3 and $i \in [n]$. The Decisional SIDH Isogeny problem is to distinguish between the following two distributions:

1. $(E, E_\phi, P, Q, \phi(P), \phi(Q))$ where $E$ is a randomly chosen supersingular curve over $\mathbb{F}_{p^2}$, the points $P, Q \in E[(p+1)/N_i]$ are a random basis for the $(p+1)/N_i$-torsion of $E(\mathbb{F}_{p^2})$, $\phi$ is a random degree-$N_i$ isogeny from $E$ and $E_\phi$ is the codomain of $\phi$.
2. $(E, E', P, Q, P', Q')$ where $E$, $P$, and $Q$ are as above, $E'$ is another randomly chosen supersingular curve over $\mathbb{F}_{p^2}$, and the points $P, Q \in E[(p+1)/N_i]$ are a basis for the $(p+1)/N_i$-torsion of $E(\mathbb{F}_{p^2})$ chosen uniformly at random subject to the constraint that $e(P, Q)^{N_i} = e(P', Q')$, where $e(\cdot, \cdot)$ denotes the Weil pairing.

The **Decisional SIDH Isogeny assumption** is that for every constant $n$ and every $i \in [n]$, no efficient algorithm can distinguish between the above two distributions with probability non-negligible in $\lambda$.

The DSSI problem was originally introduced by De Feo et al. [22]. In its original form, it is the problem of deciding whether two supersingular curves over $\mathbb{F}_{p^2}$, for $p = \ell_1^{e_1} \cdot \ell_2^{e_2} \cdot f \pm 1$ are $\ell_1^{e_1}$-isogenous to one another. Galbraith and Vercauteren [29, Definition 3] introduced the above variant, in which the distinguisher is also given extra points on each curve. This problem is also discussed in [68, Problem 3.4] and [69]. Our construction requires using more than 2 large torsions, and in particular we assume the problem to be hard for $n = 5$. A three-prime variant is considered in [35], a four-prime variant in [65], and an $n$-prime variant appears in [4, 63].

*Remark 5.* Petit [57] showed an attack on "unbalanced" SIDH variants that reveal the action of a secret degree-$A$ isogeny on the $B$-torsion of the base curve for $B \gg A$. Petit's attack, as well as its recent improvement by Kutas et al. [50], further require that $A \cdot B > p$. Even though our augmentable commitment has a similar imbalance (with $A = N_i$ and $B = \Pi_{j\neq i} N_j$), their second condition

$A \cdot B > p$ does not hold in our case. Therefore, these attacks do not currently apply to our construction.

*Remark 6.* The requirement that $e(P,Q)^{N_i} = e(P',Q')$ is needed to prevent a simple distinguishing attack based on the Weil pairing. Let $e_m \colon E[m] \times E[m] \to \mu_m$ be the Weil pairing on the $m$-torsion. Then it holds that [64, Proposition III.8.2]: $e_m(\phi(P), \phi(Q)) = e_m(P,Q)^{\deg(\phi)}$, where the first pairing is computed over $E'$. The requirement $e(P,Q)^{N_i} = e(P',Q')$ prevents distinguishing via this relation, by making sure it holds in both cases.

In the full version of this work we prove the augmentable commitment scheme $\mathcal{G}_{\mathsf{si}}$ is hiding under the Decisional SIDH Isogeny assumption.

**Weak binding.** The binding requirement builds on the conjectured difficulty of efficiently finding a pair of distinct isogenies of the same prime-power degree with the same target curve. The following problem underpins the security of Charles, Lauter, and Goren [16] hash function.

**Problem 7 (Supersingular Isogeny Collision problem).** Let $p = p(\lambda, n)$ be a prime as in Definition 3, and let $\ell$ be a different prime. Given a randomly chosen supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find a positive integer $k$, a supersingular curve $E'/\mathbb{F}_{p^2}$, and two distinct isogenies of degree $\ell^k$ from $E$ to $E'$.
The **Supersingular Isogeny Collision assumption** states that for every constant $n$, no efficient adversary solves the above problem with probability non-negligible in $\lambda$.

In the full version of this work we prove the our protocol meets the weak-binding requirement under the supersingular-isogeny collision assumption.

**One-more unpredictability.** Intuitively, we require that when a secret $K \xleftarrow{R} E[N_{\mathsf{K}}]$ is chosen at random, then the value $E/\langle M_1, K\rangle$, for a given randomly chosen $M_1 \xleftarrow{R} E[N_{\mathsf{M}}]$, should not reveal the value $E/\langle M_2, K\rangle$, for another randomly chosen $M_2 \xleftarrow{R} E[N_{\mathsf{M}}]$.

This kind of assumption appears in the group setting. For example, consider a cyclic group $\mathbb{G}$ of prime order $q$, and let $\alpha \xleftarrow{R} \mathbb{Z}_q$ be some secret. The One-More Diffie-Hellman problem [7] requires an adversary to compute the value $v^\alpha$ for $t+1$ randomly chosen values $v \xleftarrow{R} \mathcal{G}$ while allowing the adversary to make at most $t$ queries to a CDH oracle for $\alpha$ (i.e., an oracle that replies with $u^\alpha$ on a query $u \in \mathbb{G}$). The One-More Diffie-Hellman assumption states that no adversary can solve this problem for any polynomial $t$ with non-negligible probability.

Our starting point is a candidate of the One-More Diffie-Hellman assumption in the SIDH setting, introduced by Srinath and Chandrasekaran [65], called the *One-More SSCDH* assumption. Their candidate assumption stated that given $t$ queries to a SIDH oracle (i.e., an oracle that responds to a query $M \in E[N_{\mathsf{M}}]$ with $E'/\langle M, K\rangle$ for a secret $K \in E[N_{\mathsf{K}}]$), it is computationally infeasible to produce $t+1$ pairs of curves $(E/\langle M\rangle, E/\langle M,\ K\rangle)$ for $t+1$ distinct $M \in E[N_{\mathsf{M}}]$.

However, this starting point is insecure. First, Merz, Minko, and Petit [52], recently showed a polynomial-time attacks on this assumption. Moreover, this assumption is also vulnerable to the active key-recovery attack on SIDH with static keys [27]. Finally, our security proof requires giving the adversary access to a decision oracle, which opens up the possibility of computation-to-decision reductions for isogeny problems [26, 29, 67]. We now explain each of these attacks and describe how our proposed one-more problem avoids them.

*Recent attacks on one-more SIDH problems.* The attack of Merz, Minko, and Petit [52] exploits a key difference between the One-More DH assumption in the group setting and the OMSSCDH assumption [65]. In the group setting, the adversary needs to produce valid DH tuples for *random* challenges. In contrast, the assumption of Srinath and Chandrasekaran [65] relaxes this requirement and allows the challenges to be *adversarially chosen*. In the group setting, relaxing the random-challenges requirement breaks the one-more hardness: given a single DH tuple $(v, v^\alpha)$, it is easy to produce any number of random-looking DH tuples simply by choosing $\beta \xleftarrow{R} \mathbb{Z}_q$ and computing the DH tuple $(v^\beta, (v^\alpha)^\beta)$.

Even though the simple rerandomization that works in the group setting does not extend to the SIDH setting (due to the requirement that the challenges are all of the form $E/\langle M \rangle$ for $M \in E[N_{\mathsf{M}}]$), Merz et al. devise a polynomial-time attack on the above OMSSCDH assumption by computing short isogenies from a given SIDH tuple. They point out that their polynomial-time attack on OMSSCDH does not translate to a polynomial-time attack on the signature scheme of Srinath and Chandrasekaran [65] nor on the signature scheme of Jao and Soukharev [35] because the challenges in these schemes are outputs of a hash function, modeled as a random oracle. This is consistent with the group setting, where the one-more assumption is only hard for random challenges.

Therefore, to avoid this attack, we provide the adversary in our one-more problem with *random* challenges, rather than allowing it to choose the challenge curves adversarially.

*Active attacks.* The aforementioned modification prevents the specialized attack of [52]. However, the resulting problem is still vulnerable to a general active attack on SIDH with static keys due to Galbraith et al. [28]. As discussed in the introduction, by sending a sequence of queries, each of which consists of a curve $E'$ together with a maliciously crafted basis $P_{\mathsf{K}}, Q_{\mathsf{K}} \in E'[N_{\mathsf{K}}]$, an adversary can recover the secret key $K$. We therefore require the adversary to submit kernels $M$ as its solve queries, rather than arbitrary curves with (possibly malicious) torsion points. This requirement is enforced in the actual protocol using a zero-knowledge proof of knowledge, described in the Section 5.

*Search-to-decision reductions.* The security proof of our OPRF requires a stronger variant of a one-more assumption, in which the adversary is given additional access to a decision oracle that allows it to check the validity of solutions throughout its execution. In the group setting, the Gap One-More Diffie-Hellman assumption [36, 42] states that the one-more problem is hard even in the presence of such a decision oracle.

The exact same type of assumption is unsound in the SIDH setting. The issue, as shown by Galbraith and Vercauteren [29], and independently by Thormarker [67], is that the search variant of the isogeny problem can be reduced to its decisional variant. Moreover, as pointed out by Galbraith [26], a similar search-to-decision reduction applies also for the SIDH problem. (We describe this reduction for completeness in the full version of this work.) The One-More SIDH problem is thus easy if the adversary is given a full-fledged decision oracle for the SIDH problem. Therefore, we need to formulate a weaker assumption, in which the adversary is given oracle access to a more restrictive decision oracle. Intuitively, we only allow the adversary to check SIDH solutions to the challenges given to it (with respect to the secret key $K$), rather than make arbitrary SIDH decision queries. This is a much weaker assumption, and in particular, unlike a general SIDH decision oracle, the challenger answering this more restricted form of queries can be efficiently implemented.

**Attack Game 8 (Auxiliary One-More SIDH).** Let $p = p(\lambda, n) = f \cdot N_1 \cdot \ldots \cdot N_n - 1$ be as in Definition 3 and let $\mathsf{M}, \mathsf{K} \in [n]$ be distinct indices. Consider the following game, played between a challenger and an adversary:

- The challenger chooses a random supersingular curve $E_0/\mathbb{F}_{p^2}$ and a random basis $P, Q$ of $E_0[(p+1)/(N_\mathsf{M} \cdot N_\mathsf{K})]$. It then chooses a random point $K \in E_0(\mathbb{F}_{p^2})$ of order $N_\mathsf{K}$, computes the isogeny $\phi\colon E_0 \to E_0/\langle K \rangle$, and sends $E_0$, $P, Q$, and $E_0/\langle K \rangle$ to the adversary.
- The adversary makes a sequence of queries to the challenger, each of which can be one of the following two types:
  - Challenge query: the challenger chooses $M \xleftarrow{\text{R}} E_0[N_\mathsf{M}]$ and sends it to the adversary.
  - Solve query: the adversary submits $V \in E_0[(p+1)/N_\mathsf{K}]$ to the challenger, who computes the isogeny $\phi\colon E_0 \to E'$ with $\ker(\phi) = \langle V, K \rangle$, and sends $j(E') \in \mathbb{F}_{p^2}$, together with $\phi(P), \phi(Q)$ to the adversary.
  - Decision query: the adversary submits a pair $(i, j)$ to the challenger, where $i$ is a positive integer bounded by the number of challenge queries the adversary has made so far, and $j \in \mathbb{F}_{p^2}$. The challenger responds $\mathsf{true}$ if $j = j(E_0/\langle M, K \rangle)$, where $M$ is the challenger's response to the $i$th challenge query, and $\mathsf{false}$ otherwise.
- At the end, the adversary outputs a list of distinct pairs, each of the form $(i, j)$ where $i$ is a positive integer bounded by the number of challenge queries, and $j \in \mathbb{F}_{p^2}$.

We call an output-pair $(i, j)$ correct if $j$ is the $j$-invariant of the curve $E' = E/\langle M,\ K \rangle$ where $M$ is the challenger's response to the $i$th challenge query. We say that the adversary wins the game if the number of correct pairs exceeds the number of Solve queries.

The **Auxiliary One-More SIDH assumption** states that for every constant $n$ and every distinct $\mathsf{M}, \mathsf{K} \in [n]$, every efficient adversary wins the above game with probability negligible in $\lambda$.

*Remark 9.* We allow the adversary to learn the action of the secret isogeny on an auxiliary torsion group $E_0[(p+1)/(N_{\mathsf{M}} \cdot N_{\mathsf{K}})]$. (The construction of Srinath and Chandrasekaran [65, Sec. 4.4] implicitly has this type of leakage, yet their security proof seems to overlook this when reducing to their version of the OMSSCDH assumption.)

It is important that the solve query provides the adversary with the action of the secret isogeny only on this torsion. Disclosing the action of the secret isogeny on $E[N_{\mathsf{K}}]$ would leak the secret. Disclosing the action of the secret isogeny on $E[N_{\mathsf{M}}]$ would allow the adversary to break the one-more assumption, since the adversary would eventually learn the action of $\phi$ on $E[N_{\mathsf{M}}]$.

In the full version of this work, we show that $\mathcal{G}_{\mathsf{si}}$ is one-more unpredictable under the Auxiliary One-More SIDH assumption.

## 4 Oblivious PRF from augmentable commitments

We begin by giving an overview of our construction of an oblivious PRF from augmentable commitments. We do not yet give a formal security definition, so for now, we can think of an OPRF as a two party functionality $(x, k) \mapsto (F(k, x), \perp)$ where $F$ is a pseudorandom function. Intuitively, each execution should allow the user to evaluate the PRF at a single point, while providing privacy for the user's input.

Our basic protocol consists of two-rounds and is somewhat reminiscent of the DH-OPRF protocol in the group setting. Recall that in the group setting, the user, given input $x$, sends to the server the group element $\mathsf{com} \leftarrow H(x)^r$, which we can view as a commitment to $x$. The server then computes $\overline{\mathsf{com}} \leftarrow \mathsf{com}^k$ and sends it back to the user, who computes $\mathsf{com}_{\mathsf{out}} \leftarrow \overline{\mathsf{com}}^{1/r}$. Generalizing this protocol to the language of augmentable commitments, we obtain the protocol in Fig. 1.

**Handling malicious clients.** However, this basic construction has a critical problem. Our augmentable commitment scheme provides a weaker form of "one-more unpredictability", as compared to the One-More Diffie-Hellman assumption in the group setting. Specifically, the one-more-unpredictability adversary needs to submit values, rather than commitments, as its solve queries. In contrast, the group-based one-more DH assumption is stronger, in that it considers more powerful adversaries that can query the one-more challenger on group elements rather than on scalars. (The underlying reason for this security definition is to prevent the active attacks on our isogeny-based instantiation of augmentable commitments, as discussed in the introduction and in Section 3). Therefore, our construction requires the user to attach, as part of its message, a zero-knowledge proof of the committed values. We present this proof system in Section 5. This protocol is specific for the isogeny-based construction.

**Handling malicious servers.** In this simple OPRF, the user cannot detect malicious servers that use a different key on each response, or even send arbitrary responses that do not correspond to a well-defined key.

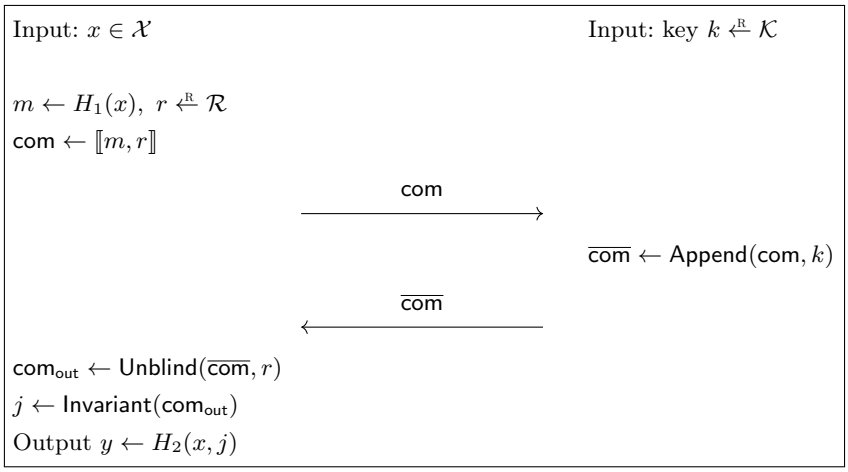



Fig. 1: The basic OPRF protocol from augmentable commitments. Note that, as presented, this basic version is not secure against malicious parties.

A verifiable OPRF provides the user with the following guarantee. On each evaluation of the OPRF, the user obtains, in addition to the output value $y = F(k, x)$, a function descriptor $\mathsf{pk}$. If on two inputs $x_1$ and $x_2$ the user obtains two outputs $y_1, \mathsf{pk}$ and $y_2, \mathsf{pk}$ with a matching function descriptor, there must exist a key $k$ such that $y_1 = F(k, x_1)$ and $y_2 = F(k, x_2)$. The function descriptor therefore commits the server to a particular function for all inputs.

In our verifiable-OPRF construction, the function descriptor is the output $y_\epsilon$ of the OPRF on some fixed point $\epsilon$. (We think of $\epsilon$ as being outside the "official" domain of the OPRF.) After obliviously evaluating the OPRF on a point $x$ and obtaining output $y_x$, the user runs $\lambda$ additional evaluations of the OPRF, each time setting the input at random as either $x$ or $\epsilon$. At the end of the $\lambda$ evaluations, the user checks that the output of each of the $\lambda$ evaluations matches either $y_\epsilon$ or $y_x$ (consistently with its random choice for that evaluation). If all $\lambda$ checks pass, the user accepts the output $y_x$ with respect to descriptor $y_\epsilon$.

An issue with the above protocol is that a malicious user may abuse the $\lambda$ evaluations to evaluate the OPRF on $\lambda$ additional points, rather than for verification. Learning the value of the OPRF on more than one point from a single instance of the protocol would violate the server's security requirement of the OPRF. To prevent this, we add an additional phase to our protocol: the server first commits to the outputs of the OPRF on the $\lambda$ verification instances. The user then proves to the server that each of the $\lambda$ verification inputs is either $x$ or $\epsilon$. (Doing this without revealing $x$ to the server requires an extra layer of blinding.) This provides the server with the assurance that the user would not learn any "extra" values of the OPRF from the verification instances. The server then opens the commitment to the verification outputs, which the client verifies as above. We present this protocol in Section 6.

In Section 7 we give the full specification (Protocol 15) of our final construction.

## 5 Zero-knowledge proof for point verification

A critical part of the OPRF construction is a zero-knowledge proof of knowledge (ZKPK) that lets the client prove to the server that its PRF query is well formed. Using the abstraction of augmentable commitments, what is needed is a ZKPK for the contents of an augmentable commitment, or more generally to the relation:

$$R_{\mathsf{com}} = \left\{ ((\mathsf{com}_0,\ \mathsf{com}_t),\ (x_1, \ldots, x_t)) : \begin{array}{c} \mathsf{com}_1 = \mathsf{Blind}(\mathsf{com}_0, x_1) \\ \mathsf{com}_i = \mathsf{Append}(\mathsf{com}_{i-1},\ x_i)\ \forall i \in [2,t] \end{array} \right\}.$$

The ZKPK we construct is specific to the instantiation of augmentable commitment from Section 3, and uses some of the algebraic properties of isogenies. Specifically, we design a custom ZKPK for the following relation $\mathcal{R}_{\mathsf{iso}}$. (In the full version of this work, we show how the relation $\mathcal{R}_{\mathsf{iso}}$ enables expressing statements about the language $R_{\mathsf{com}}$ for the augmentable commitment scheme $\mathcal{G}_{\mathsf{si}}$.)

Let $p = p(\lambda, n+1) = f \cdot N_1 \cdot \ldots \cdot N_{n+1} - 1$ be a prime as in Definition 3. For clarity, we denote $N_{\mathsf{S}} := N_{n+1}$. Let $E$ be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Define the relation:

$$\mathcal{R}_{\mathsf{iso}} := \left\{ \Big( j(E),\ P_{\mathsf{K}},\ Q_{\mathsf{K}},\ j(E'),\ P'_{\mathsf{K}},\ Q'_{\mathsf{K}},\ d \Big),\ V \right\}, \tag{1}$$

where the *statement* $\Big( j(E),\ P_{\mathsf{K}},\ Q_{\mathsf{K}},\ j(E'),\ P'_{\mathsf{K}},\ Q'_{\mathsf{K}},\ d \Big)$ contains:

- a $j$-invariant $j(E) \in \mathbb{F}_{p^2}$ of a supersingular elliptic curve $E/\mathbb{F}_{p^2}$,
- points $P_{\mathsf{K}}, Q_{\mathsf{K}} \in E[N_{\mathsf{K}}]$ for some $N_{\mathsf{K}}$ relatively prime to $N_{\mathsf{S}}$,
- a $j$-invariant $j(E') \in \mathbb{F}_{p^2}$ of a supersingular elliptic curve $E'/\mathbb{F}_{p^2}$,
- points $P'_{\mathsf{K}}, Q'_{\mathsf{K}} \in E'[N_{\mathsf{K}}]$, and
- a positive integer $d$ relatively prime to $N_{\mathsf{S}}$ and $N_{\mathsf{K}}$,

The *witness* $V$ is a point of order $d$ in $E(\mathbb{F}_{p^2})$ such that $E' = E/\langle V \rangle$ and the isogeny $\phi\colon E \to E'$ satisfies $P'_{\mathsf{K}} = \phi(P_{\mathsf{K}})$ and $Q'_{\mathsf{K}} = \phi(Q_{\mathsf{K}})$. Note that by definition, $N_{\mathsf{K}}$, $d$, and $N_{\mathsf{S}}$ all divide $(p+1)$ and are relatively prime.

**The protocol.** We design a ZKPK for the relation $\mathcal{R}_{\mathsf{iso}}$ where the verifier (server) has the statement $\Big( j(E),\ P_{\mathsf{K}},\ Q_{\mathsf{K}},\ j(E'),\ P'_{\mathsf{K}},\ Q'_{\mathsf{K}},\ d \Big)$ and the verifier (client) proves knowledge of the witness $V$. We first describe a protocol that has perfect completeness, constant soundness error, and honest-verifier computational zero knowledge. Repeating the protocol in parallel $\lambda$ times makes the soundness error negligible. Indeed, the repetitions required in this protocol (as well as in the one in the next section) are responsible for the bulk of the communication in our OPRF construction.

The protocol is based on the idea sketched by Galbraith [26, Sec 7.2], which builds on the isogeny-based identification protocol of De Feo et al. [22].

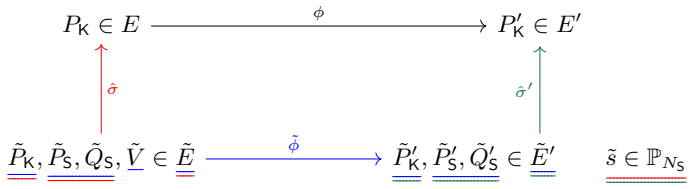
*Remark 10.* In the following, when we refer to the prover "committing" to one or more elements, we refer to a standard commitment scheme (as opposed to our augmentable commitment scheme) such as a standard hash-based commitment in the random-oracle model.

First, the prover chooses a random point $S$ of order $N_\mathsf{S}$. The prover then computes an isogeny $\sigma$ with domain $E$ and kernel $\langle S \rangle$ and an isogeny $\sigma'$ with domain $E'$ and kernel $\langle \phi(S) \rangle$. Let $\tilde{E}$ and $\tilde{E}'$ be the target curves of the isogenies $\sigma$ and $\sigma'$ respectively. For consistency of notation, we denote points on the curve $\tilde{E}$ as $\tilde{P}$, $\tilde{Q}$ etc. Similarly, we denote points on the curve $\tilde{E}'$ as $\tilde{P}'$, $\tilde{Q}'$ etc. The prover can also calculate the isogeny $\tilde{\phi}\colon \tilde{E} \to \tilde{E}'$ using the image of the generator $V$ of $\phi$ under $\sigma$.

The prover chooses a random basis $\tilde{P}_\mathsf{S}, \tilde{Q}_\mathsf{S}$ of the $N_\mathsf{S}$-torsion subgroup of $\tilde{E}$. The prover then computes the kernel of the dual isogeny $\hat{\sigma}$ and expresses its generator as $s \cdot (\tilde{P}_\mathsf{S}, \tilde{Q}_\mathsf{S})$ for some $s \in \mathbb{P}_{N_\mathsf{S}}$. (Note that the kernel of $\hat{\sigma'}$ is then generated by $s \cdot (\tilde{\phi}(\tilde{P}_\mathsf{S}), \tilde{\phi}(\tilde{Q}_\mathsf{S}))$.)

The prover commits separately to (1) the curve $\tilde{E}$ together with the points $\tilde{P}_\mathsf{S}, \tilde{Q}_\mathsf{S}$, (2) the curve $\tilde{E}'$ together with the points $\tilde{P}'_\mathsf{S} = \tilde{\phi}(\tilde{P}_\mathsf{S}), \tilde{Q}'_\mathsf{S} = \tilde{\phi}(\tilde{Q}_\mathsf{S})$, (3) the scalar $s$, (4) a random generator $\tilde{V}$ of $\ker(\tilde{\phi})$, and (5–8) the images of $P_\mathsf{K}, Q_\mathsf{K}$ under $\sigma$ and of $P'_\mathsf{K}, Q'_\mathsf{K}$ under $\sigma'$. (Committing to all those elements makes the protocol online-extractable without rewinding, which is necessary for UC security.)

Each execution of the protocol will verify the validity of only one of the two points $P'_\mathsf{K}$ and $Q'_\mathsf{K}$ according to a random choice made by the verifier. Additionally, according to another random three-way choice of the verifier, the prover will reveal one of three isogenies (i.e., either $\sigma$, $\sigma'$, or $\tilde{\phi}$) along with some points. The following diagram illustrates the commitments opened in each of the three cases where the verifier chooses to verify the validity of the point $P'_\mathsf{K}$:




- In the red case, the prover reveals the curve $\tilde{E}$, the random generators $\tilde{P}_\mathsf{S}$, $\tilde{Q}_\mathsf{S}$ of $\tilde{E}[N_\mathsf{S}]$, the element $\tilde{s} \in \mathbb{P}_{N_\mathsf{S}}$ , and the point $\tilde{P}_\mathsf{K} = \sigma(P_\mathsf{K}) \in \tilde{E}[N_\mathsf{K}]$. The verifier computes the isogeny $\hat{\sigma}\colon \tilde{E} \to \tilde{E}/\langle \tilde{s} \cdot (\tilde{P}_\mathsf{S}, s_q \tilde{Q}_\mathsf{S}) \rangle$, and checks that $\hat{\sigma}(\tilde{P}_\mathsf{K}) = [N_\mathsf{S}^2]P_\mathsf{K}$, where $[N_\mathsf{S}^2]$ is the multiplication by $N_\mathsf{S}^2$ map.
- Similarly, in the green case, the prover reveals the curve $\tilde{E}'$, the random generators $\tilde{P}'_\mathsf{S} = \tilde{\phi}(\tilde{P}_\mathsf{S})$, $\tilde{Q}'_\mathsf{S} = \tilde{\phi}(\tilde{Q}_\mathsf{S})$ of $\tilde{E}'[N_\mathsf{S}]$, the element $\tilde{s} \in \mathbb{P}_{N_\mathsf{S}}$, and the point $\tilde{P}'_\mathsf{K} = \sigma'(P'_\mathsf{K})$. The verifier computes the isogeny $\hat{\sigma'}\colon \tilde{E}' \to \tilde{E}'/\langle \tilde{s} \cdot (\tilde{P}'_\mathsf{S}, \tilde{Q}'_\mathsf{S}) \rangle$, and checks that $\hat{\sigma'}(\tilde{P}'_\mathsf{K}) = [N_\mathsf{S}]P'_\mathsf{K}$, where $[N_\mathsf{S}]$ is the multiplication by $N_\mathsf{S}$ map.
- Finally, in the blue case, the prover reveals the curves $\tilde{E}$ and $\tilde{E}'$, a random generator $\tilde{V}$ of $\ker(\tilde{\phi})$, and the points $\tilde{P}_\mathsf{S}, \tilde{Q}_\mathsf{S} \in \tilde{E}[N_\mathsf{S}]$, $\tilde{P}_\mathsf{K} \in \tilde{E}[N_\mathsf{K}]$, $\tilde{P}'_\mathsf{K} \in \tilde{E}'[N_\mathsf{K}]$, and $\tilde{P}'_\mathsf{S}, \tilde{Q}'_\mathsf{S} \in \tilde{E}'[N_\mathsf{S}]$. The verifier computes the isogeny $\tilde{\phi}\colon \tilde{E} \to \tilde{E}/\langle \tilde{V} \rangle$ and checks that $\tilde{\phi}(\tilde{P}_\mathsf{K}) = \tilde{P}'_\mathsf{K}$, $\tilde{\phi}(\tilde{P}_\mathsf{S}) = \tilde{P}'_\mathsf{S}$ and $\tilde{\phi}(\tilde{Q}_\mathsf{S}) = \tilde{Q}'_\mathsf{S}$.

*Remark 11.* In our protocol, as well as in the security game for the underlying assumption, we specifically choose to reveal the image of only a single generator of the $N_{\mathsf{K}}$-torsion under the secret random isogeny $\sigma$. The reason for this choice is to prevent a distinguishing attack using the Weil pairing. Had we revealed both images $\tilde{P}_{\mathsf{K}} = \sigma(P_{\mathsf{K}}), \tilde{Q}_{\mathsf{K}} = \sigma(Q_{\mathsf{K}})$, then the verifier would have obtained the two relations $e(\tilde{P}_{\mathsf{K}}, \tilde{V}) = e(P_{\mathsf{K}}, V)^{v \cdot \deg(\sigma)}$ and $e(\tilde{Q}_{\mathsf{K}}, \tilde{V}) = e(Q_{\mathsf{K}}, V)^{v \cdot \deg(\sigma)}$, which would allow to verifier to distinguish $V$ from random. By revealing only one out of the two points $\tilde{P}_{\mathsf{K}}, \tilde{Q}_{\mathsf{K}}$, and by revealing a random generator $v \cdot \sigma(V)$ instead of $\sigma(V)$, the protocol prevents tis pairing attack.

The zero-knowledge property of our protocol is based on the hardness of a variant of the Decisional Supersingular Product problem (DSSP), introduced by De Feo et al. [22]. As our protocol also needs to verify the action of the secret isogeny on the $N_{\mathsf{K}}$-torsion, we need to slightly strengthen the assumption by giving the adversary additional points. More specifically, we consider the following:

**Attack Game 12 (Auxiliary Decisional Supersingular Product).** Let $p = p(\lambda, n+1) = f \cdot N_1 \cdot \ldots \cdot N_{n+1}$ be as in Definition 3. Let $E_0$ be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ as above. Consider the following game, played between a challenger and an adversary:

- The adversary chooses and sends to the challenger $V_0 \in E(\mathbb{F}_{p^2})$ of order exactly $d$ relatively prime to $N_{\mathsf{S}}$, and a point $P_{\mathsf{K}} \in E(\mathbb{F}_{p^2})$ of order relatively prime to $N_{\mathsf{S}}$ and $d$.
- The challenger executes the following steps:
  - choose $c \xleftarrow{\text{R}} \{0,1\}$, $v \xleftarrow{\text{R}} \mathbb{Z}_d^*$, and a random point $V_1 \in E(\mathbb{F}_{p^2})$ of order $d$
  - compute a random degree-$N_{\mathsf{S}}$ isogeny $\sigma\colon E_0 \to E'$
  - send $j(E') \in \mathbb{F}_{p^2}$ and the points $v \cdot \sigma(V_c), \sigma(P_{\mathsf{K}}) \in E'(\mathbb{F}_{p^2})$ to the adversary
- The adversary outputs a bit $c'$.

We say that the adversary wins if $c' = c$.

The **Auxiliary Decisional Supersingular Product assumption** is that for every constant $n$, the winning probability of every efficient adversary in the above game is negligible.

In the full version of this work, we formally define sigma protocols, give the full details of the above protocol, and prove that it is special computational honest-verifier zero knowledge, under the Auxiliary Decisional Supersingular Product assumption. We also discuss how to transform this sigma protocol into a non-interactive zero-knowledge proof of knowledge (NIZKPK) in the random-oracle model using standard techniques.

**Concrete efficiency.** We estimate the size of the resulting NIZKPK. In a single execution of the above protocol, the prover sends 8 hash-based commitments in its first message. Of the three possible openings, the "blue" one, that consists

of 2 $j$-invariants and 7 points, is the largest one. The opening also includes 5 random nonces used for the hash-based commitments, each of which is $\lambda$-bits long. The size of a $j$-invariant in $\mathbb{F}_{p^2}$ is $2\log p$ bits. A naive representation of each point over $\mathbb{F}_{p^2}$ would have also been $2\log p$ bits ($x$-coordinate and a sign bit). However, Azarderakhsh et al. [5] observed that a point in an $N_i$-torsion can be represented using only $2\log N_i$ bits. Since in our construction $\log N_i \leq \log p/4$, the prover can send all 7 points in less than $4\log p$ bits, and together with the $j$-invariant, the size of the prover's last message is less than $6\log p$ bits. (In the non-interactive proof, the verifier's only message is a random challenge, which is derived from a random oracle and thus does not increase the size of the proof.) Since each execution of the protocol has soundness error 5/6, we must repeat the protocol $\lambda/\log(6/5) = 3.8\lambda$ times. Overall, we estimate the size of the proof as $3.8\lambda \cdot (13\lambda + 6\log p)$.

## 6 Zero-knowledge proof of equality of isogenies

Recall that to make our OPRF verifiable, the server must convince the verifier that it has evaluated the OPRF consistently with its evaluation on some fixed point. This boils down to proving the commitments satisfy the following relation

$$R_{\mathsf{eq}} = \left\{ ((\mathsf{com}_0, \mathsf{com}_1, \overline{\mathsf{com}}_0, \overline{\mathsf{com}}_1), k) \;\middle|\; \begin{array}{c} \mathsf{com}_0, \mathsf{com}_1, \overline{\mathsf{com}}_0, \overline{\mathsf{com}}_1 \in \mathcal{C} \\ k \in \mathcal{K} \\ \overline{\mathsf{com}}_0 = \mathsf{Append}(\mathsf{com}_0, k) \\ \overline{\mathsf{com}}_1 = \mathsf{Append}(\mathsf{com}_1, k) \end{array} \right\}$$

Moreover, the proof must be zero-knowledge, and in particular, the user should not learn any additional information about the key beyond what it already knows from $\overline{\mathsf{com}}_0$ and $\overline{\mathsf{com}}_1$.

The idea behind Protocol 13 below is as follows. The user (verifier) sends to the server $\lambda$ augmentable commitments, each of which is obtained by appending a random value $v_i$ to either $\mathsf{com}_1$ or $\mathsf{com}_2$, chosen at random. The user saves the values $v_i$ and the random choices $b_i \in \{0, 1\}$.

Next, the server (prover) appends its secret value $k$ to each of the $\lambda$ commitments, and sends to the user a hash-based commitment $h = H(j_1, \ldots, j_\lambda, s_{\mathsf{out}})$ to their invariants, where $s_{\mathsf{out}} \xleftarrow{\mathrm{R}} \{0,1\}^\lambda$.

The user then reveals to the server the random values $v_1, \ldots, v_\lambda$, and the server uses them to check that each of the $\lambda$ commitments received in the first round has indeed been obtained by appending $v_i$ to one of $\mathsf{com}_1$ or $\mathsf{com}_2$. This protects the server against a malicious user that tries to learn additional information about $k$ by sending commitments that are not $\mathsf{com}_1$ or $\mathsf{com}_2$.

Once this check passes, the server sends to the user the opening $s_{\mathsf{out}}$ to the hash-based commitment. Finally, the user computes the expected values of the invariants $j'_1, \ldots, j'_\lambda$ as $j'_i = \mathsf{Invariant}(\mathsf{Append}(\overline{\mathsf{com}}_{b_i}, v_i))$ and checks that $h = H(j'_1, \ldots, j'_\lambda, s_{\mathsf{out}})$.

This protocol is generic for augmentable commitments, but we think that its instantiation with the isogeny-based construction of augmentable commitments may be of independent interest.

**Protocol 13 (Equality of Appended Values).** Let $\mathcal{G}$ be an augmentable commitment scheme with input space $\mathcal{M} \times \mathcal{K} \times \mathcal{V} \times \mathcal{R}$, and commitment space $\mathcal{C}$. Let $\mathsf{NIZKPK}$ be a simulation-sound online-extractable proof for the relation $R_{\mathsf{com}}$. Let $H_3 \colon \{0,1\}^* \to \{0,1\}^\lambda$ be a hash function, modeled as random oracle.

**Inputs:**

- The verifier's inputs are: commitments $\mathsf{com}_0, \mathsf{com}_1, \overline{\mathsf{com}}_0, \overline{\mathsf{com}}_1 \in \mathcal{C}$.
- The prover's inputs are: commitments $\mathsf{com}_0, \mathsf{com}_1, \overline{\mathsf{com}}_0, \overline{\mathsf{com}}_1 \in \mathcal{C}$; a value $k \in \mathcal{K}$ such that $\mathsf{Append}(\mathsf{com}_0, k) = \overline{\mathsf{com}}_0$ and $\mathsf{Append}(\mathsf{com}_1, k) = \overline{\mathsf{com}}_1$.

**Evaluation:**

- The prover computes and sends to the verifier proofs $\pi_0, \pi_1$, such that for $b = 0, 1$ it holds $\pi_b \leftarrow \mathsf{NIZKPK}[(k) \colon \mathsf{Append}(\mathsf{com}_b, k) = \overline{\mathsf{com}}_b]$.
- The verifier checks the proofs and aborts if either check fails. Else, for $i = 1, \ldots, \lambda$, the verifier samples $v_i \xleftarrow{\mathrm{R}} \mathcal{V}$ and $b_i \xleftarrow{\mathrm{R}} \{0,1\}$, computes $\mathsf{com}^{(i)} \leftarrow \mathsf{Append}(\mathsf{com}_{b_i}, v_i)$, and sends $(\mathsf{com}^{(1)}, \ldots, \mathsf{com}^{(\lambda)})$ to the prover.
- The prover uses $k$ to compute, for $i = 1, \ldots, \lambda$, the commitment $\overline{\mathsf{com}}^{(i)} \leftarrow \mathsf{Append}(\mathsf{com}^{(i)}, k)$ and the invariant $j_i \leftarrow \mathsf{Invariant}(\overline{\mathsf{com}}^{(i)})$. It then chooses $s_{\mathsf{out}} \xleftarrow{\mathrm{R}} \{0,1\}^\lambda$, and sends $h \leftarrow H_3(j_1, \ldots, j_\lambda, s_{\mathsf{out}})$ to the verifier.
- The verifier sends $(b_1, v_1, \ldots, b_\lambda, v_\lambda)$ to the prover.
- The prover, for $i = 1, \ldots, \lambda$, checks that $\mathsf{Invariant}(\mathsf{Append}(\mathsf{com}_{b_i}, v_i)) = \mathsf{Invariant}(\mathsf{com}^{(i)})$. If one of the checks fail, the server aborts. Otherwise, it sends $s_{\mathsf{out}}$ to the user.
- The verifier computes the invariants $j'_i = \mathsf{Invariant}(\mathsf{Append}(\overline{\mathsf{com}}_{b_i}, v_i))$ and accepts if $h = H_3(j'_1, \ldots, j'_\lambda, s_{\mathsf{out}})$.

In the full version of this work we prove the following lemma, which shows the soundness of this protocol, and we prove the zero-knowledge property of this protocol as part of security proof of the full protocol.

**Lemma 14.** *Suppose that $\mathcal{G}$ is a secure augmentable commitment scheme, and let $\mathsf{com}_0 = [\![r_0, m_0]\!]$ and $\mathsf{com}_1 = [\![r_1, m_1]\!]$ be two commitments. Then for every efficient prover $P^*$, the probability that the honest verifier of Protocol 13 accepts on input $(\mathsf{com}_0, \mathsf{com}_1, \overline{\mathsf{com}}_0, \overline{\mathsf{com}}_1) \notin L_{\mathsf{eq}}$ when interacting with prover $P^*$ is negligible. Here $L_{\mathsf{eq}}$ is the corresponding language of $R_{\mathsf{eq}}$.*

**Concrete efficiency.** We estimate the communication complexity of the protocol. The communication is dominated by the verifier having to send $\lambda$ augmentable commitments and $\lambda$ values $v_i \in \mathcal{V}$. The size of each supersingular-isogeny-based augmentable commitment is at most $5 \log p$ bits. Moreover, a commitment that includes $v_i \in \mathcal{V}$ as one of its values does not include a basis for the $N_{\mathsf{V}}$-torsion, which cancels out having to send the $v_i$ values in the next message. Therefore, we can bound the overall communication complexity by $5\lambda \log p$ plus the size of the proofs of knowledge $\pi_0$ and $\pi_1$.

## 7 Putting it all together

We now combine the basic protocol from Section 4 with the two protocols from Sections 5 and 6 to obtain a maliciously secure verifiable OPRF.

Protocol 15 implements the OPRF ideal functionality $\mathcal{F}_{\mathsf{VOPRF}}$ as defined in the full version of this work. (That definition is based on [36] with some of the later modifications from [38, 40].)

In the full version of this work we prove the following theorem.

**Theorem 16.** *Suppose that $\mathcal{G}$ is a secure augmentable commitment scheme. Then Protocol 15 realizes ideal functionality $\mathcal{F}_{\mathsf{VOPRF}}$ in the random-oracle model.*

The main ideas of the proof are as follows. The privacy of the user's input easily follows from the hiding property of the underlying augmentable commitment scheme. The main challenge is to simulate the honest server. To this end, the simulator in the ideal world chooses a random secret key for the honest server, and uses it to simulate the interaction of the real-world adversary with that server. Specifically, each time the environment activates the honest server, the simulator responds to an adversary's message by appending its secret key to the commitment sent by the adversary.

The only way the environment can distinguish this from the real world is to find an inconsistency between the value of the OPRF computed via an honest-user honest-server interaction, and the value of the OPRF computed by the adversary directly as $H_2(x, \mathsf{Invariant}([\![m, k]\!]))$ for $m = H_1(x)$. To prevent this inconsistency, whenever the adversary makes this type of query to the random oracle $H_2$, the simulator evaluates the ideal-world OPRF at point $x$ and programs the random oracle $H_2$ to the output value of the PRF. However, the ticketing mechanism of the OPRF ideal functionality limits the number of times the simulator can evaluate the ideal-world OPRF by the number of activations of the honest server. The simulation would therefore fail if the adversary correctly predicts the value $\mathsf{Invariant}([\![m, k]\!])$ on a number of points greater than the number of server activations. However, this would violate the one-more unpredictability property of the underlying augmentable commitment scheme.

The full proof appears in the full version of this work.

### Concrete efficiency and parameter estimation

The communication complexity of the complete OPRF protocol is dominated by the communication complexity of the zero-knowledge proofs. More specifically, the protocol includes 3 $\mathsf{NIZKPK}$s for the relation $R_{\mathsf{com}}$, the size of each of which we have estimated in Section 5 to be $3.8\lambda \cdot (13\lambda + 6 \log p)$. In addition, the complete protocol executes the proof-of-equality sub-protocol once. In Section 6 we estimated the communication complexity of that sub-protocol as $5\lambda \log p$. Therefore, we can bound the communication complexity of the complete protocol as $73\lambda \log p + 148\lambda^2$.

We set $p(\lambda)$ based on the best known attacks on our assumptions. For standard SIDH problems (including the Decisional SIDH problem and the Decisional

**Protocol 15 (Augmentable-Commitment Verifiable OPRF).** The protocol involves a user $U$ and a server $S$. The protocol uses:

- An augmentable commitment scheme $\mathcal{G}$ with $m = 3$ values, $n = 1$ blinds, input space $\mathcal{M} \times \mathcal{K} \times \mathcal{V} \times \mathcal{R}$, and commitment space $\mathcal{C}$.
- A simulation-sound online-extractable $\mathsf{NIZKPK}$ for the relation $R_{\mathsf{com}}$.
- Hash functions, modeled as random oracles:
  - $H_1 \colon \{0,1\}^* \cup \{\epsilon\} \to \mathcal{M}$ (where $\epsilon$ is a special symbol), used to hash PRF inputs to the input space $\mathcal{M}$ of the commitment scheme,
  - $H_2 \colon \{0,1\}^* \to \{0,1\}^\ell$, used to hash to the PRF output space,
  - $H_3 \colon \{0,1\}^* \to \{0,1\}^\lambda$, used in Protocol 13 for proving equality of appended values.

**Initialization** On input $\textsc{Init}$ from the environment, server $S$:

- chooses $k \xleftarrow{R} \mathcal{K}$ and stores it,
- computes $m_\epsilon \leftarrow H_1(\epsilon)$, $r_\epsilon \xleftarrow{R} \mathcal{R}$, and $\mathsf{com}_\epsilon \leftarrow [\![r_\epsilon, m_\epsilon]\!]$.
- computes $\overline{\mathsf{com}}_\epsilon \leftarrow [\![r_\epsilon, m_\epsilon, k]\!]$ and a proof of knowledge of a committed value $\pi_k \leftarrow \mathsf{NIZKPK}[(k) : \mathsf{Append}(\mathsf{com}_\epsilon, k) = \overline{\mathsf{com}}_\epsilon]$,
- stores $\mathsf{pk} = (r_\epsilon, \overline{\mathsf{com}}_\epsilon, \pi_k)$ and outputs $(\textsc{Init}, \mathsf{pk})$.

**Evaluation**

- On input $(\textsc{Eval}, S, x)$, user $U$ proceeds as follows:
  - $m \leftarrow H_1(x)$, $r_m \xleftarrow{R} \mathcal{R}$, $\mathsf{com}_m \leftarrow [\![r_m, m]\!]$
  - compute proof $\pi_m \leftarrow \mathsf{NIZKPK}[(m, r_m) \colon \mathsf{com}_m = [\![r_m, m]\!]]$
  - send message $(\mathsf{com}_m, \pi_m)$ to the server
  - store $(\mathsf{com}_m, r_m)$
- On input $\textsc{ServerComplete}$ from the environment and message $(\mathsf{com}_m, \pi_m)$ from the user, server $S$ verifies the proof $\pi_m$, computes $\overline{\mathsf{com}}_m \leftarrow \mathsf{Append}(\mathsf{com}_m, k)$ and $\pi_m \leftarrow \mathsf{NIZKPK}[(k) : \mathsf{Append}(\mathsf{com}_m, k) = \overline{\mathsf{com}}_m]$, and sends the descriptor $\mathsf{pk} = (r_\epsilon, \overline{\mathsf{com}}_\epsilon, \pi_k)$ and $\overline{\mathsf{com}}_m, \pi_m$ to the user.
- On message $(\mathsf{pk} = (r_\epsilon, \overline{\mathsf{com}}_\epsilon, \pi_k), \overline{\mathsf{com}}_m, \pi_m)$ from the server, user $U$ verifies the proofs $\pi_k$, $\pi_m$.
- The user and server run Protocol 13, in which the sender proves to the user that there exists a $k$ such that $[\![r_\epsilon, m_\epsilon, k]\!] = \overline{\mathsf{com}}_\epsilon$ and $[\![r_m, m, k]\!] = \overline{\mathsf{com}}_m$.
- At the end of the equality protocol, the user, provided it accepts, computes $j \leftarrow \mathsf{Invariant}(\mathsf{Unblind}(\overline{\mathsf{com}}_m, r_m))$ and $y \leftarrow H_2(x, \mathsf{pk}, j)$ and outputs $(\textsc{Eval}, \mathsf{pk}, y)$.

Supersingular Product problem), the best known attacks are meet-in-the-middle attacks that run in time $O(\sqrt{N_i})$ [55]. Although quantum collision-finding algorithms [66] have a better asymptotic running time of $O(\sqrt[3]{N_i})$, recent work [1,34] suggests that the classical algorithm outperform the quantum ones when attacking SIDH, due to the large memory requirement of the quantum algorithms. One caveat is that our one-more assumption admits a better attack than SIDH: Merz

et al. [52] showed an attack on the schemes of [35, 65] that runs in time $N_i^{2/5}$. This *exponential-time* attack, unlike the aforementioned *polynomial-time* attack from the same paper [52], also applies to our one-more assumption. We therefore set $N_i \approx 2^{5\lambda/2}$ for $\lambda$-bit security. (The torsion used for the zero-knowledge proof does not need to be increased as it is used only within a non-interactive proof.) Overall, for $n = 5$ prime powers, the prime $p$ is $12\lambda$-bits long.

Plugging in $\log p = 12\lambda$ into the expression for the communication complexity we have calculated above, we obtain that the total communication complexity is bounded by $1024\lambda^2$ bits. For $\lambda = 128$, the communication complexity is under 2MB.

## 8 Naor-Reingold OPRF from an abelian group action

We now turn to constructing an OPRF from an abelian group action, such as the action obtained from isogenies of *ordinary* elliptic curves or from isogenies of supersingular curves over $\mathbb{F}_p$ as in CSIDH [14].

First, we show that the Naor-Reingold PRF [54] can be adapted to work with an abelian group action that satisfies a DDH-like assumption. Second, we show that the technique used to build an OPRF from the Naor-Reingold PRF carries over to the setting of an abelian group action.

A technical difficulty is that the proof of security of the Naor-Reingold PRF in [54] makes use of the random self reduction of the DDH problem in a prime order group. The DDH problem for an arbitrary abelian group action does not have the required random self reduction. We therefore need to give a new security proof for the Naor-Reingold PRF. We are able to prove security based on the DDH assumption for a group action; however the security reduction is not as efficient as the proof of Naor-Reingold in a prime order group.

Recall that an action of a group $G$ on a set $X$ is a map $G \times X \to X$ such that $(gh) \cdot x = g \cdot (h \cdot x)$ for every $g, h \in G$ and $x \in X$, and $e \cdot x = x$ for every $x \in X$, where $e \in G$ is the identity element of $G$.

Let $G$ be an abelian finite group acting on $S$ transitively and faithfully (we recall the definitions of these properties in the full version of this work), and let $s_0 \in S$ be some fixed element. We define the Naor-Reingold PRF, with key space $\mathcal{K} = G^{n+1}$ and input space $\mathcal{X} = \{0,1\}^n$, as follows:

$$F_{\mathsf{NR}}\Big((k_0, ..., k_n),\ (x_1, ..., x_n)\Big) = (k_0 k_1^{x_1} k_2^{x_2} \dots k_n^{x_n}) \cdot s_0. \tag{2}$$

The security of this PRF requires the following group-action variant of the DDH assumption to hold in $G$:

**Definition 17 (Group-Action DDH [19, 61]).** Let $G$ be an abelian group acting on a set $S$ transitively and faithfully, and let $s \in S$. We say that the *Group-Action DDH assumption* holds in $(G, s)$ if the two distributions

$$\{(a \cdot s,\ b \cdot s,\ (ab) \cdot s) : a, b \xleftarrow{R} G\} \quad \text{and} \quad \{(a \cdot s,\ b \cdot s,\ c \cdot s) : a, b, c \xleftarrow{R} G\}$$

are computationally indistinguishable.

**Theorem 18.** *Suppose that the Group-Action DDH assumption holds in* $(G, s_0)$. *Then the Naor-Reingold PRF* $F_{\mathsf{NR}}$ *is a secure pseudorandom function.*

*Proof sketch.* Boneh et al. [9, Sec. 4.1] show that the Naor-Reingold PRF is a special case of the *augmented cascade*. Therefore, to prove that (2) is a secure PRF, it suffices to show that for every polynomially bounded $Q$, the function

$$P(g, s_1, \ldots, s_Q) = (s_1,\ g \cdot s_1, \ldots, s_Q,\ g \cdot s_Q)$$

is a secure pseudorandom generator (PRG), where $g \in G$ and $s_1, \ldots, s_Q \in S$. This can be done by a simple sequence of $(Q+1)$ hybrid distributions, where at hybrid $i$, for $i = 1, \ldots, Q$, the quantity $g \cdot s_i$ is replaced by random element $t_i$ in $S$. A distinguisher for any pair of consecutive hybrid distributions gives an attack on the Group-Action DDH assumption for $(G, s_0)$. Overall, the reduction incurs a factor of $Q$ loss between an attacker on the PRG and the derived attacker on the Group-Action DDH assumption. The proof of the theorem now follows by [9, Thm. 3]. □

Next, we observe that because the group $G$ is abelian, we can evaluate $F_{\mathsf{NR}}$ obliviously with the following protocol, first described in [24] in a group of prime order.

**Protocol 19.** A client that holds input $(x_1, \ldots, x_n) \in \{0,1\}^n$ and a server that holds input $(k_0, k_1, \ldots, k_n) \in G^{n+1}$ proceed as follows:

1. For each $i = 1, \ldots, n$, the server chooses a random $r_i$ in $G$.
2. For each $i = 1, \ldots, n$, the client and server engage in a 1-out-of-2 oblivious-transfer protocol that gives to the client $r_i$ if $x_i = 0$, and $k_i r_i$ if $x_i = 1$. The client stores the output as $b_i \in G$.
3. The server sends $s' = (k_0 \prod_{i=1}^{n} r_i^{-1}) \cdot s_0$ to the client.
4. The client evaluates $(\prod_{i=1}^{n} b_i) \cdot s'$ to obtain $F_{\mathsf{NR}}$ evaluated at $(x_1, \ldots, x_n)$.

The same security argument from [24, Sec. 5] also applies to this OPRF.

**Instantiation from isogenies.** We can now instantiate the above construction using isogenies. Couveignes [19], Rostovtsev and Stolbunov [61] first proposed using a group action on the set of ordinary elliptic curves. More recently, Castryck et al. [14] proposed CSIDH, a construction that uses the set of supersingular elliptic curves defined over a prime field $\mathbb{F}_p$. Whereas the full endomorphism ring of such curves is non-commutative (and therefore does not give rise to a commutative group action), the subring of $\mathbb{F}_p$-*rational* endomorphisms is an order in an imaginary quadratic field, which gives rise to a commutative group action as in the ordinary case. The main advantage of using the CSIDH group action, over using the group action of ordinary curves, is that it is much more efficient.

More specifically, let $\mathrm{Ell}_p(\mathcal{O})$ be the set of supersingular elliptic curves over $\mathbb{F}_p$ whose $\mathbb{F}_p$-rational endomorphism ring $\mathcal{O}$ is an order in an imaginary quadratic field. The class group $\mathrm{Cl}(\mathcal{O})$, which is an abelian group, acts transitively and

faithfully on $\mathrm{Ell}_p(\mathcal{O})$. (See the full version of this work for additional background.) For $[\mathfrak{a}_0], \ldots, [\mathfrak{a}_n] \in \mathrm{Cl}(\mathcal{O})$ and $E_0 \in \mathrm{Ell}_p(\mathcal{O})$, let

$$F_{\mathsf{NR}}(([\mathfrak{a}_0], [\mathfrak{a}_1], \ldots, [\mathfrak{a}_n], E_0), (x_1, \ldots, x_n)) = j([\mathfrak{a}_n]^{x_n} \ldots [\mathfrak{a}_1]^{x_1}[\mathfrak{a}_0] \cdot E_0)\,.$$

Assuming the hardness of Group-Action DDH problem in the class group, Theorem 18 then implies that $F_{\mathsf{NR}}$ is a PRF. Moreover, instantiating Protocol 19 with the isogeny-based oblivious-transfer protocol of Lai, Galbraith, and de Saint Guilhelm [51], which is secure against malicious adversaries, gives an OPRF protocol from a commutative group action on elliptic curves.

*Remark 20.* Recently, Castryck, Sotáková, and Vercauteren [15] showed that the DDH problem is easy in ideal-class-group actions when the class number is even. Such groups are therefore unsuited for the above construction. As a countermeasure to their attack, they suggest working with supersingular elliptic curves over $\mathbb{F}_p$ for $p \equiv 3 \pmod 4$, which is already the case for CSIDH [14]. In that setting, the Group-Action DDH problem is conjectured to be hard.

*Remark 21.* Our construction targets the case of *commutative* group actions. We mention a recent work by Ji et al. [43], that studies the case of *non-commutative* group actions. The above reduction does not seem to carry over to the non-commutative case, which might explain why Ji et al. require a different assumption.

**Efficiency.** To compute the communication complexity of this instantiation, first assume without loss of generality that $n = \lambda$ (since otherwise we can compose the PRF with a $\lambda$-bit hash function). The protocol requires $n = \lambda$ executions of the OT protocol [51]. Each such execution communicates 3 elliptic curves over $\mathbb{F}_p$, 4 encryptions of class-group elements, and an additional $\lambda$-bit string. Overall, this adds up to $\lambda \cdot (3 \log p + 4 \cdot \log p/2 + \lambda) = 5\lambda \log p + \lambda^2$ bits.

Kuperberg's algorithm [48, 49] for solving the commutative-group-action discrete-log problem, runs in time $\exp(\sqrt{\log(p)})$, which requires setting $p = \Omega(\lambda^2)$. As a result, the overall communication complexity of this protocol is asymptotically $\Omega(\lambda^3)$, compared to $O(\lambda^2)$ communication in the protocol from the previous sections. While the initial CSIDH paper [14] suggested that using a 512-bit prime might be sufficient, recent analysis [10, 56] recommends using primes as large as 5280-bits long. This leads to Protocol 19 having communication complexity of 424KB.

## 9 Conclusions and open problems

We constructed two OPRFs from isogenies on elliptic curves. Our main construction of a verifiable OPRF from isogenies on supersingular elliptic curves is based on a new one-more SIDH assumption. Our construction achieves malicious security by virtue of two new zero-knowledge proofs, and introduces a new abstraction called Augmentable Commitments, which may help simplify the exposition of

future SIDH-based constructions. We also presented a second construction from commutative group actions.

*Future work.* It would be interesting to extend our OPRF to support threshold PRF evaluation, where the PRF key is distributed across multiple servers. Threshold OPRFs [38] have applications to management of passwords and keys [2,32,39]. It would also be good to reduce the communication cost of our zero-knowledge proofs, as that would improve the overall efficiency of the OPRF.

**Acknowledgements.** We would like to thank David Wu for helpful conversations. We thank Henry Corrigan-Gibbs, Michel Dellepere, and Steven Galbraith for giving helpful suggestions that improved this article. Finally, we would like to thank the anonymous Asiacrypt reviewers for their constructive comments. This work was supported in part by DARPA, NSF, ONR, and the Simons Foundation.

## References

1. Adj, G., Cervantes-Vázquez, D., Chi-Domínguez, J., Menezes, A., Rodríguez-Henríquez, F.: On the cost of computing isogenies between supersingular elliptic curves. SAC (2018)
2. Agrawal, S., Miao, P., Mohassel, P., Mukherjee, P.: PASTA: password-based threshold authentication. CCS (2018)
3. Albrecht, M.R., Davidson, A., Deo, A., Smart, N.P.: Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. Cryptology ePrint Archive, Report 2019/1271 (2019)
4. Azarderakhsh, R., Jalali, A., Jao, D., Soukharev, V.: Practical supersingular isogeny group key agreement. Cryptology ePrint Archive, Report 2019/330 (2019)
5. Azarderakhsh, R., Jao, D., Kalach, K., Koziel, B., Leonardi, C.: Key compression for isogeny-based cryptosystems. AsiaPKC@AsiaCCS (2016)
6. Barreto, P., Oliveira, G., Benits, W.: Supersingular isogeny oblivious transfer. Cryptology ePrint Archive, Report 2018/459 (2018)
7. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. J. Cryptology **16**(3), 185–215 (2003)
8. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. PKC (2003)
9. Boneh, D., Montogomery, H., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. CCS (2010)
10. Bonnetain, X., Schrottenloher, A.: Quantum security analysis of CSIDH. EUROCRYPT (2020)
11. Büscher, N., Demmler, D., Karvelas, N.P., Katzenbeisser, S., Krämer, J., Rathee, D., Schneider, T., Struck, P.: Secure two-party computation in a quantum world. ACNS (2020)
12. Camenisch, J., Neven, G., Shelat, A.: Simulatable adaptive oblivious transfer. EUROCRYPT (2007)
13. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. FOCS (2001)
14. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. ASIACRYPT (2018)

15. Castryck, W., Sotáková, J., Vercauteren, F.: Breaking the decisional diffie-hellman problem for class group actions using genus theory. CRYPTO (2020)
16. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. J. Cryptology **22**(1), 93–113 (2009)
17. Chaum, D.: Blind signatures for untraceable payments. CRYPTO (1982)
18. Costello, C., Jao, D., Longa, P., Naehrig, M., Renes, J., Urbanik, D.: Efficient compression of SIDH public keys. EUROCRYPT (2017)
19. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006)
20. Davidson, A., Sullivan, N., Wood, C.: Oblivious pseudorandom functions (OPRFs) using prime-order groups. Internet-Draft draft-irtf-cfrg-voprf01 (2019)
21. Davidson, A., Goldberg, I., Sullivan, N., Tankersley, G., Valsorda, F.: Privacy pass: Bypassing internet challenges anonymously. PoPETs **2018**(3), 164–180 (2018)
22. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Mathematical Cryptology **8**(3), 209–247 (2014)
23. Everspaugh, A., Chatterjee, R., Scott, S., Juels, A., Ristenpart, T.: The Pythia PRF service. USENIX Security (2015)
24. Freedman, M.J., Ishai, Y., Pinkas, B., Reingold, O.: Keyword search and oblivious pseudorandom functions. TCC (2005)
25. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptology **26**(1), 80–101 (2013)
26. Galbraith, S.D.: Authenticated key exchange for SIDH. Cryptology ePrint Archive, Report 2018/266 (2018)
27. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the security of supersingular isogeny cryptosystems. ASIACRYPT (2016)
28. Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. J. Cryptology **33**(1), 130–175 (2020), earlier version in ASIACRYPT 2017
29. Galbraith, S.D., Vercauteren, F.: Computational problems in supersingular elliptic curve isogenies. Quantum Inf. Process. **17**(10), 265 (2018)
30. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
31. Green, M., Hohenberger, S.: Blind identity-based encryption and simulatable oblivious transfer. ASIACRYPT (2007)
32. Harchol, Y., Abraham, I., Pinkas, B.: Distributed SSH key management with proactive RSA threshold signatures. ACNS (2018)
33. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. CRYPTO (2003)
34. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalili, A., Koziel, B., Lamacchia, B., Longa, P., et al.: SIKE: supersingular isogeny key encapsulation (2017)
35. Jao, D., Soukharev, V.: Isogeny-based quantum-resistant undeniable signatures. PQCrypto (2014)
36. Jarecki, S., Kiayias, A., Krawczyk, H.: Round-optimal password-protected secret sharing and T-PAKE in the password-only model. ASIACRYPT (2014)
37. Jarecki, S., Kiayias, A., Krawczyk, H., Xu, J.: Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online). EuroS&P (2016)
38. Jarecki, S., Kiayias, A., Krawczyk, H., Xu, J.: TOPPSS: cost-minimal password-protected secret sharing based on threshold OPRF. ACNS (2017)

39. Jarecki, S., Krawczyk, H., Resch, J.K.: Updatable oblivious key management for storage systems. CCS (2019)
40. Jarecki, S., Krawczyk, H., Xu, J.: OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks. EUROCRYPT (2018)
41. Jarecki, S., Liu, X.: Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. TCC (2009)
42. Jarecki, S., Liu, X.: Fast secure computation of set intersection. SCN (2010)
43. Ji, Z., Qiao, Y., Song, F., Yun, A.: General linear group action on tensors: A candidate for post-quantum cryptography. TCC (2019)
44. Keelveedhi, S., Bellare, M., Ristenpart, T.: Dupless: Server-aided encryption for deduplicated storage. USENIX Security (2013)
45. Kirkwood, D., Lackey, B.C., McVey, J., Motley, M., Solinas, J.A., Tuller, D.: Failure is not an option: standardization issues for post-quantum key agreement. Workshop on Cybersecurity in a Post-Quantum World (2015)
46. Kiss, Á., Liu, J., Schneider, T., Asokan, N., Pinkas, B.: Private set intersection for unequal set sizes with mobile applications. PoPETs **2017**(4), 177–197 (2017)
47. Kolesnikov, V., Kumaresan, R., Rosulek, M., Trieu, N.: Efficient batched oblivious PRF with applications to private set intersection. CCS (2016)
48. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM J. Comput. **35**(1), 170–188 (2005)
49. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. TQC (2013)
50. Kutas, P., Martindale, C., Panny, L., Petit, C., Stange, K.E.: Weak instances of SIDH variants under improved torsion-point attacks. Cryptology ePrint Archive, Report 2020/633 (2020)
51. Lai, Y.F., Galbraith, S.D., de Saint Guilhem, C.D.: Compact, efficient and UC-secure isogeny-based oblivious transfer. Cryptology ePrint Archive, Report 2020/1012 (2020)
52. Merz, S., Minko, R., Petit, C.: Another look at some isogeny hardness assumptions. CT-RSA (2020)
53. Naor, M., Pinkas, B., Reingold, O.: Distributed pseudo-random functions and KDCs. EUROCRYPT (1999)
54. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. FOCS (1997)
55. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with cryptanalytic applications. J. Cryptology **12**(1), 1–28 (1999)
56. Peikert, C.: He gives c-sieves on the CSIDH. EUROCRYPT (2020)
57. Petit, C.: Faster algorithms for isogeny problems using torsion point images. ASIACRYPT (2017)
58. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: Spot-light: Lightweight private set intersection from sparse OT extension. CRYPTO (2019)
59. Pinkas, B., Schneider, T., Zohner, M.: Faster private set intersection based on OT extension. USENIX Security (2014)
60. Pinkas, B., Schneider, T., Zohner, M.: Scalable private set intersection based on OT extension. ACM Trans. Priv. Secur. **21**(2), 7:1–7:35 (2018)
61. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145 (2006)
62. Sahu, R.A., Gini, A., Pal, A.: Supersingular isogeny-based designated verifier blind signature. Cryptology ePrint Archive, Report 2019/1498 (2019)

63. de Saint Guilhem, C.D., Orsini, E., Petit, C., Smart, N.P.: Secure oblivious transfer from semi-commutative masking. Cryptology ePrint Archive, Report 2018/648 (2018)
64. Silverman, J.: The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, Springer New York (2009)
65. Srinath, M.S., Chandrasekaran, V.: Isogeny-based quantum-resistant undeniable blind signature scheme. I. J. Network Security **20**(1), 9–18 (2018)
66. Tani, S.: Claw finding algorithms using quantum walk. Theor. Comput. Sci. **410**(50), 5285–5297 (2009)
67. Thormarker, E.: Post-quantum cryptography: supersingular isogeny Diffie-Hellman key exchange. Ph.D. thesis, Thesis, Stockholm University (2017)
68. Urbanik, D., Jao, D.: Sok: The problem landscape of SIDH. AsiaPKC@AsiaCCS (2018)
69. Vitse, V.: Simple oblivious transfer protocols compatible with supersingular isogenies. AFRICACRYPT (2019)