A Post-Quantum Round-Optimal Oblivious PRF from Isogenies

Andrea Basso

University of Bristol, United Kingdom
University of Birmingham, United Kingdom
andrea.basso@bristol.ac.uk

Abstract. An oblivious pseudorandom function, or OPRF, is an important primitive that is used to build many advanced cryptographic protocols. Despite its relevance, very few post-quantum solutions exist.
In this work, we propose a novel OPRF protocol that is post-quantum, verifiable, round-optimal, and moderately compact. Our protocol is based on a previous SIDH-based construction by Boneh, Kogan, and Woo, which was later shown to be insecure due to an attack on its one-more unpredictability.
We first propose an efficient countermeasure against this attack by redefining the PRF function to use irrational isogenies. This prevents a malicious user from independently evaluating the PRF.
The SIDH-based construction by Boneh, Kogan, and Woo is also vulnerable to the recent attacks on SIDH. We thus demonstrate how to efficiently incorporate the countermeasures against such attacks to obtain a secure OPRF protocol. To achieve this, we also propose the first proof of isogeny knowledge that is compatible with masked torsion points, which may be of independent interest.
Lastly, we design a novel non-interactive proof of knowledge of parallel isogenies, which reduces the number of communication rounds of the OPRF to the theoretically-optimal two.
Putting everything together, we obtain the most compact post-quantum verifiable OPRF protocol.

Keywords: Oblivious Pseudorandom Functions · Isogenies · SIDH

1 Introduction

An oblivious pseudorandom function (OPRF) is a two-party protocol between a user and a server. The two parties obliviously evaluate a PRF on a user-controlled input with a secret key held by the server. After engaging in the protocol, the user learns only the output of the PRF on their chosen input, while the server does not learn anything, neither the user's input nor the output of the function. Oblivious PRFs can also satisfy a stronger property called *verifiability*: in a verifiable OPRF (vOPRF), the server initially commits to its secret key, and during the execution of the protocol it provides a proof that it has behaved honestly and it has used the previously committed secret key.

Oblivious PRFs have widespread applications: they can be used to build password-management systems [ECS$^+$15], adaptive oblivious transfers [JL09], password-protected secret sharing [JKK14], and private set intersection [JL09], which can in turn be used for privacy-preserving contact discovery in messaging services [DRRT18] or for checking compromised credentials [LPA$^+$19]. For instance, the web browser Microsoft Edge uses an OPRF-based protocol to detect compromised passwords. Another practical use case of OPRFs is the privacy-preserving authorisation mechanism known as Privacy Pass [DGS$^+$18]. Developed and currently deployed by Cloudflare, Privacy Pass reduces the number of CAPTCHAs that users need to complete by issuing a number of tokens, which users can spend to avoid solving a second CAPTCHA. To prevent the server (i.e. Cloudflare, in this case) from tracking users across websites, the user queries must be oblivious. OPRFs are also used within OPAQUE [JKX18], a strong asymmetric password-authenticated key exchange that allows a user and a server to authenticate each other based on a shared password with strong security guarantees without the need to communicate the password. For these reasons, there are ongoing efforts to integrate OPAQUE into TLS 1.3[1]. Overall, OPRFs are a fundamental tool for developing privacy-preserving solutions, and they are set to be standardized by the Crypto Forum Research Group (CFRG) [DFHSW23].

[1] https://blog.cloudflare.com/opaque-oblivious-passwords/

It is possible to build an OPRF using generic multi-party computation techniques, but such solutions can be inefficient, and they require more rounds of communication than what an ad-hoc construction can achieve. Indeed, highly-efficient and round-optimal (i.e., with two rounds) constructions exist based on Diffie-Hellman [JKK14] or RSA blind signatures [Cha82]. All such constructions are classical, i.e. they are vulnerable to quantum attacks. The rapid development of quantum computers requires an urgent transition to post-quantum solutions, but very few quantum-resistant OPRFs are reported in the literature. The first quantum-secure verifiable OPRF was proposed by Albrecht, Davidson, Deo and Smart [ADDS21]. The protocol is based on the learning-with-errors problem and the short-integer-solution problem in one dimension, and it only requires two rounds of communication. However, the construction can be characterized as a feasibility result, as a single OPRF execution requires communicating hundreds of gigabytes of data. The only other post-quantum solutions were proposed by Boneh, Kogan, and Woo [BKW20]. The authors proposed two moderately-compact OPRFs based on isogenies, one relying on SIDH and one on CSIDH. The protocol based on CSIDH is a non-verifiable, three-round OPRF, which is obtained by combining a Naor-Reingold PRF [NR97] with a CSIDH-based oblivious transfer protocol [LGd21] to make the PRF evaluation oblivious. The OPRF based on SIDH is verifiable, but requires an even higher number of communication rounds, since the verifiability proof is highly interactive. A later work by Basso, Kutas, Merz, Petit and Sanso [BKM$^+$21] cryptanalyzed the SIDH-based OPRF by demonstrating two attacks against the one-more unpredictability of the protocol, i.e. it showed that a malicious user can recover sufficient information to independently evaluate the PRF on any input. The first attack is polynomial-time, but it can be easily prevented with a simple countermeasure; the second attack is subexponential but still practical, and the authors argue that there is no simple countermeasure against it. More recently, a series of works [CD22,MM22,Rob22] developed an efficient attack on SIDH that also extends to the SIDH-based OPRF.

Contributions. In this work, we propose an OPRF protocol that is post-quantum secure, verifiable, round-optimal, and moderately compact (≈9 MB per execution), with a security proof in the UC framework [Can01] in the random-oracle model. To do so, we follow the same high-level approach as the SIDH-based OPRF by Boneh, Kogan, Woo [BKW20], but with the following changes:

- We propose an efficient countermeasure against the one-more unpredictability attack by Basso, Kutas, Merz, Petit, and Sanso [BKM$^+$21]. We modify the PRF definition, and in particular we use irrational isogenies to map the user's input to an elliptic curve. In this way, the information that allowed an attacker to independently evaluate the PRF is no longer defined over a field of small extension. A malicious user may still attempt to carry out the attack from [BKM$^+$21], but this would now require the attacker to work with points with exponentially many coordinates over the base field, which makes the attack infeasible. Besides preventing the attack, this change results in a smaller prime and a more compact protocol.
- We discuss how to integrate MSIDH, a recently-proposed countermeasure [FMP23] against the SIDH attacks that relies on masked torsion, into the OPRF protocol. This requires using longer isogenies and a larger prime, but a series of optimizations allow us to maintain a reasonable communication cost. To integrate MSIDH, we also propose the first zero-knowledge proof of knowledge that can guarantee the correctness of an MSIDH public key, which may be of independent interest. The proof relies on splitting the masking value into three multiplicative shares: this enables the prover to build a commutative SIDH square and reveal a single edge, together with the torsion point images, without leaking any information about the witness. Repeating the process multiple times yields a proof with negligible soundness error.
- We propose a novel proof of knowledge that can guarantee that two isogenies are parallel, i.e. they are computed by applying the same secret key to two starting curves and torsion points. The protocol is obtained by evaluating two proofs of isogeny knowledge in parallel *with correlated randomness*. The proof can be proved zero-knowledge under a new yet mild assumption. Such a proof can be used by the server to show that it has used a previously committed secret key, which is the key ingredient to make the OPRF verifiable. Since the proof is a proof of knowledge, it can be made non-interactive through standard transformations; this makes the proposed OPRF the first isogeny-based OPRF to be round-optimal.

Paper organization. In Section 2, we introduce the main constructions needed for the rest of the paper. In Section 3, we present the OPRF ideal functionality, together with the security notions and assumptions needed to implement it. Section 4 presents novel countermeasures against the one-more unpredictability attack by Basso, Kutas, Merz, Petit and Sanso, while Section 5 discusses how to prevent the SIDH attacks, and Section 6 presents the new proof of parallel isogeny used to achieve verifiability. In Section 7, we put everything together to obtain the new OPRF protocol, estimate its communication complexity, and compare it with the existing solutions.

2 Preliminaries

In this section, we present the notation used in the rest of the paper, and we briefly introduce the SIDH protocol, the recent attacks on SIDH, the OPRF construction by Boneh, Kogan, and Woo [BKW20], and the attack on the protocol by Basso, Kutas, Merz, Petit, and Sanso [BKM+21].

2.1 SIDH

The Supersingular Isogeny Diffie-Hellman (SIDH) [JD11] is a key-exchange protocol based on isogenies between supersingular elliptic curves. For information on elliptic curves and isogenies, we refer the reader to [Sil09]. The protocol parameters include a prime p of the form $p = ABf - 1$, where A and B are smooth coprime integers and f a small cofactor, a supersingular curve E_0 defined over $\mathbb{F}_{p^2}$, and the basis P_A, Q_A and P_B, Q_B that span, respectively, $E_0[A]$ and $E_0[B]$. One party samples a secret key $a \overset{\$}{\leftarrow} \mathbb{Z}_A$, computes the isogeny $\phi_A : E_0 \to E_A \coloneqq E_0/\langle P_A + [a]Q_A\rangle$, and publishes $\mathsf{pk}_A = (E_A, R_A = \phi_A(P_B), S_A = \phi_A(Q_B))$. The second party proceeds similarly by sampling a secret key $b \overset{\$}{\leftarrow} \mathbb{Z}_B$, computing $\phi_B : E_0 \to E_B \coloneqq E_0/\langle P_B + [b]Q_B\rangle$, and revealing $\mathsf{pk}_B = (E_B, R_B = \phi_B(P_A), S_B = \phi_B(Q_A))$. Then, both parties can agree on a shared secret by translating their secret isogeny to the starting curve in the other party's public key using the revealed torsion information. In other words, the first party computes $\phi'_A : E_B \to E_{AB} \coloneqq E_B/\langle R_B + [a]S_B\rangle$, and the second party computes $\phi'_B : E_A \to E_{BA} \coloneqq E_A/\langle R_A + [b]S_A\rangle$. The codomain curves E_{AB} and E_{BA} are isomorphic, and thus their j-invariant is the same and provides the shared secret of the key exchange. Note that it is possible to represent the points in the public keys more compactly than two x-coordinates, which requires $4\log p$ bits. If the points are expressed in terms of a canonical basis, i.e. a basis deterministically computed from the curve, their coefficients only require $4\log A$ or $4\log B$ bits [AJK+16,CLN16]. In the rest of the paper, we write $P, Q = \mathcal{B}_N(E)$ for a canonical basis of order N on E. We also refer to the setup described above as the *SIDH square* (E_0, E_A, E_B, E_{AB}) with edges $(\phi_A, \phi_B, \phi'_A, \phi'_B)$.

Generally, isogenies do not commute, which means that two parties computing an SIDH-like exchange would not agree on a shared secret if they only revealed the isogeny codomain. To avoid the problem, SIDH reveals the image of a torsion basis that allows each party to translate their isogeny such that they commute. Torsion points are thus a key element of the SIDH protocol, but they also allow attackers to perform adaptive attacks against static-key SIDH [GPST16]. To prevent such attacks, both parties can provide a proof of torsion point correctness, such as the proof proposed in [BKW20,DFDGZ22]. Unfortunately, revealing the torsion point images also enabled the recent passive attacks on SIDH.

The SIDH attacks. The security of the SIDH protocol hinges on the hardness of recomputing the secret isogenies given the public key. While the problem of finding an isogeny between two curves is believed to be hard, the presence of torsion point images in SIDH makes it easier since more information is revealed about the secret isogeny. In a series of works by Castryck and Decru [CD22], Maino and Martindale [MM22], and Robert [Rob22], the authors propose a polynomial-time algorithm that can compute an isogeny of smooth degree d given the domain and codomain curves, the degree d, and the image of a torsion basis of order at least $\sqrt{d}$. This fully breaks the SIDH key exchange and all protocols based on it. Some counteremasures have been proposed [FMP23], based on either secret-degree isogenies or on masked torsion images. We discuss these approaches in the context of the OPRF protocol in Section 5.

2.2 The OPRF construction by Boneh, Kogan, Woo

Boneh, Kogan, and Woo [BKW20] introduced a verifiable OPRF protocol based on SIDH, which uses a prime p of the form $p = N_{\mathsf{M}} N_{\mathsf{B}} N_{\mathsf{K}} N_1 N_2 f - 1$, where the values N_i are coprime smooth integers and f is a small cofactor. Initially, the server commits to its key k by publishing the curve E_C obtained as the codomain of the N_{K}-isogeny starting from $\tilde{E}$ with kernel $\langle \tilde{P} + [k]\tilde{Q} \rangle$, where the values $\tilde{E}, \tilde{P}, \tilde{Q}$ are protocol parameters. The commitment also include a zero-knowledge proof π_C of the correctness of the computation. Then, to evaluate the PRF on input $m \in \mathcal{M}$, where $\mathcal{M}$ defines the input space, the user computes an isogeny ϕ_m of degree N_{M} by hashing the input with $H : \mathcal{M} \to \mathbb{Z}_{N_{\mathsf{M}}}$ and computing $\phi_m : E_0 \to E_m := E_0 / \langle P + [H(m)]Q \rangle$, where the curve E_0 and the points P, Q are also protocol parameters. Then, the user blinds the curve E_m by computing a second isogeny $\phi_b : E_m \to E_{mb}$ of degree N_{B}. The user sends the curve E_{mb} and the torsion images $R_K = \phi_b \circ \phi_m(P_K), S_K = \phi_b \circ \phi_m(Q_K)$ to the server, where the points P_K, Q_K are also protocol parameters of order N_{K}. The user also provides a non-interactive zero-knowledge proof that torsion information was honestly computed. The server validates the proof, computes the isogeny $\phi_k : E_{mb} \to E_{mbk} := E_{mb} / \langle R_K + [k]S_K \rangle$ based on its secret key k, and sends the curve E_{mrk}, the image $\phi_k(E_{mb}[N_{\mathsf{B}}])$, and a non-interactive zero-knowledge proof of correctness to the user. Then, the server and the user engage in an interactive protocol where the server proves that the isogeny ϕ_k has used the same key k as the committed value. If the user is convinced, they use the provided torsion information to undo the blinding isogeny, i.e. to compute the translation of the dual of the blinding isogeny, to obtain the curve E_{mk}. The output of the OPRF is then the hash $H\big(m, j(E_{mk}), (E_C, \pi_C)\big)$. The main exchange, without the commitments and the proofs, is represented in Fig. 1.

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] \arrow[r, squiggly, shift right=2] \arrow[dd, dashed, "\phi'_k"] & E_m \arrow[dr, "\phi_b"] \arrow[dd, squiggly] & \\
 & & E_{mb} \arrow[dd, "\phi_k"] \\
E_k \arrow[r, dashed, "\phi'_m"] & E_{mk} & \\
 & & E_{mbk} \arrow[ul, "\hat{\phi}'_b"]
\end{tikzcd}$$

Fig. 1. The OPRF construction by Boneh, Kogan, and Woo. The protocol, without the relevant zero-knowledge proofs, is represented by the solid lines: the isogenies $\longrightarrow$ (ϕ_m, ϕ_b, $\hat{\phi}'_b$) are computed by the client, while the isogeny $\longrightarrow$ (ϕ_k) is computed by the server. The isogenies $\rightsquigarrow$ represent the PRF evaluation, and the isogenies $\dashrightarrow$ are relevant to the attack presented in [BKM+21]. The figure is based on Fig 1 of [BKM+21].

2.3 The BKMPS attacks

Basso, Kutas, Merz, Petit, and Sanso [BKM+21] proposed two attacks against the one-more unpredictability of the OPRF protocol by Boneh, Kogan, Woo [BKW20] OPRF. We refer to them as the BKMPS attacks.

In the first attack, an attacker who acts as a malicious user engages in the OPRF with a message isogeny ϕ_m with kernel generator a point M, of order ℓ^e. The attacker repeats the process with message isogenies with kernel generators $[\ell]M, [\ell^2]M, \dots, [\ell^e]M$. The outputs of the PRF are the curves that lie on the isogeny path of $\phi'_m : E_k \to E_{mk}$ (see Fig. 1), which allows the attacker to compute a generator of the kernel of such isogeny. The recomputed generator is a scalar multiple $\phi'_k(M)$, where ϕ'_k is the isogeny parallel to the server's secret isogeny, i.e. its kernel is generated by $P_k + [k]Q_k$. By repeating this process three times with points M_1, M_2 and $M_3 \coloneqq M_1 + M_2$ (where M_1 and M_2 are linearly independent), the attacker obtains

$$R \coloneqq [\alpha]\phi'_k(M_0), \quad S \coloneqq [\beta]\phi'_k(M_1),$$
$$T \coloneqq [\gamma]\phi'_k(M_3) = [\gamma/\alpha]R + [\gamma/\beta]S,$$

for some unknown values α, β, γ. By expressing T in terms of R and S, the attacker obtains the values γ/α and γ/β and the points $R' \coloneqq [\gamma/\alpha]R = [\gamma]\phi'_k(M_0)$ and $S' \coloneqq [\gamma/\beta]S = [\gamma]\phi'_k(M_1)$. Finally, the attacker can evaluate the PRF on any input m by computing $E_K/\langle R' + [H(m)]S'\rangle$. The attack is polynomial time, but it crucially rely on using message isogenies ϕ_m of varying degree. The attack can be thwarted by server checking the order of the isogeny ϕ_m, which is possible because of the proof of knowledge provided by user.

The authors of [BKM+21] also propose a second attack that cannot be easily prevented. The attack procceeds in a similar way to the previous one, but the malicious user uses only isogenies of full degree. To obtain the curves on the path of ϕ_m, the attacker needs to find the middle curve between two PRF outputs. This introduces a trade-off between the complexity of the attack and the number of queries. Minimizing both yields a subexponential yet practical attack on the one-more unpredictability of the protocol.

3 Oblivious pseudorandom functions

Oblivious pseudorandom functions were originally proposed by Naor and Reingold [NR97], who defined an OPRF via an ideal functionality. Subsequent work [FIPR05,JL09] defined OPRFs in terms of the two-party computation $(k, x) \mapsto (\perp, f(k, x))$, but such a definition has several drawbacks. On one side, it is hard to build protocols that satisfy such a definition, because the security proof would require extracting the user's input, while at the same the definition is not secure enough, because it does not guarantee any security under composability. Since OPRFs are mainly used as builiding blocks in larger protocols, such a security guarantee is highly needed. For these reasons, Jarecki et al. [JKK14,JKKX16,JKX18] proposed new definitions in the UC framework [Can01]. To avoid extracting the user's input, the ideal functionality introduces a ticketing system that increases a counter when the PRF is evaluated and decreases the counter when the user receives the PRF output. This captures the idea that a malicious user should learn only the PRF output for one input for each interaction. This results in the definition of Fig. 2, which is based on the definitions by Jarecki et al. [JKK14,JKKX17,JKX18].

3.1 Security assumptions

To prove that the OPRF protocol we propose implements the functionality of Fig. 2, we will make use of the properties listed in this section. Since our protocol and security proof follows the same high-level structure as that of the OPRF protocol by Boneh, Kogan, and Woo [BKW20], these properties are also based on those of the augmentable commitment framework proposed in [BKW20]. Unlike [BKW20], we avoid the abstraction of augmentable commitments due to its restrictiveness (the counteremasures of Section 4 would not be possible within that framework), and we prefer an explicit description throughout this work.

Correctness. Firstly, we require the OPRF to be correct, i.e. the output of the protocol is the output of function that deterministically depends only on the user's input and the server's secret key. In other words, we want that the blinding process that guarantees the obliviousness of the user's input does not affect the final output. In the context of our protocol, we want that the unblinding isogeny undoes the effect of the blinding isogeny. This is contained in the following lemma, whose proof follows from the correctness of the SIDH protocol [JD11].

Parameters: The PRF output ℓ, polynomial in the security parameter λ.

Convention: For every identifier S, the counter $\mathsf{tx}[S]$ is initially set to zero. For every value $\pi \in \{0,1\}^*$ and $x \in \{0,1\}^*$, the value $F(\pi, x)$ is initially undefined, and whenever such a value is referenced, the functionality assigns a random ℓ-bit string $F(\pi, x) \xleftarrow{\$} \{0,1\}^\ell$.

Initialization

- On message INIT from party S, forward (INIT, S) to the adversary $\mathcal{A}$.
- On message (PARAM, S, π) from adversary $\mathcal{A}$, if $\mathsf{param}[S]$ is undefined, set $\mathsf{param}[S] = \pi$.

Evaluation

- On message (EVAL, S, x) from $P \in \{U, \mathcal{A}\}$, record $\langle P, x\rangle$ and forward the message (EVAL, P, S) to $\mathcal{A}$.
- On message SERVERCOMPLETE from server S, send $(\text{SERVERCOMPLETE}, S)$ to $\mathcal{A}$ and increment $\mathsf{tx}[S]$.
- On message $(\text{USERCOMPLETE}, P, \pi)$ from $\mathcal{A}$, retrieve the record $\langle P, x\rangle$, delete it from the list of records, and decrement $\mathsf{tx}[S]$ if there exists an honest server S such that $\mathsf{param}[S] = \pi$; abort if no such record exists or if $\mathsf{tx}[S] = 0$. Then, send $(\text{EVAL}, \pi, F(\pi, x))$ to P.

Fig. 2. Functionality $\mathcal{F}_{\mathsf{vOPRF}}$.

Lemma 1 (Correctness). *Let p be a prime of the form $p = N_{\mathsf{B}}N_{\mathsf{K}}f - 1$, where N_B, N_K, f are smooth coprime integers. Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ and let P_B, Q_B and P_K, Q_K be respectively a basis of $E_0[N_{\mathsf{B}}]$ and $E_0[N_{\mathsf{K}}]$. Let also b and k be two values in $\mathbb{Z}_{N_{\mathsf{B}}}$ and $\mathbb{Z}_{N_K}$. Then, consider the isogenies*

$$\begin{aligned}\phi_B &: E_0 \to E_B \coloneqq E_0/\langle P_B + [b]Q_B\rangle,\\ \phi_K &: E_0 \to E_K \coloneqq E_0/\langle P_K + [k]Q_K\rangle,\\ \phi'_k &: E_B \to E_{BK} \coloneqq E_B/\langle \phi_B(P_K) + [k]\phi_B(Q_K)\rangle.\end{aligned}$$

If R_B, S_B is a basis of $E_B[N_{\mathsf{B}}]$ and the values b_0, b_1 satisfy $\ker\hat{\phi}_B = \langle [b_0]R_B + [b_1]S_B\rangle$, then we have

$$j\left(E_{BK}/\langle [b_0]\phi'_k(R_B) + [b_1]\phi'_k(S_B)\rangle\right) = j(E_K).$$

Input hiding. To ensure that the OPRF is oblivious, we want that the server does not learn the user's input. That holds in the strongest sense, i.e. the server should not learn the user's input even when the input is randomly chosen between two inputs *chosen by the server*. In other words, the user must apply a blinding step that fully hides the chosen input. In the context of isogenies, we want the following problem to be hard.

Problem 1. Let p be a prime of the form $p = N_{\mathsf{B}}N_{\mathsf{K}}f - 1$, where $N_{\mathsf{B}}N_{\mathsf{K}}, f$ are smooth coprime integers. Let E_0 and E_1 be two supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ and chosen by the adversary, and let P_0, Q_0 and P_1, Q_1 be a basis of $E_0[N_{\mathsf{K}}]$ and $E_1[N_{\mathsf{B}}]$, respectively, such that $e_{N_{\mathsf{K}}}(P_0, Q_0) = e_{N_{\mathsf{K}}}(P_1, Q_1)$. Let i be a random bit, i.e. $i \xleftarrow{\$} \{0,1\}$, and B a random point in $E_i[N_{\mathsf{B}}]$, and write $\phi : E_i \to E' := E_i/\langle B\rangle$. Output i given E' and $f(\phi(P_i), \phi(Q_i))$, where the latter is some auxiliary torsion information.

The hardness of the problem clearly depends on the function f; if the torsion images were directly revealed, Problem 1 would be easy due to the SIDH attacks. We thus delay specifying the function f until Section 5, where we discuss the SIDH counteremasures to use within the OPRF protocol. In the section, we also state the variant of the Decisional Isogeny problem that Problem 1 reduces to.

One-more unpredictability. A key property of an OPRF is that the user learns the output of the PRF only on its input of choice. That means that a malicious user should not learn the output on more inputs than the number of OPRF executions. The BKMPS attack [BKM+21] on the OPRF by Boneh, Kogan, and Woo [BKW20] targets the one-more unpredictability, since it shows that a malicious user can extract enough information to indipendently evaluate the OPRF on any input of their choice. We propose an efficient counteremasure against the one-more unpredictability attack in the next section; we thus delay until then a formalization of the isogeny-related assumption (see Problem 4) we need to guarantee the one-more unpredictability of the OPRF protocol.

Committment binding. At the beginning of the OPRF protocol, the server commits to a secret key k, so that during each OPRF execution it can prove that the same key was used. To guarantee verifiability, we want a commitment scheme with an associated proof of input reuse. We propose to commit to a key k by fixing a special curve $\tilde{E}$ with a basis $\tilde{P}, \tilde{Q}$ of $\tilde{E}[N_{\mathsf{K}}]$ and revealing $j(\tilde{E}/\langle \tilde{P} + [k]\tilde{Q}\rangle)$. The proof of input reuse, which in the context of isogenies becomes a proof of parallel isogenies, is presented in Section 5.2. To guarantee that the committment is binding, we want that the following problem to be hard.

Problem 2 (Collision finding problem). Let E_0 be a supersingular elliptic curve of unknown endomorphism ring. Find two distinct cyclic isogenies $\phi_0 : E_0 \to E$ and $\phi_1 : E_0 \to E'$ such that $j(E) = j(E')$.

Problem 2 has been studied in the context of the CGL hash function [CLG09], and it has been shown to be heuristically equivalent to the following problem, which underpins every isogeny-based protocol [PL17,EHL+18].

Problem 3 (Endomorphism Ring problem). Let E be a supersingular elliptic curve. Find $\mathrm{End}(E)$.

4 Countermeasures against the one-more unpredictability attack

The original protocol by Boneh, Kogan and Woo starts by mapping an input m to an isogeny ϕ_m. If we denote with N_{M} the torsion space dedicated to the message, the protocol fixes a basis P, Q of $E_0[N_{\mathsf{M}}]$ and computes the isogeny ϕ_m given by

$$\phi_m : E_0 \to E_0/\langle P + [H(m)]Q\rangle =: E_m, \tag{1}$$

where $H(\cdot)$ maps the message m onto an element of $\mathbb{Z}_{N_{\mathsf{M}}}$.

The subexponential attack [BKM+21] recovers the image P_k, Q_k of the torsion basis P, Q, up to scalar multiplication, under the secret isogeny $\phi'_k : E_0 \to E_k$. With such information, the attacker can evaluate the PRF on any input of their choice. The output curve of the PRF is the curve computed as $E_k/\langle P_k + [H(m)]Q_k\rangle$. Any countermeasures against such an attack need to prevent the attacker from evaluating the OPRF without interacting with the server. A first approach might try to prevent the attacker from recovering the points P_k, Q_k altogether, but it appears to be hard since the curve E_k is fixed, because the server needs to hold a long-term static key to satisfy the OPRF definition, which in turn also fixes the curve E_k. Moreover, the BKMPS attack could be prevented by requiring the user to send only honestly-generated queries. The attacker needs to send carefully-chosen queries, where the kernel of the message isogeny does not necessarily satisfy Eq. (1). However, there is no simple way to prove in zero-knowledge that the kernel was honestly computed, besides using very expensive generic techniques. An other approach might require to simply increase the parameters. The attack is subexponential, and it is possible to obtain λ bits of security if the isogeny ϕ_m has degree 2^{λ^2} (this can be reduced if we limit the number of queries the attacker can make). This would require using very long isogenies (the degree would be $2^{16,384}$ for $\lambda = 128$) and very large primes.

Instead, in this section we propose a novel and efficient countermeasure that sidesteps these issues. Our main idea is to accept that an attacker may recover the curve E_k and points P_k, Q_k on it, but to prevent those points from being sufficient to evaluate the desired isogeny. To do so, we require that the isogeny ϕ_m has an irrational kernel, i.e. its kernel is defined over a sufficiently-large extension field. Such an isogeny can be efficiently computed as a composition of rational isogenies. More formally, assume that $N_{\mathsf{M}} = \ell^e$, and e is the highest power of ℓ that divides $p + 1$. Then, given an input $m \in \mathcal{M}$, we compute the isogeny ϕ_m in the following way:

1. We first map the message m to two elements in $\mathbb{Z}_{\ell^e}$ through two hash functions H_0, H_1 that are collision resistant. We thus have $m_0 = H_0(m)$ and $m_1 = H_1(m)$.
2. Given the starting curve E_0 and two points P_0, Q_0 spanning $E_0[\ell^e]$, we compute the isogeny
$$\phi_0 : E_0 \to E_1 \coloneqq E_0/\langle P_0 + [m_0]Q_0 \rangle.$$
3. We determine a canonical basis P_1, Q_1 of $E_1[\ell^e]$ and compute the isogeny
$$\phi_1 : E_1 \to E_m \coloneqq E_1/\langle P_1 + [m_1]Q_1 \rangle,$$
4. The isogeny $\phi_m : E_0 \to E_m$ is the composition $\phi_1 \circ \phi_0$.

An attacker may still try to apply the one-more unpredictability attack. In the original case, the attacker recovers three isogenies from E_k to E_{mk} and they combine their kernel generators to obtain the image points P_k, Q_k. In the proposed construction, the attacker can still recover three (or more) isogenies from E_k to E_{mk}. However, the kernel generators of these isogenies are points of order ℓ^{2e}, and thus they are defined only over the extension field $\mathbb{F}_{p^{2\ell^e}}$. This is an exponentially large field, and even just representing such a point—let alone doing any computation—would be exponential in the security parameter. To guarantee security, it is important that the degree of ϕ_m is a prime power. If the degree were a product of prime powers, it is possible to represent a large extension by working over several smaller extensions because of the Chinese Remainder Theorem. This can reduce the complexity of working over a large extension and thus reduce the security of the proposed countermeasures.

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_0"] \arrow[d, dashed] & E_1 \arrow[r, "\phi_1"] \arrow[d, dashed] & E_m \arrow[d, dashed] \\
E_k \arrow[r, "\phi_{m_0}"'] & E_{m_0k} \arrow[r, dotted, "\phi_{m_1}"'] & E_{mk}
\end{tikzcd}$$

Fig. 3. Summary of the proposed countermeasure (this does not depict the blinding/unblinding phase). Isogenies in red are known or can be computed by the attacker, isogenies in black are unknown to the attacker, and the dotted isogeny represents the missing isogeny that the attacker needs to compute to succeed in the attack.

The attacker can work with the kernel generators of only the first half of the isogenies and obtain a basis P_k, Q_k of order ℓ^e (see Fig. 3). This allows them to evaluate the first isogeny ϕ_{m_0} to obtain the curve E_{m_0k} for any message m. However, the attacker has no way of computing the remaining isogeny ϕ_{m_1}. To do so, the attacker would need to map the canonical basis on E_1 to E_{m_0k}, which does not seem to be possible without knowing the server secret key. Alternatively, the attacker could map the points P, Q and P_k, Q_k under the isogenies ϕ_0 and ϕ'_k. At least one of the image points on each curve has full order, and the point of full order on E_{m_0k} is the image of the point of full order on E_1. This suggest such an approach could be used to find a basis, but the second point on each curve is always a scalar multiple of the first point[2]. Hence, guessing the remaining point has exponential complexity ℓ^e. Lastly, the attacker cannot use a similar strategy as the one-more unpredictability attack to recover a basis on E_{m_0k} because the curve E_{m_0k} depends on the message m. It thus changes at every interaction, and it is hard for an attacker to find two messages that have the same first curve E_1 and E_{m_0k} since we assume that the hash function H_0 is collision-resistant. Note that we require H_0 and H_1 to be collision-resistant, but we conjecturize that only H_0 needs to be. Overall, the knowledge of E_{m_0k} does not help the attacker learn any information on the curve E_{mk}, which successfully prevents the the one-more unpredictability attack.

[2] If $\ker \phi = \langle P + \alpha Q \rangle$, it follows that $\phi(P) = -\alpha\phi(Q)$.

Optimizations. We can extend this approach to obtain a more compact protocol. Rather than limiting ourselves to two isogenies, we can extend this to an arbitrary number. Let I be an integer greater than one, and let H_i be distinct functions for every $i \in \{1, \dots, I\}$, which are modelled as random oracles. Then, given an input m and a starting curve E_0, the isogeny ϕ_m and the curve E_m can be computed as in Algorithm 1. This modification can result in a more compact OPRF protocol because only the smaller isogenies ϕ_i need to be defined over $\mathbb{F}_{p^2}$; thus, using more isogenies can result in a smaller prime p while maintaining the same degree of the isogeny ϕ_m. In this case, note that the functions H_i cannot be collision-resistant if their output space becomes smaller than $2^{2\lambda}$; however, the case where $I > 2$ is clearly more secure than $I = 2$ because an attacker can recover even less information. Since we require H_0 to be collision-resistant when $I = 2$, in the case $I > 2$ it is thus sufficient to ask that the concatenation $H_0(x)||H_1(x)||\dots||H_n(x)$ is collision-resistant where n is the smallest value such that the concatenation output is larger than $2^{2\lambda}$. In other words, the functions H_i can be obtained by splicing the output a collision-resistant hash function.

In the rest of the paper, we write $(\phi_m, E_m) = \mathcal{H}_I(x)$ to refer to the function in Algorithm 1; we also write $[P_0, P_1, \dots, P_{I-1}]_{E,N}$ to denote a list points of order N where the point P_0 belongs to E, and the point P_i belongs to $E_i \coloneqq E_{i_1}/\langle P_{i-1}\rangle$. We refer to this as a *sequence*, whose associated isogeny is the composition of the isogenies $E_i \to E_i/\langle P_i\rangle$.

Algorithm 1 Function $\mathcal{H}_I$ mapping the input m to the curve E_m

```
for $i \leftarrow 0$ to $I-1$ do
    Set $m_i = H_i(m)$;
    Set $P_i, Q_i = \mathcal{B}_M(E)$;
    Compute $\phi_i : E_i \to E_{i+1} \coloneqq E_i/\langle P_i + [m_i]Q_i\rangle$;
Set $\phi_m = \phi_{I-1} \circ \dots \circ \phi_0$;
Set $E_m = E_{I-1}$;
return $\phi_m, E_m$;
```

A new assumption. We proposed a modified protocol that prevents the existing one-more unpredictability attacks. As in the original construction, the one-more unpredictability of the resulting protocol relies on the hardness of a novel problem, which is the following.

Problem 4 (One-more unpredictability). Let p be a prime of the form $p = N_\mathsf{M} N_\mathsf{K} f - 1$, where N_M and N_K are smooth coprime integers, and f a cofactor. Let $\mathcal{H}_I$ be a function as in Algorithm 1. Let E_0 be a supersingular curve defined over $\mathbb{F}_{p^2}$, and let K be a point on E_0 of order N_K. Write ϕ_K for the isogeny $\phi_K : E_0 \to E_K \coloneqq E_0/\langle K\rangle$. Given the curves E_0, E_K and an oracle that responds to the following queries:

- challenge: returns a random sequence $[M_0, \dots, M_{I-1}]_{E_0,N_\mathsf{M}}$,
- solve($[V_0, \dots, V_{I-1}]_{E_0,N_\mathsf{M}}$): returns $j(E_V/\langle\phi_V(K)\rangle)$, where ϕ_V is the isogeny associated to the input sequence,
- decide(i, j): returns true if j is equal to the output of a solve query with input the response of the i-th challenge query, and false otherwise,

For any value n, produce n pairs (i, j) such that decide(i, j) = true with less than n solve queries.

The problem is based on Game 12 of [BKW20], but compared to it, this game involves multiple points during the challenge and solve query to abstract the behavior described in the previous section. Moreover, the problem includes the countermeasures against the polynomial time attack of [BKM+21], i.e. the attacker can only query points of the correct order. This can be replicated in the OPRF setting by checking the order of the isogenies in the proof of isogeny knowledge. We included these countermeasures to prevent possible attacks since they are inexpensive. However, we conjecture that the problem remains hard even if the adversary is allowed to submit solve queries with points of arbitrary order. Furthermore, the problem remains hard after the SIDH attacks since it does not involve exchanging any torsion points.

Countermeasure costs. We briefly discuss the impact of the proposed countermeasures on the performance of the OPRF protocol. Firstly, we need to determine the parameters ℓ, e, and I. In the previous section, we presented two possible attacks on the Auxiliary One-More SIDH assumption: the first requires to work over the extension field $\mathbb{F}_{p^{2\ell^e}}$, while the second obtains a point of full order on E_1 and its image on E_{m_0k}, fixes a linearly-independent point on E_1, and then guesses its image E_{m_0k}. The first attack involves using points with coordinates with ℓ^e values in $\mathbb{F}_p$, while the second requires guessing the correct image out of the ℓ^e possibilities. It may be tempting to set $\ell^e \approx 2^\lambda$, but when $I = 2$ we require that the hash functions H_0, H_1 are collision-resistant, so their output space must be larger than $2^{2\lambda}$. Hence, we set $\ell^e \approx 2^{2\lambda}$ and the degree of ϕ_m to be $2^{4\lambda}$.

If we want to minimize the bandwidth consumption of the protocol, we can set $e = 1$ and $\ell^I \approx 2^{4\lambda}$. Moreover, we can choose ℓ such that isogenies of degree ℓ are defined over a small extension, such $\mathbb{F}_{p^4}$, rather $\mathbb{F}_{p^2}$.

The choice of e thus determines the size of the message component N_M and the prime p: if $I = 2$, the message component N_M is already smaller than the value N_M in the original construction, which used $N_\mathsf{M} \approx 2^{5/2\lambda}$. If $e = 1$, the message component N_M is one, since the prime p does not need to change to allow computations of the message isogeny. This means that not only do the proposed countermeasures protect against existing attacks, but also they reduce the prime size leading to a more compact and efficient protocol.

5 Countermeasures against the SIDH attacks

The recent series of attacks by Castryck and Decru [CD22], Maino and Martindale [MM22], and Robert [Rob22] exploits torsion-point information to break SIDH. These attacks trivially translate to the OPRF, where any third party can recover both the user's hashed input (which breaks obliviousness) and the server's secret key. In this section, we discuss how to adapt the existing SIDH countermeasures to work in the OPRF setting. After modifying the main exchange, we propose a novel proof of isogeny knowledge that works together with the countermeasures, which may be of independent interest since it is the first proof to prove the correctness of torsion point images in the SIDH-with-countermeasure setting. This proof can be used together with the patched SIDH to obtain a post-quantum non-interactive key-exchange.

Combining the countermeasures together with the novel proof of torsion point correctness, we obtain an SIDH-based OPRF that is resistant against the SIDH attacks. While the countermeasures impose larger parameters, the resulting protocol remains the most compact post-quantum vOPRF.

5.1 Protecting the exchange

The OPRF exchange is based on SIDH, but it has some differences from a simple SIDH key exchange. In particular, in the OPRF protocol the two parties compute isogenies of different lengths and need to prove the correctness of their outputs. Moreover, the server starts its computation from a curve provided by the user and also needs to prove that it used the same key it has previously committed.

The attacks on SIDH recover an isogeny $\phi : E \to E'$ of degree d when provided with the curves E, E', the degree d and the image of the n-torsion $\phi(E[n])$, where the size of n satisfies at least $n \approx d^{1/2}$. This suggests three possible countermeasures, as discussed in [FMP23]:

1. Hide the degree d of the isogeny ϕ by choosing a random $d' \mid d$.
2. Increase the degree d of the isogeny ϕ, such that $d \gg n^2$,
3. Mask the exact torsion images by providing a scalar multiple.

In the OPRF setting, the hidden-degree countermeasure does not appear to work. Both parties need to prove the correctness of their revealed torsion points, and all the proofs of isogeny knowledge in the literature rely on constructing an SIDH square and revealing some sides. This inevitably leaks the degree of the secret isogeny, which makes the hidden-degree countermeasure ill-suited to work with zero-knowledge proofs.

Relying only on longer isogeny may seem like a valuable approach, as it does not require on any new assumption. In the SIDH setting, it is not possible to protect both parties with such a strategy because it would require both isogenies to be longer than the other. In the OPRF, however, the user computes the message and blinding isogenies in the first round, provides enough torsion information for the server to compute its isogeny, and the server reveals the torsion information needed for the user to invert the blinding isogeny. This suggests that the protocol could be secure if the server's isogeny is sufficiently longer than the blinding isogeny, and if the composition of the message and blinding isogeny is sufficiently longer the server's isogeny. This approach could lead to a compact and fairly efficient protocol, but unfortunately it does not guarantee the input hiding property. The server should not distinguish the user's input even when the user chooses between two server-controlled messages. This approach is thus inadequate for an OPRF, but it might still be useful for specific applications where the message space has sufficiently-large entropy and such a strong security assumption is not needed.

Thus, to guarantee the security of the SIDH-based OPRF we need to rely on the masked-torsion countermeasure, as in masked SIDH (M-SIDH) [FMP23]. Let $\phi : E \to E'$ be the isogeny we want to protect, and let P, Q be a basis of $E[n]$, for some n coprime with d. Given a basis $P' = \phi(P), Q' = \phi(Q)$, the other party computes their isogeny with kernel $\langle P' + [x]Q' \rangle$, where x is the secret key. Thus, it is possible to reveal $[\alpha]P', [\alpha]Q'$, for some random α coprime with the torsion order n, without affecting the correctness of the protocol. However, an attacker can recover the value α^2 from the Weil pairing, since $e([\alpha]P', [\alpha]Q') = e(P, Q)^{\alpha^2 \deg \phi}$. To ensure that the attacker cannot recover the value α, we want that any value has at least 2^λ square roots modulo n, hence n needs to be the product of at least λ prime powers. This, however, is not enough to guarantee security, as an attacker can guess the correct square root modulo some $n' \mid n$ with $n' > d^{1/2}$ in less than $\mathcal{O}(2^\lambda)$ guesses. We thus also require that $d > n'$, where n' is the product of the powers of the λ largest primes dividing n. From now on, we write $n = f_{\mathsf{MSIDH}}(\lambda, d)$ to denote the smallest value n that can guarantee λ bits of security when used in M-SIDH with an isogeny of degree d. Lastly, the countermeasure analysis in [FMP23] shows that an attack is possible for certain parameters when the starting curve has a small endomorphism. In our case, such an attack does not apply even if the OPRF starting curve E_0 has a known endomorphism ring with a small endomorphism ι. The composition of the message and blinding isogeny $\phi_x \circ \phi_m$ is sufficiently long that the attack does not apply, while considering the blinding isogeny ϕ_x alone (remember that in the security game the attacker can control the messages) does not help either. Even if the attacker can guess the input message, the smallest endomorphism known on the domain of the blinding isogeny is $\hat{\phi}_m \circ \iota \circ \phi_m$, which is too large. The server computes its isogeny starting from a curve E_{mx} that is sent by the user, which generally could be an avenue for attack since MSIDH is insecure for special starting curves. However, the user also submits a proof that the user knows an isogeny of long degree between E_0 and E_{mx}. This guarantees that the smallest known endomorphism is again sufficiently large, and thus the attack does not apply to the server's isogeny as well.

We can now formulate the following problem, on whose hardness the input hiding property of the OPRF is based.

Problem 5 (Decisional M-SIDH isogeny problem). Let E_0 be a supersingular elliptic curve, with a basis P, Q be of $E_0[n]$. Distinguish between the following distributions:

- (E_1, R, S), where E_1 is the codomain of a d-isogeny $\phi : E_0 \to E_1$, where d is coprime with n, and the points R, S are the masked images of P, Q, i.e. $R = [\alpha]\phi(P)$ and $S = [\alpha]\phi(Q)$ for some $\alpha \xleftarrow{\$} \mathbb{Z}_n^*$;
- (E_1, R, S), where E_1 is a random supersingular elliptic curve and the points R, S are a random basis of $E_1[n]$ such that $e(R, S) = e(P, Q)^{\alpha^2 d}$, for some value α.

The hardness of the problem clearly depends on the choices of n and d; the problem (conjecturally) requires $O(2^\lambda)$ operations to solve when $n > f_{\mathsf{MSIDH}}(\lambda, d)$, i.e. the product of the λ largest prime powers dividing n is smaller than $\sqrt{d}$.

Concrete cost. We have shown it is possible to protect the OPRF protocol from the SIDH attacks. Unfortunately, the proposed countermeasure do come at a significant cost. The degrees of the blinding isogeny and the server's isogeny are the same as in SIDH with the same countermeasures. At security level $\lambda = 128$, that corresponds to isogenies of degree $\approx 2^{2956}$. More generally, we see experimentally that the degree of the isogenies scales log-linearly in the security parameter with a constant of ≈ 3.3. We thus have that the degree of the blinding isogeny and the server's isogeny must be $\approx 2^{3.3\lambda \log \lambda}$ to guarantee the security of the protocol.

5.2 Adapting the proof of isogeny knowledge

In the previous section, we showed how it is possible to protect the OPRF against the SIDH attacks using masked torsion points. However, in the OPRF protocol both parties need to prove the correctness of their torsion images to prevent adaptive attacks and guarantee the verifiability of the execution. This leads to an issue, because both the user and the server want to prove that their torsion points were honestly generated, but these points are also scaled by a secret value. Thus, two parties want to prove that both points were honestly generated and scaled by the same value.

In this section, we propose a zero-knowledge proof of isogeny knowledge that can guarantee the correctness of torsion points up to a scalar, i.e. a proof for the following relation:

$$\mathcal{R}_{\mathsf{iso}} = \left\{ ((E_0, P_0, Q_0, E_1, P_1, Q_1), (\phi, \alpha)) \;\middle|\; \begin{array}{c} \phi : E_0 \to E_1 \text{ is a cyclic } d\text{-isogeny}, \\ P_1 = [\alpha]\phi(P_0), \\ Q_1 = [\alpha]\phi(Q_0). \end{array} \right\}.$$

In the literature, we can find two proofs of isogeny knowledge that also guarantee the correctness of torsion point images. The first proof constructs an SIDH square and explicitly maps the torsion images through all the sides of the square. This proof was proposed by Boneh, Kogan, and Woo [BKW20] for the OPRF protocol, based on a previous idea by Galbraith [Gal18]. The second proof [DFDGZ22], instead, is an extension of the simpler proof of isogeny knowledge by De Feo and Jao [JD11]. The first proof requires a larger prime, but the torsion images are explicitly mapped, which makes it well-suited to support masked torsion. We thus propose a new proof based on the same approach as [BKW20] and [Gal18], although with some notable differences. Building a more compact proof based on the second approach [DFDGZ22] remains an open problem.

The main idea is that the masking constant α can be split into three shares $\alpha = \alpha_1\alpha_2\alpha_3$. The prover can mask the torsion points with α_i when computing the i-th side of the SIDH square, so that the composition of the three side isogenies, together with their masking values, forms a commutative diagram with the isogeny ϕ with masking value α. The proof remains zero-knowledge because each single value α_i is independent of α. More formally, let E_0 and E_1 be supersingular elliptic curves with points $P_0, Q_0 \in E_0[n]$ and $P_1, Q_1 \in E_0[n]$. The prover wants to prove knowledge of a d-isogeny $\phi : E_0 \to E_1$ and a value $\alpha \in \mathbb{Z}_n$ such that $P_1 = [\alpha]\phi(P_0)$ and $Q_1 = [\alpha]\phi(Q_0)$. This only makes sense if ϕ is secret, thus let us assume $n = f_{\mathsf{MSIDH}}(\lambda, d)$. The prover generates a random isogeny $\psi : E_0 \to E_2$ of degree s, where $s \approx n$ is a smooth number coprime with both n and d, and generates the SIDH square (E_0, E_1, E_2, E_3) with edges $(\phi, \psi, \phi', \psi')$. To guarantee soundness, the prover needs to show that ψ and ψ' are parallel: the prover thus generates a s-basis R_2, S_2 on E_2, maps it to E_3 to obtain R_3, S_3, and expresses the kernels of $\hat{\psi}$ and $\hat{\psi}'$ in terms of R_2, S_2 and R_3, S_3 with the same linear coefficients. The prover also splits α in three shares $\alpha = \alpha_1\alpha_2\alpha_3$ and maps the points P_0, Q_0 through ψ and ϕ' with masking values α_1 and α_2 to obtain the points

$$P_2 = [\alpha_1]\psi(P_0),\ Q_2 = [\alpha_1]\psi(Q_0),$$
$$P_3 = [\alpha_2]\phi'(P_2),\ Q_3 = [\alpha_2]\phi'(Q_2),$$

which implies that P_3 and Q_3 also satisfy the relation

$$[\alpha_3]P_3 = \psi'(P_1),\ [\alpha_3]Q_3 = \psi'(Q_1).$$

Hence, the SIDH square commutes with respect to the points P_i, Q_i, i.e. if we restrict ourselves to the n-torsion, we have

$$[\alpha][s]\phi = [\alpha_3]\hat{\psi}' \circ [\alpha_2]\phi' \circ [\alpha_1]\psi.$$

Thus, the witness can be split into three components, and hence we obtain a proof with ternary challenges. The prover initially commits to the curves E_2, E_3 and the relevant points on them with a commitment scheme

$\mathsf{C}(\cdot)$. Then, depending on the challenge, the prover responds with one edge of the SIDH square, the relevant curves and points, and the corresponding commitment openings. The proof is described in Fig. 4. Since each iteration has soundness error $2/3$, the proof must be repeated $-\lambda \log_{2/3}(2) \approx 1$ times to achieve a soundness error of $2^{-\lambda}$.

Remark 1. If the kernel of the isogeny ϕ is not defined over a small extension field, as in the case of the message isogeny, the proof can be computed by gluing together multiple SIDH squares, as shown in [BCC+22].

```
$\mathsf{P}_1((E_0, P_0, Q_0), (E_1, P_1, Q_1), \phi, \alpha)$:
  Sample a random cyclic isogeny $\psi : E_0 \to E_2$ of degree $s$;
  Construct the SIDH square $(E_0, E_1, E_2, E_3, \phi', \psi')$ on $(\phi, \psi)$;
  Sample random units $\alpha_1, \alpha_2$ mod $n$ and set $a_3 \coloneqq \alpha/\alpha_1\alpha_2$;
  Set $P_2, Q_2 \coloneqq [\alpha_1]\psi(P_1), [\alpha_1]\psi(Q_1)$, and $P_3, Q_3 \coloneqq [\alpha_2]\phi'(P_2), [\alpha_2]\phi'(Q_2)$;
  Let $R_2, S_2$ be a basis of $E_2[s]$ and set $R_3, S_3 \coloneqq \phi'(R_2), \phi'(R_3)$;
  Write $K = [a]R_2 + [b]S_2$ for $K$ a random generator of $\ker \hat{\psi}$
  Sample random strings $r_1, \dots, r_7$;
  return $\big(\mathsf{st}, \mathsf{C}(E_2, R_2, S_2, P_2, Q_2; r_1),\ \mathsf{C}(E_3, R_3, S_3, P_3, Q_3; r_2),$
         $\mathsf{C}(a, b; r_3),\ \mathsf{C}(\phi'; r_4),\ \mathsf{C}(\alpha_1; r_5),\ \mathsf{C}(\alpha_2; r_6),\ \mathsf{C}(\alpha_3; r_7)\big)$.
```

```
$\mathsf{P}_2(\mathsf{st}, \mathsf{chall})$:
  if $\mathsf{chall} == -1$ then
      return $((E_2, R_2, S_2, P_2, Q_2, r_1), (a, b, r_3), (\alpha_1, r_5))$;
  else if $\mathsf{chall} == 0$ then
      return $((E_2, R_2, S_2, P_2, Q_2, r_1), (E_3, R_3, S_3, P_3, Q_3, r_2), (\phi', r_4), (\alpha_2, r_6))$;
  else if $\mathsf{chall} == 1$ then
      return $((E_3, R_3, S_3, P_3, Q_3, r_2), (a, b, r_3), (\alpha_3, r_7))$;
```

```
$\mathsf{V}((E_0, P_0, Q_0), (E_1, P_1, Q_1), (\mathsf{com}_1, \dots, \mathsf{com}_9), \mathsf{chall}, \mathsf{resp})$:
  if $\mathsf{chall} == -1$ then
      $((E_2, R_2, S_2, P_2, Q_2, r_1), (a, b, r_3), (\alpha_1, r_5)) = \mathsf{resp}$;
      Check $\mathsf{com}_1 = \mathsf{C}(E_2, R_2, S_2, P_2, Q_2; r_1)$,
            $\mathsf{com}_3 = \mathsf{C}(a, b; r_3)$, $\mathsf{com}_5 = \mathsf{C}(\alpha_1; r_5)$;
      Let $\hat{\psi}$ be the isogeny with kernel $\langle [a]R_2 + [b]S_2 \rangle$;
      Check $\hat{\psi}$ is an $s$-isogeny from $E_2$ to $E_0$;
      Check $[\alpha_1 s]P_0 = \hat{\psi}(P_2)$ and $[\alpha_1 s]Q_0 = \hat{\psi}(Q_2)$;
  else if $\mathsf{chall} == 0$ then
      $((E_2, R_2, S_2, P_2, Q_2, r_1), (E_3, R_3, S_3, P_3, Q_3, r_2), (\phi', r_4), (\alpha_2, r_6)) = \mathsf{resp}$;
      Check $\mathsf{com}_1 = \mathsf{C}(E_2, R_2, S_2, P_2, Q_2; r_1)$,
            $\mathsf{com}_2 = \mathsf{C}(E_3, R_3, S_3, P_3, Q_3; r_2)$,
            $\mathsf{com}_4 = \mathsf{C}(\phi'; r_4)$, $\mathsf{com}_6 = \mathsf{C}(\alpha_2; r_6)$;
      Check $\phi'$ is a $d$-isogeny from $E_1$ to $E_2$;
      Check $R_3, S_3 = \phi'(R_2), \phi'(R_3)$;
      Check $P_3, Q_3 = [\alpha_2]\phi'(P_2), [\alpha_2]\phi'(Q_2)$;
  else if $\mathsf{chall} == 1$ then
      $((E_3, R_3, S_3, P_3, Q_3, r_2), (a, b, r_3), (\alpha_3, r_7)) = \mathsf{resp}$;
      Check $\mathsf{com}_2 = \mathsf{C}(E_3, R_3, S_3, P_3, Q_3; r_2)$,
            $\mathsf{com}_3 = \mathsf{C}(a, b; r_3)$, $\mathsf{com}_7 = \mathsf{C}(\alpha_3; r_7)$;
      Check $\langle R_3, S_3 \rangle = E_3[s]$;
      Let $\hat{\psi}'$ be the isogeny with kernel $\langle [a]R_3 + [b]S_3 \rangle$;
      Check $\hat{\psi}$ is an $s$-isogeny from $E_3$ to $E_1$;
      Check $[\alpha_3 s]P_1 = \hat{\psi}'(P_3)$ and $[\alpha_3 s]Q_1 = \hat{\psi}'(Q_3)$;
```

Fig. 4. Interactive proof of knowledge for the relation $\mathcal{R}_{\mathsf{iso}}$.

We now sketch the proofs of correctness, three-special soundness and zero-knowledge. Given the similarity of the zero-knowledge proof with those in [BKW20], the proofs also follow a similar approach.

- **Correctness.** A honest prover always generates proofs that are accepted by the verifier. The verifier recomputes the same operations as the prover and checks that the outputs match. The only difference is in the $\mathsf{chall} = \pm 1$ cases, where the verifier computes the dual of ψ and ψ', which then introduces the s factor in the point equality check.
- **Three-special soundness.** The protocol is three-special sound because there exists an extractor that extracts the witness given three accepting transcripts with the same commitments and different challenges. The isogeny ϕ can be computed by mapping the kernel of ϕ' (from $\mathsf{chall} = 0$) under the isogeny $\hat{\psi}$ (from $\mathsf{chall} = -1$). Since the isogenies ψ and ψ' are parallel (from all the challenges combined), this guarantees that ϕ is a d-isogeny from E_0 to E_1. The masking value α can be recomputed as the product of α_1, α_2, and α_3.
- **Zero-knowledge.** We sketch a simulator that given a statement $(E_0, P_0, Q_0, E_1, P_1, Q_1)$ and a challenge chall can simulate a valid transcript without knowledge of the witness. For the case $\mathsf{chall} = -1$, the simulator behaves like an honest prover. For $\mathsf{chall} = +1$, the situation is similar: the simulator can compute a d-isogeny ψ', pick a random basis R_3, S_3 of $E_3[d]$ and a random value $\alpha_3 \in \mathbb{Z}_n^*$, and compute the values a, b and points P_3, Q_3 that pass verification. Note that the points R_3, S_3 are uniformly random among the bases of $E_3[d]$, and the value α_3 is uniformly random and independent of α; the simulated values are thus distributed as the honestly-generated ones. The case of $\mathsf{chall} = 0$ is more complicated: the simulator can sample a random curve E_2, generate a random basis P_2, Q_2 of $E_2[n]$ that satisfies $e(P_2, Q_2) = e(P_0, Q_0)^{x^2 s}$ for some random x, pick a random d-isogeny $\phi' : E_2 \to E_3$, and compute the image points on E_3. In this case, the indistinguishability of the simulator's output is only computational. It is thus based on the conjectured hardness of the following problem, which is a modified version of the Decisional Supersingular Product (DSSP) problem introduced in [JD11].

Problem 6 (DSSP with Torsion (DSSPwT) problem). Given an isogeny $\phi : E_0 \to E_1$ of degree d and points P_0, Q_0 such that $\langle P_0, Q_0 \rangle = E_0[n]$, where $n = f_{\mathsf{MSIDH}}(\lambda, d)$, distinguish between the following distributions:

- $\mathcal{D}_0 = \{(E_2, P_2, Q_2, V)\}$, where E_2 is the codomain of an s-isogeny $\psi : E_0 \to E_2$, the points P_2, Q_2 satisfy $P_2 = [\alpha]\psi(P_0)$, $Q_2 = [\alpha]\psi(Q_0)$ for some $\alpha \in \mathbb{Z}_n^*$, and V is a generator of $\psi(\ker \phi)$.
- $\mathcal{D}_1 = \{(E_2, P_2, Q_2, V)\}$, where E_2 is a random supersingular curve with the same cardinality as E_0, P_2 and Q_2 are two random points on E_2 of order n such that $e(P_2, Q_2) = e(P_0, Q_0)^{\alpha^2} s$ for some $\alpha \in \mathbb{Z}_n^*$, and the point V is a random point on E_2 of order d.

Note that [BKW20] argues that a similar proof can only reveal one torsion point (either P_i or Q_i) at a time to prevent a distinguishing attack on the simulator. The attack they present relies on computing the Weil pairing between two points of coprime order, and thus their pairing is always one. The attack thus does not apply, and the simulated transcript remains undistinguishable under Weil pairing checks because the sampled points P_2, Q_2 are guaranteed to have the same pairing as the honestly-generated points. By revealing both points P_i and Q_i we obtain a significantly more efficient proof, since it has 1/3 soundness rather than 1/6.

Optimizations. For simplicity, the proof in Fig. 4 contains a schematic description of the protocol, but the proof can be made more efficient through a series of optimizations.

In the commitment phase, the value α_2 is only revealed together with the isogeny ϕ', and thus they can be committed together. Note that we have the prover commit to ϕ' to make the proof online-extractable without recursion, which is necessary to achieve a proof in the UC model. For applications of this proof outside of the OPRF context, the prover can avoid committing to ϕ'. The masking values α_1 and α_3 are independent of α, even when considered together, because α_2 is uniformly random. They can then be committed together and revealed both in the response to challenges $\mathsf{chall} = \pm 1$. Since the commitment for a, b is also revealed when $\mathsf{chall} = \pm 1$, the values a, b, α_1, α_3 can all be committed together. When $\mathsf{chall} = -1$, the curve E_3 and the points P_3, Q_3 are not revealed, and thus learning α_3 does not provide any information. The same applies to α_1 when $\mathsf{chall} = +1$. This allow us to reduce the number of commitments to four.

To further reduce the communication between prover and verifier, the basis R_2, S_2 on E_2 can be chosen canonically, so that it can be recomputed from E_2. Moreover, for the challenge $\mathsf{chall} = -1$, the prover can avoid revealing the curve E_2, the points P_2, Q_2 and the coefficients a, b by revealing instead a kernel generator of ψ. The prover can recompute E_2, P_2, Q_2 and obtain a, b by writing a kernel generator of $\hat{\psi}$ in terms of the canonical basis R_2, S_2. Normally, the recomputed a, b would not be the same as those computed by the verifier since they are not unique. The problem can be avoided by fixing a canonical way to compute the coefficients, such as prescribing that one of the two coefficients must be one, and that a must be one if both coefficients are invertible mod s. The same approach holds for $\mathsf{chall} = +1$, except that the points R_3, S_3 have to be revealed by the prover. In the case of the horizontal isogeny, the prover can avoid revealing E_3 and the points R_3, S_3 and P_3, Q_3. They can all be recomputed from the remaining values.

Concrete cost. Each repetition of the proof requires two commitments, which are 2λ-bit long and use a λ-bit long opening. When $\mathsf{chall} = -1$, the prover reveals one s-isogeny, a masking value, and two commitment openings, which requires $\log n + \log s + 2\lambda$ bits. When $\mathsf{chall} = +1$, the prover also reveals two torsion points of order s: if they are compressed as in [AJK+16,CLN16], the response requires $5 \log s + \log n + 2\lambda$ bits. Lastly, for $\mathsf{chall} = 0$, the prover reveals a curve, a d-isogeny, two points of order n, a masking value, and three openings; thus, the answer requires $2 \log p + \log d + 5 \log n + 3\lambda$ bits.

For server-side proofs, we can take $d \approx n \approx s \approx \sqrt[3]{p}$, which makes the largest response (when $\mathsf{chall} = 0$) to be $4 \log p + 3\lambda$ bit long. In the case of the proof run by the user, d is slightly longer to include the degree of the message isogeny; we thus have $d \approx \sqrt[3]{p} + 4\lambda$. In this case, the largest response is $4 \log p + 7\lambda$ bit long.

In the OPRF setting, we rely on the Unruh transformation [Unr14] to obtain a non-interactive zero-knowledge proof that is universally composable. The bandwidth of such a proof can be bounded from above by $t(|\mathsf{com}| + 3|\mathsf{rsp}|)$, where t is the number of repetitions, the factor 3 comes from the size of the challenge space, and $|\mathsf{com}|$ and $|\mathsf{rsp}|$ represent the size of the commitments and the longest response (in our case, when $\mathsf{chall} = 0$). Since the soundness error of the sigma protocol is 2/3, the protocol needs to be repeated $t = \lambda / \log(3/2)$ times. This gives a total size of $1.7\lambda(12 \log p + 25\lambda)$ bits for server-side proofs, while user-side proofs require $1.7\lambda(12 \log p + 37\lambda)$ bits.

For other applications where security in the UC framework may not be required, the more efficient Fiat-Shamir transform [FS87] is sufficient to obtain a NIZKP. In that scenario, we estimate an average proof where the three challenges appear equally to require $\approx 1.7\lambda(20/9 \log p + 7/3\lambda)$ bits, while a worst-case proof, with only $\mathsf{chall} = 0$ challenges, to require $\approx 1.7\lambda(4 \log p + 7\lambda)$ bits.

6 Verifiability

Oblivious PRFs can satisfy a stronger security property called *verifiability*. Informally, this guarantees that the server behaves honestly and always uses the same long-term static key. This is needed to guarantee the privacy of the user in those instances where the user may later reveal the output of the OPRF. A malicious server may behave “honestly” while also using different secret keys on different interactions. After learning the OPRF output of the user, the server can then test which secret key was used to produce that specific output and thus link the user to a specific user-server interaction.

The construction by Boneh, Kogan, and Woo. The OPRF protocol by Boneh, Kogan, and Woo achieves verifiability by introducing three components. First, the server initially commits to a secret key k. The commitment is in the form of an elliptic curve $E_C \coloneqq E/\langle P + [k]Q \rangle$, where the curve E and the points P, Q are fixed parameters. Second, during the OPRF execution, the server provides a zero-knowledge proof that its computations used the same key as the one in the commitment. We refer to this proof as a *proof of parallel isogeny* (PoPI). Lastly, the server also provides two *proofs of isogeny knowledge* (PoIKs) that guarantee the correctness of the computations during both the commitment stage and the OPRF execution.

The proof of parallel isogeny proposed by Boneh, Kogan, and Woo relies on the user and the server engaging in an SIDH exchange, where one of the sides is either the commitment isogeny or the the secret server isogeny in the OPRF protocol. The user only reveals the codomain of an isogeny starting from either starting curve, which means the server cannot distinguish the two cases. Thus, if the keys used in the protocol and the commitment were different, the server could not do better than randomly guessing which starting curve the user used. By repeating the protocol λ times, the server can prove the parallelness of the isogenies with soundness $2^{-\lambda}$. However, this proof is inherently interactive. Since the server also needs to defend against adaptive attacks [GPST16], the proof uses an approach similar to the Fujisaki-Okamato transform, which requires five rounds of interaction. Moreover, the proof relies on multiple SIDH exchanges, and it is thus broken by the attacks on SIDH [CD22,MM22,Rob22]. It may be possible to avoid some of the issues, for instance by starting from a curve of unknown endomorphism ring using a trusted setup and by switching to an SIDH version that is resistant to the recent attacks. However, it seems impossible to obtain a non-interactive proof using a similar approach.

Our proposal. We introduce a novel public-coin proof protocol of parallel isogeny that sidesteps the problems discussed above. Since the proof does not rely on private randomness, we obtain a proof *of knowledge* that can be made non-interactive via the Fiat-Shamir transform [FS87] or the Unruh transform [Unr15]. In the OPRF setting, we will rely on the latter to achieve the online-extractability without rewinding needed to get a proof in the UC model. Our main approach relies on executing two proofs of isogeny knowledge in parallel *with correlated randomness*. Since part of the randomness used is shared, we can obtain a proof of parallelness without needing additional computations.

Firstly, we formalize the notion of parallelness. We say that two d-isogenies $\phi : E_0 \to E_1$ and $\tilde{\phi} : \tilde{E}_0 \to \tilde{E}_1$ are parallel with respect to the bases $R, S \in E_0[d]$ and $\tilde{R}, \tilde{S} \in E_0'[d]$ if there exists coefficients $a, b \in \mathbb{Z}_d$ such that $\ker \phi = \langle [a]R + [b]S \rangle$ and $\ker \tilde{\phi} = \langle [a]\tilde{R} + [b]\tilde{S} \rangle$. This suggests that the parallelness relation that we are proving is the following:

$$\mathcal{R}_{\mathsf{par}} = \left\{ ((E_0, R, S, E_1, \tilde{E}_0, \tilde{R}, \tilde{S}, \tilde{E}_1), k_0, k_1) \;\middle|\; \begin{array}{c} E_0/\langle [k_0]R + [k_1]S \rangle \cong E_1, \\ \tilde{E}_0/\langle [k_0]\tilde{R} + [k_1]\tilde{S} \rangle \cong \tilde{E}_1 \end{array} \right\}.$$

However, as discussed before, we are combining several proofs together to obtain a larger proof that simultaneously proves knowledge of two isogenies and guarantees the two isogenies are parallel. We thus obtain a proof for the following relation, where we consider the case of a secret key with two coefficients for completeness. For practical reasons, the OPRF will fix $k_0 = 1$ without any loss of security.

$$\mathcal{R}^*_{\mathsf{par}} = \left\{ \begin{array}{c} ((E_0, R, S, P_0, Q_0, E_1, P_1, Q_1, \\ \tilde{E}_0, \tilde{R}, \tilde{S}, \tilde{P}_0, \tilde{Q}_0, \tilde{E}_1, \tilde{P}_1, \tilde{Q}_1), \\ (k_0, k_1, \alpha, \alpha')) \end{array} \;\middle|\; \begin{array}{c} \ker \phi = \langle [k_0]R + [k_1]S \rangle, \\ \ker \phi' = \langle [k_0]\tilde{R} + [k_1]\tilde{S} \rangle, \\ (E_0, P_0, Q_0, E_1, P_1, Q_1), (\phi, \alpha) \in \mathcal{R}_{\mathsf{iso}}, \\ (\tilde{E}_0, \tilde{P}_0, \tilde{Q}_0, \tilde{E}_1, \tilde{P}_1, \tilde{Q}_1), (\phi', \alpha') \in \mathcal{R}_{\mathsf{iso}} \end{array} \right\}.$$

Now, let the curve $\tilde{E}_0$ with a d-basis $\tilde{R}, \tilde{S}$ be fixed protocol parameters. Using the same notation as before, assume that server has committed to its key (k_0, k_1) by publishing the codomain of the d-isogeny $\tilde{\phi}$ that has kernel $\langle [k_0]\tilde{R} + [k_1]\tilde{S} \rangle$. The server may also reveal some torsion information in its commitment, but as we will discuss later, this is not strictly needed. During the OPRF execution, the server receives a curve E_0 with a d-basis R, S on it, and it computes $\phi : E_0 \to E_1 \coloneqq E_0/\langle [k_0]R + [k_1]S \rangle$. The server then wants to prove that it knows the isogenies ϕ and $\tilde{\phi}$ and that they are parallel.

If the server simply ran two instances of the PoIK from Fig. 4 in parallel, there would be no way to convince the prover that the isogenies are indeed parallel. If the proofs share the same challenges, i.e. the verifier sends the same challenges to both proofs, the server would respond with both ϕ and $\tilde{\phi}'$ when $\mathsf{chall} = 0$. However, the isogenies ϕ and $\tilde{\phi}'$ are not parallel with respect to the bases R_2, S_2 and $\tilde{R}_2, \tilde{S}_2$ since they are randomly generated. We thus to want to modify the proof such that the bases R_2, S_2 and $\tilde{R}_2, \tilde{S}_2$ are related to R_0, S_0 and $\tilde{R}_0, \tilde{S}_0$, so that when ϕ and $\tilde{\phi}$ are parallel, so are ϕ' and $\tilde{\phi}'$. One way to do this is by computing the basis R_2, S_2 as $R_2, S_2 = \psi(R_0), \psi(S_0)$ (and similarly for $\tilde{R}_2, \tilde{S}_2$) in both proofs, where ψ is the vertical isogeny used in the proof of knowledge. This is however not zero-knowledge, because when $\mathsf{chall} = 0$, the verifier could recompute the secret isogeny ϕ. Instead, we propose that the prover generates four random coefficients $w, x, y, z \in \mathbb{Z}_d$ such that $wz - xy \neq 0 \bmod d$, and computes R_2 and S_2 as the solution of

$$R_0 = [w]\psi(R_2) + [x]\psi(S_2), \quad S_0 = [y]\psi(R_2) + [z]\psi(S_2).$$

This is then secure, because the basis R_2, S_2 is uniformly random. Thus, for a single proof, this change only affects how the random points R_2, S_2 are generated, but does not affect the security of the proof. The rest of the proof needs to be modified to ensure that the process is followed correctly, i.e. we want the prover to reveal the values w, x, y, z together with ψ so that the verifier can verify the correctness of R_2 and S_2. The modified proof is denoted by $\mathcal{P}^*_{\mathsf{iso}}$, and it is represented explicitly in Fig. 5.

```
$\mathsf{P}^*_1\Big(((E_0, R_0, S_0, P_0, Q_0), (E_1, P_1, Q_1), \phi, \alpha), (w, x, y, z)\Big)$:
1-4: Same as $\mathsf{P}_1$ in Fig. 4.
 5:  Set $R_2 := [w]\psi(R_0) + [x]\psi(S_0), S_2 := [y]\psi(R_0) + [z]\psi(S_0)$;
6-8: Same as $\mathsf{P}_1$ in Fig. 4.

$\mathsf{P}^*_2(\mathsf{st}, \mathsf{chall})$:
 1: if $\mathsf{chall} == -1$ then
 2:     return $((E_2, R_2, S_2, P_2, Q_2, r_1), (a, b, r_3), (\alpha_1, r_5), (w, x, y, z))$;
 3: else
 4:     Same as $\mathsf{P}_2$ in Fig. 4.

$\mathsf{V}^*\Big((((E_0, R_0, S_0, P_0, Q_0), (E_1, P_1, Q_1), \phi, \alpha), \mathsf{com}, \mathsf{chall}, \mathsf{resp}\Big)$:
 1: if $\mathsf{chall} == -1$ then
 2:     $((E_2, R_2, S_2, P_2, Q_2, r_1), (a, b, r_3), (\alpha_1, r_5), (w, x, y, z)) = \mathsf{resp}$;
3-6:    Same as $\mathsf{V}$ in Fig. 4.
 7:     Check $R_2 := [w]\psi(R_0) + [x]\psi(S_0), S_2 := [y]\psi(R_0) + [z]\psi(S_0)$;
 8: else
 9:     Same as $\mathsf{V}$ in Fig. 4.
```

Fig. 5. Modified proof of knowledge for the relation $\mathcal{R}_{\mathsf{iso}}$ where the basis randomness is explicit. The expressions in magenta denote the changes from Fig. 4.

Now, if the prover executes the modified proof of isogeny knowledge for ϕ and $\tilde{\phi}$ in parallel, with the same challenges, and with the same values x, w, y, z, the isogenies $\phi', \tilde{\phi}'$ revealed when $\mathsf{chall} = 0$ are parallel when the isogenies $\phi, \tilde{\phi}$ are also parallel, as shown in the following lemma.

Lemma 2. *Let notation be as above. The isogenies $\phi, \tilde{\phi}$ are parallel if and only if the isogenies $\phi', \tilde{\phi}'$ are also parallel.*

Proof. Assume the isogeny ϕ has kernel $\langle [k_0]R_0 + [k_1]S_0 \rangle$ and the isogeny $\tilde{\phi}$ has kernel $\langle [\tilde{k}_0]\tilde{R}_0 + [\tilde{k}_1]\tilde{S}_0 \rangle$. The kernel of ϕ' is the image of the kernel of ϕ under ψ, i.e. $\ker \phi' = \psi(\ker \phi)$. Since $\ker \phi = \langle [k_0]R_0 + [k_1]S_0 \rangle$, it follows that

$$\ker \phi' = \langle [k_0]\psi(R_0) + [k_1]\psi(S_0) \rangle = \langle [wk_0 + yk_1]R_2 + [xk_0 + zk_1]S_2 \rangle.$$

Similarly, we obtain

$$\ker \tilde{\phi}' = \langle [w\tilde{k}_0 + y\tilde{k}_1]\tilde{R}_2 + [x\tilde{k}_0 + z\tilde{k}_1]\tilde{S}_2 \rangle.$$

If the isogenies $\phi, \tilde{\phi}$ are parallel, then $k_0 = \tilde{k}_0$ and $k_1 = \tilde{k}_1$ for some choice of $k_0, k_1, \tilde{k}_0, \tilde{k}_1$. Similarly, if the isogenies $\phi', \tilde{\phi}'$ are parallel, then $wk_0 + yk_1 = w\tilde{k}_0 + y\tilde{k}_1$ and $xk_0 + zk_1 = x\tilde{k}_0 + z\tilde{k}_1$, for the same $k_0, k_1, \tilde{k}_0, \tilde{k}_1$.

Since the coefficients w, x, y, z were chosen such that $wz - xy \neq 0 \bmod d$, they form an invertible matrix, which implies the two statements are equivalent, i.e.

$$\begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} \tilde{k}_0 \\ \tilde{k}_1 \end{bmatrix} \iff \begin{bmatrix} w & x \\ y & z \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} w & x \\ y & z \end{bmatrix} \begin{bmatrix} \tilde{k}_0 \\ \tilde{k}_1 \end{bmatrix}.$$

□

```
$\mathsf{P}_1((E_0, R, S, P_0, Q_0, E_1, P_1, Q_1), (\tilde{E}_0, \tilde{R}, \tilde{S}, \tilde{P}_0, \tilde{Q}_0, \tilde{E}_1, \tilde{P}_1, \tilde{Q}_1), k_0, k_1, \alpha, \alpha')$:
  Sample random coefficients $w, x, y, z$ such that $wz - xy \neq 0 \bmod d$;
  Compute $\phi : E_0 \to E_0/\langle [k_0]R + [k_1]S \rangle \cong E_1$
  Compute $\phi' : \tilde{E}_0 \to \tilde{E}_0/\langle [k_0]\tilde{R} + [k_1]\tilde{S} \rangle \cong \tilde{E}_1$
  Run $\mathsf{P}_1^*((E_0, P_0, Q_0, E_1, P_1, Q_1), \phi, \alpha, (w, x, y, z))$ to get $\mathsf{st}, \mathsf{com}$;
  Run $\mathsf{P}_1^*((\tilde{E}_0, \tilde{P}_0, \tilde{Q}_0, \tilde{E}_1, \tilde{P}_1, \tilde{Q}_1), \tilde{\phi}', \tilde{\alpha}', (w, x, y, z))$ to get $\tilde{\mathsf{st}}, \widetilde{\mathsf{com}}$;
  return $\big((\mathsf{st}, \tilde{\mathsf{st}}), (\mathsf{com}, \widetilde{\mathsf{com}})\big)$;

$\mathsf{P}_2((\mathsf{st}, \tilde{\mathsf{st}}), \mathsf{chall})$:
  return $(\mathsf{P}_2^*(\mathsf{st}, \mathsf{chall}), \mathsf{P}_2^*(\tilde{\mathsf{st}}, \mathsf{chall}))$;

$\mathsf{V}((E_0, R, S, P_0, Q_0, E_1, P_1, Q_1), (\tilde{E}_0, \tilde{R}, \tilde{S}, \tilde{P}_0, \tilde{Q}_0, \tilde{E}_1, \tilde{P}_1, \tilde{Q}_1),$
    $(\mathsf{com}, \widetilde{\mathsf{com}}), \mathsf{chall}, (\mathsf{resp}, \widetilde{\mathsf{resp}}))$:
  Set $v := \mathsf{V}^*((E_0, R, S, P_0, Q_0, E_1, P_1, Q_1), \mathsf{com}, \mathsf{chall}, \mathsf{resp})$;
  Set $\tilde{v} := \mathsf{V}^*((\tilde{E}_0, \tilde{R}, \tilde{S}, \tilde{P}_0, \tilde{Q}_0, \tilde{E}_1, \tilde{P}_1, \tilde{Q}_1), \widetilde{\mathsf{com}}, \mathsf{chall}, \widetilde{\mathsf{resp}})$;
  return $v \wedge \tilde{v}$;
```

Fig. 6. Interactive proof of knowledge for the relation $\mathcal{R}^*_{\mathsf{par}}$.

We can now use the proof $\mathcal{P}^*_{\mathsf{iso}}$ from Fig. 5 to construct our proof of parallel isogeny knowledge. The prover runs two such proofs in parallel, with the same randomness (w, x, y, z), and responds to the verifier's challenges with the responses of the individual proofs. The resulting proof is represented explicitly in Fig. 6. The security proofs follow closely those of the PoIK $\mathcal{P}_{\mathsf{iso}}$ in Section 5.2: correctness of $\mathcal{P}_{\mathsf{iso}}$ implies correctness of $\mathcal{P}_{\mathsf{par}}$, while the soundness of $\mathcal{P}_{\mathsf{par}}$ follows from the soundness of $\mathcal{P}_{\mathsf{iso}}$ and Lemma 2. The argument for zero-knowledge is also similar, but it is based on the hardness of the following problem, which takes into consideration that the two parallel instance partially share the same randomness.

Problem 7 (Double DSSP with Torsion (DDSSPwT) problem). Let $\mathcal{D}_0$ and $\mathcal{D}_1$ be as in Problem 6. Given:

1. two d-isogenies $\phi : E_0 \to E_1$, $\tilde{\phi} : \tilde{E}_0 \to \tilde{E}_1$,
2. the points $R_0, S_0 \in E_0[d]$ and $\tilde{R}_0, \tilde{S}_0 \in \tilde{E}_0[d]$,
3. the points $P_0, Q_0 \in E_0[n]$ and $\tilde{P}_0, \tilde{Q}_0 \in \tilde{E}_0[n]$, where $n = f_{\mathsf{MSIDH}}(\lambda, d)$,

distinguish between the following distributions:

- $\mathcal{D}_0^* = \left\{ \begin{matrix} (E_2, R_2, S_2, P_2, Q_2, V), \\ (\tilde{E}_2, \tilde{R}_2, \tilde{S}_2, \tilde{P}_2, \tilde{Q}_2, \tilde{V}) \end{matrix} \right\}$, where the curves and the n-torsion points follow the $\mathcal{D}_0$-distribution, i.e. we have that $(E_2, P_2, Q_2, V) \leftarrow \mathcal{D}_0$, and $(\tilde{E}_2, \tilde{P}_2, \tilde{Q}_2, \tilde{V}) \leftarrow \mathcal{D}_0$, and moreover

  $$\begin{bmatrix} R_2 \\ S_2 \end{bmatrix} = B \begin{bmatrix} \psi(R_0) \\ \psi(S_0) \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \tilde{R}_2 \\ \tilde{S}_2 \end{bmatrix} = B \begin{bmatrix} \tilde{\psi}(\tilde{R}_0) \\ \tilde{\psi}(\tilde{S}_0) \end{bmatrix},$$

  for some $B \in \mathrm{GL}_2(\mathbb{Z}_n)$, and ψ and $\tilde{\psi}$ being respectively the s-isogenies between E_0 and E_2 and $\tilde{E}_0$ and $\tilde{E}_2$ that are guaranteed to exist because of the $\mathcal{D}_0$ distribution;

- $\mathcal{D}_1^* = \left\{ \begin{matrix} (E_2, R_2, S_2, P_2, Q_2, V), \\ (\tilde{E}_2, \tilde{R}_2, \tilde{S}_2, \tilde{P}_2, \tilde{Q}_2, \tilde{V}) \end{matrix} \right\}$, where the curves and the n-torsion points follow the $\mathcal{D}_1$-distribution, i.e. we have that $(E_2, P_2, Q_2, V) \leftarrow \mathcal{D}_1$, and $(\tilde{E}_2, \tilde{P}_2, \tilde{Q}_2, \tilde{V}) \leftarrow \mathcal{D}_1$, and moreover the points R_2, S_2 and $\tilde{R}_2, \tilde{S}_2$ form a random basis of $E_2[d]$ and $\tilde{E}_2[d]$, respectively.

The proof $\mathcal{P}_{\mathsf{par}}$ is a proof of knowledge, and it can be made non-interactive with standards transformations, such as the Fiat-Shamir [FS87] or the Unruh [Unr15] transform. This is the first non-interactive proof of parallelness.

Optimizations. For simplicity, the presentation of the proof $\mathcal{R}^*_{\mathsf{par}}$ preferred a schematic description, but it is possible to improve the protocol to make it more compact. Besides the optimizations applicable to the proof $\mathcal{P}_{\mathsf{iso}}$ described in Section 5.2, we remark that parallelness is independent of torsion images. Thus, the proofs of isogeny knowledge do not need to guarantee the correctness of torsion images to prove parallelness. However, in the OPRF context, the correctness of the torsion images revealed by the server is needed to guarantee verifiability: a malicious server might otherwise reveal incorrect torsion points to different users and use that information to match OPRF outputs to specific interactions. Hence, the proof can be made more efficient by avoiding proving the correctness of torsion images for the commitment isogeny.

Concrete cost. The proof described in Fig. 5 adds the communication of the values w, x, y, z when $\mathsf{chall} = -1$. In that case, the prover's response requires $\log n + \log s + 4 \log d + 2\lambda$ bits, while the answer to when $\mathsf{chall} = 1$ remains unchanged. In the case of $\mathsf{chall} = 0$, the response is also larger because the points R_2, S_2 need to be communicated explicitly. However, the $\mathcal{R}_{\mathsf{iso}}$ proof for the committment isogeny does not need to include torsion point information. Hence, the $\mathcal{R}^*_{\mathsf{par}}$ response to $\mathsf{chall} = 0$ requires $4 \log p + 2 \log d + 5 \log n + 4 \log s + 6\lambda$. Setting $d \approx n \approx s \approx \sqrt[3]{p}$, we obtain the size of the response to $\mathsf{chall} = 0$ is $|\mathsf{resp}_0| = 23/3 \log p + 6\lambda$.

We rely on the Unruh transform [Unr14] to obtain a universally composable NIZKP, which has proof size of $1.7\lambda(23 \log p + 26\lambda)$ bits. The same sigma protocol, made non-interactive with the Fiat-Shamir transform [FS87], would require $\approx 1.7\lambda(49/9 \log p + 26/3\lambda)$ bits for an average proof, while a worst-case proof would require $\approx 1.7\lambda(9 \log p + 10\lambda)$ bits.

7 A new OPRF protocol

In this section, we combine the countermeasures presented in Section 4, the SIDH countermeasures and the novel proof of isogeny knowledge discussed in Section 5, and the non-interactive proof of parallel isogeny introduced in Section 6 to obtain a verifiable OPRF protocol that is post-quantum secure, round-optimal, and moderately compact.

The OPRF protocol is a two-party protocol between a user U and a server S. Let N_{M}, N_{B}, N_{K} be coprime numbers representing the degrees of the message isogeny, the blinding isogeny, and the server's isogeny, respectively. Let p be a prime of the form $p = N_{\mathsf{M}} N_{\mathsf{B}} N_{\mathsf{K}} f - 1$, for some cofactor f, and let $E_0, \tilde{E}$ be two supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Moreover, let P, Q be a fixed basis of $E_0[N_{\mathsf{M}}]$ and let $\tilde{P}, \tilde{Q}$ be a fixed basis of $\tilde{E}[N_{\mathsf{K}}]$. The first curve is used to compute the PRF, while the second is used within the server's commitment.

At a high-level, to evaluate the OPRF on an input x, the user maps the input to a curve E_m according to Algorithm 1 and computes a blinding isogeny $\phi_b : E_m \to E_{mb}$. The user then sends the codomain curve, together with torsion images and a proof of their correctness, to the server, which computes a second isogeny $\phi_k : E_{mb} \to E_{mbk}$. The torsion information is appropriately masked to avoid the SIDH attacks. The server then responds with the curve E_{mbk}, some torsion information, a proof of their correctness, and a proof that it has used the previously-committed secret key. The user then concludes by using the torsion information provided by the server to undo the blinding isogeny and compute the curve E_{mk}. Its j-invariant is then hashed together with the input and the server's public key to form the PRF output. The protocol is described in Fig. 7 and it realized the OPRF ideal functionality of Fig. 2, which allows us to state the following theorem.

Theorem 1. *The protocol described in Fig. 7 realizes the ideal functionality $\mathcal{F}_{\mathsf{vOPRF}}$ of Fig. 2 in the random oracle model.*

The proof follows the same line as the security proof of the OPRF protocol by Boneh, Kogan, and Woo [BKW20, Theorem 20], since the hardness assumption of Problem 4 and the proof $\mathcal{P}_{\mathsf{iso}}$ are a drop-in replacement for the Auxiliary One-More SIDH assumption and the NIZKPK proof used in [BKW20], respectively. At a high level, the case of an honest user and malicious server in the proof is simple because the server only interacts with the user through their first query, and in that case the user's security corresponds to the input hiding property, guaranteed by the hardness of Problem 1. The case of a malicious user is more complicated, because the user has output. The server can be simulated as a honest server, but to ensure that the malicious user output is indistinguishable from the ideal-world, the random oracle $\bar{H}$ can be programmed to output the ideal-world output. This would create a problem with the ticketing system of the ideal functionality if the adversary could produce more OPRF outputs than the number of interactions, but the one-more unpredictability property prevents that. The main difference between this proof and that of [BKW20] is the use of a non-interactive proof of parallel isogeny that can be simulated in the proof, which results in a simpler proof since the proof of knowledge can be simulated. Note that the proof in [BKW20] is written in terms of the augmentable commitment abstraction, which we preferred avoiding; since the same security properties can be directly expressed in terms of the OPRF protocol, as shown in Section 3, the difference is purely syntactical.

Parameter selection. Firstly, we discuss how to select the starting curves E_0 and $\tilde{E}$. As mentioned in Section 5, the cryptanalysis on masked-torsion SIDH with a starting curve with small endomorphism [FMP23, Section 4.2] does not apply here, since the message isogeny removes this property from the starting curve of the blinding isogeny. Hence, the curve E_0 does not need to have unknown endomorphism ring. However, the situation is different for $\tilde{E}$: as observed in [BKM$^+$21], knowledge of $\operatorname{End}\tilde{E}$ allows to find collisions in the server's commitment. Thus, knowing $\operatorname{End}\tilde{E}$ would allow the server to break verifiability, since it could prove parallelness to two distinct isogenies. It is thus necessary that the curve $\tilde{E}$ is generated by a trusted party or through a multiparty trusted setup ceremony, such as the one presented in [BCC$^+$22].

The main parameter of the OPRF protocol is the prime p. Firstly, if the message isogeny is the composition of many isogenies whose kernel is defined over $\mathbb{F}_{p^4}$, the value $p+1$ does not need have a dedicated factor. Then, for the main exchange, i.e. the blinding, server's isogeny, unblinding part, we need to smooth coprime integers N_{B} and N_{K} that are highly composite to prevent the SIDH attacks. Following the analysis of Section 5, we have $N_{\mathsf{B}} \approx N_{\mathsf{K}} \approx 2^{3.3\lambda \log \lambda}$. Lastly, the proofs of knowledge $\mathcal{P}_{\mathsf{iso}}$ and $\mathcal{P}_{\mathsf{par}}$ require a third cofactor N_{S} that is coprime with both N_{B} and N_{K}. To guarantee the hardness of Problems 6 and 7, the integer N_{S} needs to be of the same length as N_{B} and N_{K}. However, since torsion points of order N_{S} do not need to be masked, the value N_{S} can be a prime power. Putting this together, we obtain that the prime p needs to be of the form $p = N_{\mathsf{B}} N_{\mathsf{S}} N_{\mathsf{K}} f - 1$ and at least $10\lambda \log \lambda$ bit long. For $\lambda = 128$, this corresponds to 8868-bit long prime. While it does not affect the prime size, the degree of the message isogeny ϕ_m also needs to be selected. Following the analysis in Section 4, we set $\deg \phi_m \approx 2^{4\lambda}$.

Note that the new computation of the message isogeny and the new proofs of knowledge has significantly reduced the size of the prime; compared to the OPRF protocol by Boneh, Kogan, and Woo, we use a prime that is 5.8× larger, while relying on an SIDH protocol with isogenies that are 9.2× longer.

Efficiency. We now estimate the communication cost of the OPRF protocol. The largest components are the non-interactive proofs of knowledge: given the analysis of the previous sections, they are less than $1.7\lambda(35 \log p + 51\lambda)$-bit long. Since $\log p \approx 10\lambda \log \lambda$, we obtain that one OPRF execution requires $1.7\lambda^2(350 \log \lambda + 51)$ bits of communication. For $\lambda = 128$, this corresponds to a transcript of 8.7 MB.

We remark that the size of the proofs is particularly large due to the Unruh transform needed to prove security in the UC framework. If the proofs were made non-interactive via the Fiat-Shamir transform, a

[3] The proof algorithm does not receive torsion points because, as discussed in Section 6, they are not necessary to prove parallelness.

Parameters. A prime p of the form $p = N_{\mathsf{M}}N_{\mathsf{B}}N_{\mathsf{K}}f - 1$, where $N_{\mathsf{M}}, N_{\mathsf{B}}, N_{\mathsf{K}}$ are smooth coprime integers and f a smooth cofactor. E_0 and $\tilde{E}$ are supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, where $\operatorname{End}\tilde{E}$ is unknown, and $P, Q \in E_0[N_{\mathsf{M}}]$ and $\tilde{P}, \tilde{Q} \in E[N_{\mathsf{K}}]$ are fixed bases.
The protocol also relies on several functions:

- $H_i : \{0,1\}^* \to \mathbb{Z}_M$ for $i \in \{1, \ldots, I\}$, where I is such that $N_{\mathsf{M}}^I > 2^{4\lambda}$, to use within $\mathcal{H}_I$,
- $\bar{H} : \{0,1\}^* \to \{0,1\}^\lambda$, to hash the final PRF output,

and two non-interactive proofs of knowledge:

- $\mathcal{P}_{\mathsf{iso}}$, for the user to prove correctness of torsion images,
- $\mathcal{P}_{\mathsf{par}}$, for the server to prove it computed honestly with the committed key.

Initialization. On input INIT from the environment, the server S:

- sample $k \leftarrow \mathbb{Z}_K$ and stores it,
- computes the curve $\tilde{E}_C = \tilde{E}/\langle \tilde{P} + [k]\tilde{Q} \rangle$,
- stores $\mathsf{pk} = (j(E_C))$ and outputs $(\text{INIT}, \mathsf{pk})$.

Evaluation. On input INIT from the environment, the server S:

- On input (EVAL, S, x), the user U proceeds as follows:
    1. Sample $\alpha \leftarrow \mathbb{Z}_N^*$ and $b \leftarrow \mathbb{Z}_B$,
    2. Compute $(\phi_m, E_m) = \mathcal{H}_I(x)$;
    3. Compute $\phi_b : E_m \to E_{mb} \coloneqq E_m/\langle P_m + [b]Q_m \rangle$, where $P_m, Q_m = \mathcal{B}_B(E_m)$,
    4. Set $\phi_{mb} = \phi_b \circ \phi_1 \circ \phi_0$, $R = [\alpha]\phi_{mb}(P)$, $S = [\alpha]\phi_{mb}(Q)$,
    5. Compute $\pi_c \leftarrow \mathcal{P}_{\mathsf{iso}}(E_0, P, Q, E_{mb}, R, S, \phi_{mb}, \alpha)$,
    6. Send message (E_{mb}, R, S, π_c) to the server and store ϕ_b
- On input SERVERCOMPLETE from the environment and message (E_{mb}, R, S, π_c) from the user U, the server S proceeds as follows:
    1. Verify the proof π_c,
    2. Sample $\alpha_k \leftarrow \mathbb{Z}_n^*$,
    3. Compute $\phi_k : E_{mb} \to E_{mbk} \coloneqq E_{mb}/\langle R + [k]S \rangle$,
    4. Compute $R_k = [\alpha_k]\phi_k(P_b)$, $S_k = [\alpha_k]\phi_k(Q_b)$, where $P_b, Q_b = \mathcal{B}_B(E_{mb})$,
    5. Compute $\pi_k \leftarrow \mathcal{P}_{\mathsf{par}}((E_{mb}, P_b, Q_b, E_{mbk}, R_k, S_k), (\tilde{E}, \tilde{P}, \tilde{Q}, \tilde{E}_C), k, \alpha_k)$[3],
    6. Send $(\mathsf{pk}, E_{mbk}, R_k, S_k, \pi_k)$ to the user U
- On input $(\mathsf{pk} = j(E_c), E_{mbk}, R_k, S_k, \pi_k)$ from the server S, the user U proceeds as follows:
    1. Verify the proof π_k,
    2. Compute b_0, b_1 such that $\langle [b_0]P_b + [b_1]Q_b \rangle = \ker \hat{\phi}_b$, where $P_b, Q_b = \mathcal{B}_d(E_{mb})$,
    3. Compute $\phi_u : E_{mbk} \to E_{mk} \coloneqq E_{mbk}/\langle [b_0]R_k + [b_1]S_k \rangle$,
    4. Compute $y = \bar{H}(x, \mathsf{pk}, j(E_m k))$ and output $(\text{EVAL}, \mathsf{pk}, y)$.

Fig. 7. The verifiable OPRF protocol.

single execution of the verifiable OPRF with $\lambda = 128$ would require 1.9 MB of communication on average and 3.8 MB in the worst case. Such an OPRF may be used in instances where security in the UC framework is not necessary. Alternatively, recent work [LR22] has shown it is possible to obtain NIZKPs that are secure in the General UC framework with the Fiat-Shamir transform when the underlying sigma protocol satisfies specifc criteria. We leave an analysis of the applicability of [LR22] to the proposed construction for future work.

A direct comparison with the protocol by Boneh, Kogan, and Woo [BKW20] is not simple since their bandwidth estimate does not appear to include the Unruh transform overhead. We estimate that one execution of the OPRF from [BKW20] requires at least 10.9 MB[4]. Our protocol is thus more compact than that in [BKW20], despite being round-optimal and secure against both the one-more unpredictability attack and the SIDH attacks. This is made possible by the fact that the sigma protocols are highly optimized and have ternary challenges, which significantly reduces the overhead introduced in the Unruh transform. Indeed, if we compare a version of the two protocols with the Fiat-Shamir transform, our OPRF uses 31% more bandwidth than the one in [BKW20].

We summarize the state of post-quantum OPRF protocols in Table 1. When compared to the CSIDH-based OPRF in [BKW20], our OPRF offers verifiability and a lower number of communication rounds. This comes at the cost of a significantly higher bandwidth; however, if we remove the large server-side proof needed for verifiability, our protocol requires has a transcript of 3.0 MB.

Table 1. Comparison of existing post-quantum OPRF protocols. For verifiable OPRFs, the Rounds column does not include the committment round since it takes place once and it is not repeated during each execution.

Protocol	Rounds	Bandwidth (avg.)	Verifiable	Secure
[ADDS21] (LWE/SIS)	2	>128 GB	✓	✓
[BKW20] (SIDH)	6	>10.9 MB	✓	✗
[BKW20] (CSIDH)	3	424 kB	✗	✓
[This work]	2	8.7 MB	✓	✓

8 Conclusion

In this work, we presented a post-quantum verifiable oblivious PRF protocol that is moderately compact and round-optimal. The protocol is the first round-optimal OPRF based on isogenies, and its communication cost is several orders of magnitude smaller than the existing round-optimal protocol. To obtain this protocol, we started from an insecure protocol by Boneh, Kogan, and Woo, and we proposed an efficient countermeasure against the one-more unpredictability attack, integrated the existing SIDH countermeasures, developed a new zero-knowledge proof of isogeny that works with the SIDH countermeasures, and proposed a novel non-interactive proof of parallel isogeny that reduced the number of rounds to two.

The protocol is an important stepping stone towards fully practical post-quantum OPRFs, but its performance is hindered by the inefficiency of the SIDH countermeasures. In future work, we aim at developing more efficient solutions: a moderate reduction in the degree of the isogenies would significantly improve the efficiency of the protocol. It is also interesting to improve the proof of parallel isogeny by avoiding validating the commitment isogeny at every interaction.

Acknowledgements. The author would like to thank Christophe Petit and Luca de Feo for several suggestions, and Tako Boris Fouotsa, Christophe Petit, Chloe Martindale, and the anonymous reviewers of the workshop PQCifris 2022 for feedback on earlier drafts of this work. The author also thanks Luca de Feo, Antonin Leroux, and Benjamin Wesolowski for fruitful discussions on isogeny-based zero-knowledge proofs at the Banff International Research Station workshop "Supersingular Isogeny Graphs in Cryptography".

[4] In [BKW20, Section 5], the authors estimate that the largest response in the sigma protocol R_{com} requires $6 \log p + 5\lambda$ bits. The protocol has a challenge space of size 6, and it needs to be repeated 3.8λ times to obtain a negligible soundness error. Without considering the size of the committments, the Unruh-based NIZKP contains 6 hashed values that are as long as the largest response, per each iteration. The transcript of an R_{com} proof thus requires at least $3.8\lambda \times 6(6 \log p + 5\lambda)$ bits. The entire OPRF hence requires three times as much (three such proofs are used), plus $5\lambda \log p$ bits for the proof of parallel isogenies.

References

ADDS21. Martin R. Albrecht, Alex Davidson, Amit Deo, and Nigel P. Smart. Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 261–289. Springer, Heidelberg, May 2021. doi:10.1007/978-3-030-75248-4_10.

AJK$^+$16. Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key compression for isogeny-based cryptosystems. In *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography*, pages 1–10. ACM, 2016.

BCC$^+$22. Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. Cryptology ePrint Archive, Paper 2022/1469, 2022. https://eprint.iacr.org/2022/1469. URL: https://eprint.iacr.org/2022/1469.

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

BKW20. Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 520–550. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64834-3_18.

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001. doi:10.1109/SFCS.2001.959888.

CD22. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Report 2022/975, 2022. https://eprint.iacr.org/2022/975.

Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

CLG09. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, January 2009. doi:10.1007/s00145-007-9002-x.

CLN16. Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 572–601. Springer, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_21.

DFDGZ22. Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, Cham, 2022. Springer International Publishing.

DFHSW23. Alex Davidson, Armando Faz-Hernandez, Nick Sullivan, and Christopher A. Wood. Oblivious Pseudorandom Functions (OPRFs) using Prime-Order Groups. Internet-Draft draft-irtf-cfrg-voprf-17, Internet Research Task Force, 2023. Work in Progress.

DGS$^+$18. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *Proc. Priv. Enhancing Technol.*, 2018(3):164–180, 2018. doi:10.1515/popets-2018-0026.

DRRT18. Daniel Demmler, Peter Rindal, Mike Rosulek, and Ni Trieu. PIR-PSI: scaling private contact discovery. *Proc. Priv. Enhancing Technol.*, 2018(4):159–178, 2018. doi:10.1515/popets-2018-0037.

ECS$^+$15. Adam Everspaugh, Rahul Chatterjee, Samuel Scott, Ari Juels, and Thomas Ristenpart. The pythia PRF service. In Jaeyeon Jung and Thorsten Holz, editors, *USENIX Security 2015*, pages 547–562. USENIX Association, August 2015.

EHL$^+$18. Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 329–368. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78372-7_11.

FIPR05. Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 303–324. Springer, Heidelberg, February 2005. doi:10.1007/978-3-540-30576-7_17.

FMP23. Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-sidh and md-sidh: countering sidh attacks by masking information. Cryptology ePrint Archive, Paper 2023/013, 2023. URL: https://eprint.iacr.org/2023/013.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.

Gal18. Steven D. Galbraith. Authenticated key exchange for SIDH. Cryptology ePrint Archive, Report 2018/266, 2018. https://eprint.iacr.org/2018/266.

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

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

JKK14. Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 233–253. Springer, Heidelberg, December 2014. doi:10.1007/978-3-662-45608-8_13.

JKKX16. S. Jarecki, A. Kiayias, H. Krawczyk, and J. Xu. Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online). In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 276–291, Los Alamitos, CA, USA, mar 2016. IEEE Computer Society. doi:10.1109/EuroSP.2016.30.

JKKX17. Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. TOPPSS: Cost-minimal password-protected secret sharing based on threshold OPRF. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17*, volume 10355 of *LNCS*, pages 39–58. Springer, Heidelberg, July 2017. doi:10.1007/978-3-319-61204-1_3.

JKX18. Stanislaw Jarecki, Hugo Krawczyk, and Jiayu Xu. OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 456–486. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78372-7_15.

JL09. Stanislaw Jarecki and Xiaomin Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 577–594. Springer, Heidelberg, March 2009. doi:10.1007/978-3-642-00457-5_34.

LGd21. Yi-Fu Lai, Steven D. Galbraith, and Cyprien de Saint Guilhem. Compact, efficient and UC-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 213–241. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77870-5_8.

LPA$^+$19. Lucy Li, Bijeeta Pal, Junade Ali, Nick Sullivan, Rahul Chatterjee, and Thomas Ristenpart. Protocols for checking compromised credentials. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 1387–1403. ACM Press, November 2019. doi:10.1145/3319535.3354229.

LR22. Anna Lysyanskaya and Leah Namisa Rosenbloom. Universally composable Σ-protocols in the global random-oracle model. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography*, pages 203–233, Cham, 2022. Springer Nature Switzerland.

MM22. Luciano Maino and Chloe Martindale. An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Report 2022/1026, 2022. https://eprint.iacr.org/2022/1026.

NR97. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997. doi:10.1109/SFCS.1997.646134.

PL17. Christophe Petit and Kristin Lauter. Hard and easy problems for supersingular isogeny graphs. Cryptology ePrint Archive, Report 2017/962, 2017. https://eprint.iacr.org/2017/962.

Rob22. Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, Report 2022/1038, 2022. https://eprint.iacr.org/2022/1038.

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

Unr14. Dominique Unruh. Quantum position verification in the random oracle model. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2014. doi:10.1007/978-3-662-44381-1_1.

Unr15. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 755–784. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_25.