A Note on Constructing SIDH-PoK-based Signatures after Castryck-Decru Attack

Jesús-Javier Chi-Domínguez[1]

Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE
jesus.dominguez@tii.ae

Abstract. This paper centers on the SIDH proof of knowledge work by De Feo, Dobson, Galbraith, and Zobernig, which points out that the Castryck-Decru attack does not apply to their first 3-special soundness construction. This work analyzes and explicitly describes an optimized recoverable Sigma protocol based on that 3-special soundness SIDH-PoK construction. We also discuss the impact of moving to B-SIDH and G2SIDH setups in terms of sizes.

Due to the Castryck-Decru attack, we decided to write this paper relying on a theoretical analysis to list expected optimized signature sizes instead of updating eprint 2022/475. We point out that this work is a theoretical analysis extension of eprint 2022/475.

Keywords: Isogeny-based cryptography · Proof-of-Knowledge · Sigma protocol · Signature scheme · Recoverable Sigma protocol

1 Introduction

> *"If someone is able to show me that what I think or do is not right, I will happily change, for I seek the truth, by which no one was ever truly harmed..."*
>
> *Marcus Aurelius*

In 2014, De Feo, Jao, and Plût proposed a post-quantum Diffie-Hellman protocol relying on the hardness of finding an isogeny between two supersingular curves, the SIDH protocol [31,19]. Their work was not only limited to key-exchange procedures; they also presented a Zero-Knowledge protocol based on the SIDH construction. In 2018, Yoo, Azarderakhsh, Jalali, Jao, and Soukharev combined that Zero-Knowledge SIDH with the Fiat-Shamir transformation to get a signature scheme [43]. Independently, Galbraith, Petit, Shani, and Ti improved in [29] the signature sizes of [43], and proposed a signature-scheme based on the problem of computing the endomorphism ring of a supersingular elliptic curve.

In 2021, Ghantous, Katsumata, Pintore, and Veroni revisited the proofs for the special soundness property in the SIDH-based identification protocol [30].

Their analysis relies on collisions in the supersingular isogeny graph; assuming evenly distributed cycles over the vertex set, their existence does not affect the security of the SIDH-based signatures. Subsequently, De Feo, Dobson, Galbraith, and Zobernig [17] found an issue and provided a counterexample, with the soundness proof for the Zero-Knowledge SIDH construction. Such an issue applies to the constructions from [43] and [29], but the authors stressed that SIDH signature schemes are still secure, a reasonable computational assumption, and no known attack exists yet. Additionally, [17] presents an isogeny-based Proof of Knowledge (PoK) that relies on a new hardness assumption, and is immune to previously presented adaptive attacks [28,2,23,27]. The main result from [17] proposes an efficient non-interactive SIDH-key validation. The principal difference between [19,29] and [17] constructions is that they have 2-special and 3-special soundness, respectively.

Sadly, the recent work by Castryck and Decru [9] presented a (heuristically) polynomial SIDH key-recovery attack that breaks SIDH (and SIKE) in hours. The three vital ingredients for the applicability of the Castryck-Decru attack are

- The public and fixed isogeny degree;
- The image of the auxiliary torsion points under the secret isogeny; and
- The endomorphism ring of the isogeny domain curve.

The followed-up work by Maino and Martindale in [37] provided an algorithm that does not require the knowledge of the endomorphism ring of the domain curve. Subsequently, Robert demonstrated the existence of a polynomial key-recovery attack on SIDH [40]. Even the works from [37] and [40] remain theoretical; Castryck and Decru gave a public Magma code implementation of their attack, which was improved by Oudompheng and Pope in Sagemath code [38]. It is worth mentioning that Castryck-Decru's family attacks apply to [31,19,43] but do not extend to the construction from [17, §5.3] and the quaternion-based proposal of [29].

As the primary motivation of this work, it is of interest to determine the efficiency (in sizes) for the 3-special soundness construction in [17, §5.3] and analyze the impact of using B-SIDH [14] and G2SIDH [33] in such a 3-special soundness protocol, hoping to reduce sizes.

Related work. In 2019, De Feo and Galbraith proposed a signature scheme named SeaSign by combining the Commutative SIDH (CSIDH) [11] and Fiat-Shamir transformation with aborts [18]. SeaSign aims to have shorter keys than lattice signatures, but signing and verification are currently costly. Later, Decru, Panny, and Vercauteren improved SeaSign performance by allowing the prover not to answer a limited number of said parallel executions to decrease the rejection probability [22]. Subsequently, Beullens, Kleinjung, and Vercauteren introduced a promising signature scheme labeled as CSI-FiSh [6] by integrating similar optimizations of SeaSign on Stolbunov's signature scheme [41]. They showed that including quadratic twists cuts the public key size in half, being 300 times faster and about three times smaller than any optimized version of SeaSign. In 2020, Kaafarani, Katsumata, and Pintore suggested a Lossy variant of CSI-FiSh with smaller signature sizes but two times slower than the original CSI-FiSh [24].

A disadvantage of SeaSign, CSI-FiSh and its lossy variant, and the new scheme from [5], is that their current proposals and implementations use CSIDH-512, which seems to bring lower quantum security than NIST Level 1 [8,39,7,12]. In particular, such state-of-the-art works hint CSIDH instances with 2048 bits are good choices to close NIST Level 1 of security. Nevertheless, using large CSIDH instantiations (with about 2048 bits) would considerably slowdown on the performance and increase public-key sizes for these CSIDH-based schemes, negatively impacting higher security levels compared to NIST Levels 3 and 5. The signature sizes remain the same, which makes CSIDH-based signature attractive.

Lastly, De Feo, Kohel, Leroux, Petit, and Wesolowski introduced the current shortest isogeny-based signature scheme SQIsign [20]. They only target NIST level 1 of security, with signatures of 204 bytes, secret keys of 16 bytes, and public keys of 64 bytes; their C-code implementation claims 0.6 seconds for key generation, 2.5 seconds for signing, and 50 milliseconds for verification.

Contributions. We provide a detailed description to construct a Signature scheme based on [17, Section 5.3]. We explicitly describe a non-interactive recoverable Sigma protocol over isogenies. Such sigma protocols prove the knowledge of an isogeny under the Fixed degree relation given in [17,4]. We also estimate the expected signature sizes by using built-in blocks SIDH, B-SIDH, and G2SIDH; we applied (as far as we know) all possible tricks to reduce signature sizes as much as possible.

Outline. We organize the paper as follows. We present all mathematical tools required to describe the 3-special soundness construction from [17, §5.3] in Section 2. Since the Sigma protocol proves the knowledge of an isogeny by using SIDH as a built-in block, we explain SIDH in Section 2.1 and the Sigma protocol in Section 2.2. To understand how the Sigma protocol works, we proceed in Section 3.1 to detail tricks to reduce its commitment and response sizes. After that, we present in Section 3.2 a recoverable Sigma protocol to construct a signature protocol. Subsequently, we mention in Section 3.3 that replacing SIDH with B-SIDH reduces the sizes. We show in Section 3.4 how G2SIDH can help to reduce the sizes even more [1]. In Section 3.5, we list (to the best of our knowledge) all isogeny-based signatures still secure against the Castryck-Decru attack. Finally, we conclude with some open problems and remarks in Section 4.

2 Preliminaries

In this section, we introduce all mathematical tools required in the SIDH constructions from [31,19]. Let $p = 2^a 3^b - 1$ be a prime number satisfying $p \equiv$

---

[1] We highlight that we did not dig into the mathematical tools required for G2SIDH; we took it as a black box. However, we mention the main differences between SIDH and G2SIDH and take essential properties to describe how the recoverable Sigma protocol will impact.

3 mod 4 for some $a, b \in \mathbb{Z}^+$. Let $\mathbb{F}_p$ be a prime field with p elements and $\mathbb{F}_{p^2}$ a quadratic field extension of $\mathbb{F}_p$. We let E be a supersingular curve determined by Equation 1 and assume E has exactly $\#E(\mathbb{F}_{p^2}) = (p+1)^2$ points over $\mathbb{F}_{p^2}$.

$$E\colon y^2 = x^3 + Ax^2 + x, \quad A \in \mathbb{F}_{p^2} \setminus \{\pm 2\}. \tag{1}$$

The point at infinity ∞ of E plays the role of the neutral element. We say $P \in E$ is an order-d point if d is the smallest positive integer such that

$$[d]P = \underbrace{P + \cdots + P}_{d \text{ times}} = \infty,$$

and write $E[d]$ to denote the d-torsion subgroup $\{P \in E(\overline{\mathbb{F}}_{p^2}) \mid [d]P = \infty\}$. The j-invariant of the curve E is $\frac{256(A^2-3)^3}{A^2-4}$.

Isogenies From Kernel. We only consider separable isogenies. An isogeny $\phi\colon E \to E'$ over $\mathbb{F}_{p^2}$ is a non-zero rational map fixing the point at infinity, $\phi(\infty) = \infty$. If such isogeny exists, we say E and E' are isogenous over $\mathbb{F}_{p^2}$, which happens if and only if $\#E(\mathbb{F}_{p^2}) = \#E'(\mathbb{F}_{p^2})$. The kernel $\ker\phi$ of ϕ is the subgroup $\{P \in E(\mathbb{F}_{p^2}) \mid \phi(P) = \infty\}$. We refer to ϕ as d-isogeny when $\#\ker\phi = d$ holds. The dual d-isogeny $\widehat{\phi}\colon E' \to E$ of ϕ is the isogeny satisfying

$$\widehat{\phi} \circ \phi\colon P \mapsto [d]P \quad \text{and} \quad \phi \circ \widehat{\phi}\colon P \mapsto [d]P.$$

2.1 SIDH protocol

The core idea of [17, §5.3] relies on the SIDH-square construction. So, let us list the SIDH setup as follows:

- the quadratic field extension $\mathbb{F}_{p^2}$ of $\mathbb{F}_p$ along with $p = 2^a 3^b - 1$;
- the starting supersingular curve $E_0\colon y^2 = x^3 + 6x^2 + x$ [2];
- the order-2^a basis $\{P_0, Q_0\}$ satisfying $\langle P_0, Q_0\rangle = E_0[2^a]$; and
- the order-3^b basis $\{P_0', Q_0'\}$ satisfying $\langle P_0', Q_0'\rangle = E_0[3^b]$.

The SIDH key generation is slightly different for each entity. Alice generates public keys according to order-3^b points, and her private keys determine secret 2^a-isogenies. In contrast, Bob's public keys are concerning order-2^a points and his private keys to 3^b-isogenies. We sketch as follows Alice and Bob's key generations and derivations.

---

[2] We choose the same E_0 as in
– SIKE proposal [1], but it can be a different curve.

Alice key generation.

1. Alice samples a random integer $\mathsf{sk} \xleftarrow{\$} \llbracket 0 \mathinner{..} 2^a - 1 \rrbracket$ as her private key;
2. She then computes the 2^a-isogeny $\phi\colon E_0 \to E_1$ with kernel generated by $K_\phi = P_0 + [\mathsf{sk}]Q_0$; and
3. She sets as her public key $\mathsf{pk} = (E_1, \phi(P_0'), \phi(Q_0'))$, and send it to Bob.

Bob key generation.

1. Bob samples a random integer $\mathsf{sk}' \xleftarrow{\$} \llbracket 0 \mathinner{..} 3^b - 1 \rrbracket$ as his private key;
2. He then computes the 3^b-isogeny $\psi\colon E_0 \to E_2$ with kernel generated by $K_\psi = P_0' + [\mathsf{sk}']Q_0'$; and
3. He sets as his public key $\mathsf{pk}' = (E_2, \psi(P_0), \psi(Q_0))$, and send it to Alice.

Alice key derivation.

1. Alice computes the 2^a-isogeny $\phi'\colon E_2 \to E_3$ with kernel generated by $K_{\phi'} \coloneqq \psi(K_\phi) = \psi(P_0) + [\mathsf{sk}]\psi(Q_0)$; and
2. She finally sets as her secret shared the j-invariant $j(E_3)$ of E_3.

Bob key derivation.

1. Bob computes the 3^b-isogeny $\psi'\colon E_1 \to E_3'$ with kernel generated by $K_{\psi'} \coloneqq \phi(K_\psi) = \phi(P_0') + [\mathsf{sk}']\phi(Q_0')$; and
2. He finally sets as his secret shared the j-invariant $j(E_3')$ of E_3'.

In the original SIDH construction from [31,19] and also in [1], the secret shared corresponds with the j-invariant of the curves E_3 and E_3'. However, Leonardi showed that the ending curves E_3 and E_3' are equal to each other [35]. We illustrate the diagram determined by the SIDH protocol in Figure 1.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\phi"] \arrow[d, dashed, "\psi"'] & E_1 \arrow[d, dashed, "\psi'"] \\
E_2 \arrow[r, dashed, "\phi'"'] & E_3
\end{tikzcd}$$

Fig. 1: Dashed arrows are secret and all curves are public. Horizontal and vertical arrows denote 2^a-isogenies and 3^b-isogenies, respectively. .

Next, we summarize the constructions from [17] in Section 2.2. In particular, we only focus on the constructions based on Definition 1. The idea behind [17, §5.3] is to randomly generate SIDH-squares, as illustrated in Figure 1, to prove the knowledge of the secret isogeny.

Definition 1 (Fixed degree relation). *Given a public curve* $\mathsf{pk} = E_i$ *generated by Alice or Bob without revealing any image of auxiliary points, we define the Fixed degree relation by Equation 2.*

$$\mathcal{R}_{\mathrm{deg}} \coloneqq \{(E_0, E_i, d, \omega) \mid \omega\colon E_0 \to E_i \textit{ is a } d\textit{-isogeny}\}\,. \tag{2}$$

2.2 The still secure Sigma protocol 3-special sound

This section describes the construction from [17, §5.3]. The setup is the same as in Section 2.1. Given a public 2^a-isogenous curve E_1 to E_0. The prover (Peggy) wants to convince the verifier (Victor) that she knows the secret 2^a-isogeny $\phi\colon E_0 \to E_1$, which implies knowing $\ker\phi = \langle K_\phi\rangle$. Let $\lambda \in \{128, 192, 256\}$ a security parameter, and $\mathcal{H}$ be a cryptographic hash function with output length 2λ.

Public and private keys. Here, the public key is $\mathsf{pk} = E_1$, while $\mathsf{sk} = \phi$ determines the private key.

Commitment. This block proceeds by constructing random SIDH-squares described in Figure 1 as follows.

- Peggy picks a random order-3^b kernel generator K_ψ in E_0;
- She evaluates K_ψ under the secret isogeny ϕ to get $K_{\psi'} = \phi\left(K_\psi\right)$;
- She constructs an SIDH-square as in Figure 1 determined by
  - the 3^b-isogeny $\psi\colon E_0 \to E_2$ with $\ker\psi = \langle K_\psi\rangle$,
  - the 3^b-isogeny $\psi'\colon E_1 \to E_3$ with $\ker\psi' = \langle K_{\psi'}\rangle$, and
  - the 2^a-isogeny $\phi'\colon E_2 \to E_3$ with $\ker\phi' = \langle K_{\phi'}\rangle$ where $K_{\phi'} = \psi\left(K_\phi\right)$;
- She chooses a random basis $\{P_2, Q_2\}$ of $E_2\left[3^b\right]$;
- She evaluates P_2 and Q_2 under the secret isogeny ϕ' to get $P_3 = \phi'\left(P_2\right)$ and $Q_3 = \phi'\left(Q_2\right)$;
- She looks for two integers $c, d \in [\![0\,..\,3^b-1]\!]$ such that
  - The dual isogeny $\widehat{\psi}\colon E_2 \to E_0$ of ψ has kernel generator $K_{\widehat{\psi}} = [c]P_2 + [d]Q_2$, and
  - The dual isogeny $\widehat{\psi'}\colon E_3 \to E_1$ of ψ' has kernel generator $K_{\widehat{\psi'}} = [c]P_3 + [d]Q_3$;
- She selects three random numbers r_R, r_L, and r from $\{0,1\}^\lambda$.
- Next, She commits $\mathtt{com}_2 = (E_2, P_2, Q_2)$ and $\mathtt{com}_3 = (E_3, P_3, Q_3)$ as
  - $\mathtt{com}_L = \mathcal{H}\left(\mathtt{com}_2 \mid\mid r_L\right)$,
  - $\mathtt{com}_R = \mathcal{H}\left(\mathtt{com}_3 \mid\mid r_R\right)$, and
  - $\mathtt{com}' = \mathcal{H}\left((c,d) \mid\mid r\right)$;
- Finally, She sends the commitment message $\mathtt{com} \leftarrow (\mathtt{com}_L, \mathtt{com}_R, \mathtt{com}')$ to Victor.

Challenge. Victor picks a uniformly random challenge $\mathtt{chall} \xleftarrow{\$} \{-1, 0, 1\}$, and send it to Peggy.

Response. Once Peggy receives the challenge $\texttt{chall}$, she performs the following:

- If $\texttt{chall} = 1$, she sends $\texttt{resp} \leftarrow (\texttt{com}_2, r_L, K_{\phi'}, \texttt{com}_3, r_R)$ to Victor.
- If $\texttt{chall} = 0$, she sends $\texttt{resp} \leftarrow (\texttt{com}_3, r_R, c, d, r)$ to Victor.
- If $\texttt{chall} = -1$, she sends $\texttt{resp} \leftarrow (\texttt{com}_2, r_L, c, d, r)$ to Victor.

Verification. Depending on the challenge, Victor does the following calculations to validate the commitment and response:

- $(\texttt{com}_L, \texttt{com}_R, \texttt{com}') \leftarrow \texttt{com}$
- If $\texttt{chall} = 1$,
  - He parses
    - $(\texttt{com}_2, r_L, K_{\phi'}, \texttt{com}_3, r_R) \leftarrow \texttt{resp}$,
    - $(E_2, P_2, Q_2) \leftarrow \texttt{com}_2$, and
    - $(E_3, P_3, Q_3) \leftarrow \texttt{com}_3$;
  - He **rejects** if $\mathcal{H}\left(\texttt{com}_2 \mid\mid r_L\right) \neq \texttt{com}_L$ or $\mathcal{H}\left(\texttt{com}_3 \mid\mid r_R\right) \neq \texttt{com}_R$;
  - He **rejects** if $K_{\phi'} \notin E_2$ or $K_{\phi'}$ does not have order 2^a;
  - He computes the 2^a-isogeny $\phi' \colon E_2 \to E_3'$ with kernel generator $K_{\phi'}$;
  - Finally, Victor **accepts** if and only if $E_3 = E_3'$, $P_3 = \phi'(P_2)$ and $Q_3 = \phi'(Q_2)$, otherwise **rejects**.
- If $\texttt{chall} = 0$,
  - He parses
    - $(\texttt{com}_3, r_R, c, d, r) \leftarrow \texttt{resp}$, and
    - $(E_3, P_3, Q_3) \leftarrow \texttt{com}_3$;
  - Victor **rejects** if $\mathcal{H}\left((c, d) \mid\mid r\right) \neq \texttt{com}'$ or $\mathcal{H}\left(\texttt{com}_3 \mid\mid r_R\right) \neq \texttt{com}_R$;
  - He computes $K_{\widehat{\psi'}}$ as $[c]P_3 + [d]Q_3$;
  - He **rejects** if $K_{\psi'}$ does not have order 3^b;
  - He computes the 3^b-isogeny $\psi' \colon E_3 \to E_1'$ with kernel generator $K_{\psi'}$;
  - Finally, Victor **accepts** if and only if $E_1 = E_1'$, otherwise **rejects**.
- If $\texttt{chall} = -1$,
  - He parses
    - $(\texttt{com}_2, r_L, c, d, r) \leftarrow \texttt{resp}$, and
    - $(E_2, P_2, Q_2) \leftarrow \texttt{com}_2$;
  - Victor **rejects** if $\mathcal{H}\left(\texttt{com}_2 \mid\mid r_L\right) \neq \texttt{com}_L$ or $\mathcal{H}\left((c, d) \mid\mid r\right) \neq \texttt{com}'$;
  - He computes $K_{\widehat{\psi}}$ as $[c]P_2 + [d]Q_2$;
  - He **rejects** if K_{ψ} does not have order 3^b;
  - He computes the 3^b-isogeny $\psi \colon E_2 \to E_0'$ with kernel generator K_{ψ};
  - Finally, Victor **accepts** if and only if $E_0 = E_0'$, otherwise **rejects**.

Remark 1. The computations in the **Response** and **Verification** concerning the case $\texttt{chall} = 1$ correspond with the horizontal arrows of Figure 1. While $\texttt{chall} = 0$ and $\texttt{chall} = -1$ determines the right-vertical and left-vertical arrows, respectively.

The current wave of attacks by Castryck-Decru [9], Maino-Martindale [37], and Robert [40] do not extend to the Sigma protocol from [17, §5.3], which is described above in Section 2.2. Given that the public keys do not include images of any auxiliary point the current Castryck-Decru family attacks do not help to find (either in a polynomial or subexponential time) the secret isogeny ϕ. Additionally,

- If `chall` $= 1$. The kernel generator $K_{\phi'}$ of $\phi' \colon E_2 \to E_3$ is revealed, along with the points P_2, Q_2 and their respectively image $P_3 = \phi'(P_2)$ and $Q_3 = \phi'(Q_2)$. Therefore, any key-recovery attack from [9,37,40] recovers a kernel generator for the 2^a-isogeny ϕ', which is already public.
- If `chall` $= 0$. The kernel generator $K_{\widehat{\psi'}}$ of the (expected) dual 3^b-isogeny $\widehat{\psi'} \colon E_3 \to E_1$ is public, along with the image points $P_3 = \phi'(P_2)$ and $Q_3 = \phi'(Q_2)$. Now, the curve E_2 and the points $P_2, Q_2 \in E_2$ are not revealed, and thus the points P_3 and Q_3 looks like random points. Furthermore, there are no image of auxiliary points under ϕ' (or its dual). So, the current Castryck-Decru family attacks do not help to find the secret 2^a-isogeny ϕ'.
- If `chall` $= -1$. The kernel generator $K_{\widehat{\psi}}$ of the (expected) dual 3^b-isogeny $\widehat{\psi} \colon E_2 \to E_0$ is public, along with two random points P_2 and Q_2. Now, the curve E_3 and the random points $P_3, Q_3 \in E_2$ are not revealed. In fact, there are no image of auxiliary points under ϕ (or its dual). So, the current Castryck-Decru family attacks do not help to find the secret 2^a-isogeny ϕ.

2.3 Sigma protocol & the Fiat-Shamir transform

As a way to describe the security assumption, Figure 2 illustrates the hard problem of the Sigma protocol from Section 2.2, and assumes the cases from Figure 2a, Figure 2b, and Figure 2c do not simultaneously occur for a fixed instance. Essentially, the hardness assumption relies on distinguishing between well-formed and altered instances (E_2, E_3, ϕ'), that is on the Decisional Supersingular Product Problem (DSPP) [17].

Definition 2 (Decisional Supersingular Product Problem (DSPP): Alice's case). *Let E_0 be a Montgomery curve as in the SIDH setting (see Section 2.1). Given a 2^a-isogeny $\phi \colon E_0 \to E_1$ with kernel $\langle K_\phi \rangle$, the Decisional Supersingular Product Problem (DSPP) asks to distinguish between the following two distributions:*

- *(E_2, E_3, ϕ') is the bottom of a random SIDH-square as in Figure 1. That is, for a randomly chosen order-3^b kernel $\langle K_\psi \rangle$, we have E_2 is the codomain curve of the 3^b-isogeny ψ with kernel $\langle K_\psi \rangle$, E_3 is the codomain curve of the 3^b-isogeny ψ' with kernel $\langle \phi(K_\psi) \rangle$, and $\phi' \colon E_2 \to E_3$ is the 2^a-isogeny with kernel $\langle \psi(K_\phi) \rangle$.*
- *(E_2, E_3, ϕ') such that E_2 is a randomly chosen elliptic curve with same cardinality as E_0, and $\phi' \colon E_2 \to E_3$ is a random 2^a-isogeny with cyclic kernel.*

The sigma protocol described in Section 2.2 is 3-special soundness under the relation given by Definition 1. Furthermore, when repeated λ times, it becomes a Special Honest-Verifier Zero-Knowledge (SHVZK) PoK with soundness $(2/3)^{\kappa}$, assuming the DSPP is computationally hard and the commitment scheme determined by $\mathcal{H}$ is computationally binding and statistically hiding [17, Theorem 4].

Signature scheme using the strong Fiat-Shamir transform [25,3]. The main idea is to avoid the interaction between Peggy and Victor by allowing Peggy to generate the challenge as the hash of the statement and the commitment. In our case, Peggy would first generate κ commitments $\texttt{com}_i$ and then obtains the challenge $(\texttt{chall}_1, \ldots, \texttt{chall}_{\kappa-1}) = \mathcal{RO}(\mathsf{pk}, m, \texttt{com}_0, \ldots, \texttt{com}_{\kappa-1})$, where m is the message to be signed. We denote by $\mathcal{RO}$ a random oracle that outputs strings in $\{-1, 0, 1\}^{\kappa}$. Each challenge $\texttt{chall}_i$ determines the response values for $\texttt{com}_i$. This transformation is secure [42] in the Quantum Random Oracle Model (QROM).

$$\begin{tikzcd}
E_0' = E_0? \arrow[r, dashed, "\phi"] & E_1 \arrow[d, dashed, "\psi'"] \\
E_2 \arrow[u, "\widehat{\psi}"] \arrow[r, dashed, "\phi'"'] & E_3
\end{tikzcd}$$

(a) Given $\ker \widehat{\psi} = \langle K_{\widehat{\psi}} \rangle$. The prover **accepts** if the codomain curve E_0' of $\widehat{\psi}$ is equal to E_0; otherwise **rejects**.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\phi"] \arrow[d, dashed, "\psi"'] & E_1' = E_1? \\
E_2 \arrow[r, dashed, "\phi'"'] & E_3 \arrow[u, "\widehat{\psi}'"']
\end{tikzcd}$$

(b) Given $\ker \widehat{\psi}' = \langle K_{\widehat{\psi}'} \rangle$. The prover **accepts** if the codomain curve E_1' of $\widehat{\psi}'$ is equal to E_1; otherwise **rejects**.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\phi"] \arrow[d, dashed, "\psi"'] & E_1 \arrow[d, dashed, "\psi'"] \\
E_2 \arrow[r, "\phi'"'] & E_3' = E_3?
\end{tikzcd}$$

(c) Given $\ker \phi' = \langle K_{\phi}' \rangle$. The verifier **accepts** if and only if the codomain curve E_3' of ϕ' is equal to E_3, $P_3 = \phi'(P_2)$ and $Q_3 = \phi'(Q_2)$; otherwise **rejects**.

Fig. 2: Dashed arrows and curves labeled with gray ink are secret and unknown by the adversary and distinguisher.

3 Efficient Sigma construction built-in functions

This section describes a way to optimize the Sigma construction described in Section 2.2 via recoverable Sigma protocols and applying the tricks from [29] and [4].

3.1 Reducing sizes according state-of-the-art tricks

A commitment $\texttt{com} = (\texttt{com}_L, \texttt{com}_R, \texttt{com}')$ has fixed bitlength equals 6λ. Recall

- $\texttt{com}_L = \mathcal{H}(\texttt{com}_2 \mid\mid r_L)$ with $\texttt{com}_2 = (E_2, P_2, Q_2)$;
- $\texttt{com}_R = \mathcal{H}(\texttt{com}_3 \mid\mid r_R)$ with $\texttt{com}_3 = (E_3, P_3, Q_3)$; and
- $\texttt{com}' = \mathcal{H}((c,d) \mid\mid r)$ where $K_\psi = [c]P_2 + [d]Q_2$ and $K_{\psi'} = [c]P_3 + [d]Q_3$ hold.;

The response $\texttt{resp}$ has a different size depending on if $\texttt{chall} = 1$ holds; let us analyze the cases below:

Case $\texttt{chall} \neq 1$. The response includes $\log_2(p)$ bits that determines (c,d). Notice, we can do it better by computing either $\mathit{\Delta} = \left(cd^{-1} \bmod 3^b\right)$ or $\mathit{\Delta} = \left(dc^{-1} \bmod 3^b\right)$ plus one bit $b \in \{0,1\}$ to decide which point is multiplied by $\mathit{\Delta}$: either $P_j + [\mathit{\Delta}]Q_j$ or $[\mathit{\Delta}]P_j + Q_j$ as kernel point generator for $j \coloneqq 2,3$. In other words, we suggest to replace (c,d) by $(b,\mathit{\Delta})$, and update the commit $\texttt{com}'$ as $\mathcal{H}((b,\mathit{\Delta}) \mid\mid r)$. That trick reduces (c,d) of $\log_2(p)$ bits to $(b,\mathit{\Delta})$ of $\frac{\log_2(p)}{2}$ bits. Now, let $\texttt{CanonicalBasis}_3(E)$ denotes the procedure to find two order-3^b points P' and Q' such that $\langle P', Q'\rangle = E[3^b]$, and set $j \in \{2,3\}$. The commitment $\texttt{com}_j = (E_j, P_j, Q_j)$ has $6\log_2(p)$ bits. The idea is to compute $P', Q' \leftarrow \texttt{CanonicalBasis}_3(E_j)$ and find integers $\alpha_{P_j}, \alpha_{Q_j}, \beta_{P_j}, \beta_{Q_j} \in [\![0 \mathinner{..} 3^b - 1]\!]$ such that $P_j = [\alpha_{P_j}]P' + [\beta_{P_j}]Q'$ and $Q_j = [\alpha_{Q_j}]P' + [\beta_{Q_j}]Q'$. Therefore, replace the commitment $\texttt{com}_j = (E_j, P_j, Q_j)$ by $\texttt{com}_j = \left(E_j, (\alpha_{P_j}, \beta_{P_j}), (d_{Q_j}, \beta_{Q_j})\right)$. That trick reduces the sizes from $6\log_2(p)$ bits to about $4\log_2(p)$ bits.

Case $\texttt{chall} = 1$. The response includes both $\texttt{com}_2$ and $\texttt{com}_3$, along with the kernel order-2^a point generator $K_{\phi'}$. Same trick as in the case $\texttt{chall} \neq 1$ allows to reduce the commitment size of $(\texttt{com}_2, \texttt{com}_3)$ from $12\log_2(p)$ to $8\log_2(p)$; since we can get P_3 and Q_3 from P_2 and Q_2 using ϕ', we do not need to include P_3 and Q_3 in the response, which reduces up to $6\log_2(p)$ bits. Let $\texttt{CanonicalBasis}_2(E_2)$ denotes the procedure to find two order-2^a points P and Q such that $\langle P, Q\rangle = E_2[2^a]$. Analogously to the 3^b-torsion basis case, we can reduce $K_{\phi'}$ by finding two integers $\alpha, \beta \in [\![0 \mathinner{..} 2^a - 1]\!]$ such that $K_{\phi'} = [\alpha]P + [\beta]Q$. Moreover, we suggest to represent $K_{\phi'}$ using $\frac{\log_2 p}{2}$ by computing either $\mathit{\Delta}_2 = \left(\alpha\beta^{-1} \bmod 2^a\right)$ or $\mathit{\Delta}_2 = \left(\beta\alpha^{-1} \bmod 2^a\right)$ plus one bit $b_2 \in \{0,1\}$ to decide which point is multiplied by $\mathit{\Delta}_2$: either $P + [\mathit{\Delta}_2]Q$ or $[\mathit{\Delta}_2]P + Q$ as kernel point generator.

Reducing via recoverable Sigma protocol. Following the hints from [4, c.f. Remark 3], we transform the Sigma protocol into a recoverable Sigma protocol. That is, the signer can output $(\texttt{chall}, \texttt{resp})$ as signature instead of $(\texttt{com}, \texttt{resp})$. Given a signature $(\texttt{chall}, \texttt{resp})$, Victor then first recomputes $\texttt{com}$, and checks that $\texttt{chall} = \mathcal{H}(\texttt{pk}, m, \texttt{com})$ before verfiying the transcript.

3.2 Explicit description of an efficient recoverable Sigma protocol

Let us assume Peggy wants to convince Victor that she knows the secret 2^a-isogeny $\phi \colon E_0 \to E_1$, which implies knowing $\ker \phi = \langle K_\phi \rangle$. Let m be a message to be signed.

Signing. Peggy proceeds as follows:

- She computes $(\texttt{com}_2, r_L)$, $(\texttt{com}_3, r_R)$, $((c,d), r)$, and
- $K_{\phi'}$ as in the $\texttt{commitment}$ procedure from Section 2.2;
- She evaluates
  - $\texttt{com}_L = \mathcal{H}\,(\texttt{com}_2 \mid\mid r_L)$,
  - $\texttt{com}_R = \mathcal{H}\,(\texttt{com}_3 \mid\mid r_R)$, and
  - $\texttt{com}' = \mathcal{H}\,((b, \Delta) \mid\mid r)$ where (b, Δ) are computed as in Section 3.1;
- She calculates $\texttt{com}_{\mathcal{H}} \leftarrow \mathcal{H}(\texttt{pk} \mid\mid m \mid\mid \texttt{com})$ with $\texttt{com} = (\texttt{com}_L, \texttt{com}_R, \texttt{com}')$;
- She picks as random challenge as $\texttt{chall} \leftarrow \texttt{PRNG}\,(\texttt{com}_{\mathcal{H}}) \in \{-1, 0, 1\}$;
  - If $\texttt{chall} = 1$, she gets (b_2, Δ_2), $(\alpha_{P_j}, \beta_{P_j})$, and $(\alpha_{Q_j}, \beta_{Q_j})$ for $j \coloneqq 2, 3$ as in Section 3.1, and sets

$$\texttt{resp} \leftarrow \big(\texttt{com}', E_2, (\alpha_{P_2}, \beta_{P_2}), (\alpha_{Q_2}, \beta_{Q_2}), r_L, (b_2, \Delta_2), E_3, r_R\big);$$

  - If $\texttt{chall} = 0$, she obtains $(\alpha_{P_3}, \beta_{P_3})$ and $(\alpha_{Q_3}, \beta_{Q_3})$ as in Section 3.1, and sets

$$\texttt{resp} \leftarrow \big(\texttt{com}_L, E_3, (\alpha_{P_3}, \beta_{P_3}), (\alpha_{Q_3}, \beta_{Q_3}), r_R, (b, \Delta), r\big);$$

  - If $\texttt{chall} = -1$, she computes $(\alpha_{P_2}, \beta_{P_2})$ and $(\alpha_{Q_2}, \beta_{Q_2})$ as in Section 3.1, and sets

$$\texttt{resp} \leftarrow \big(\texttt{com}_R, E_2, (\alpha_{P_2}, \beta_{P_2}), (\alpha_{Q_2}, \beta_{Q_2}), r_L, (b, \Delta), r\big);$$

- Finally, Peggy sends $\sigma \leftarrow (\texttt{chall}_{\mathcal{H}}, \texttt{resp})$ to Victor.

Verifying. Victor does the below calculations to validate the signature $\sigma = (\texttt{chall}_{\mathcal{H}}, \texttt{resp})$:

- He computes the challenge as $\texttt{chall} \leftarrow \texttt{PRNG}\,(\texttt{com}_{\mathcal{H}}) \in \{-1, 0, 1\}$;
- If $\texttt{chall} = 1$,
  - He takes $\texttt{com}'$, r_L, and r_R from $\texttt{resp}$;
  - He reconstructs $\texttt{com}_2 = (E_2, P_2, Q_2)$, E_3, and $K_{\phi'}$ from $\texttt{resp}$;
  - He computes the 2^a-isogeny $\phi' \colon E_2 \to E_3'$ with kernel generator $K_{\phi'}$;
  - He evaluates P_2 and Q_2 under ϕ' to get $P_3 = \phi'(P_2)$ and $Q_3 = \phi'(Q_2)$;
  - He sets $\texttt{com}_3 = (E_3, P_3, Q_3)$, and calculates $\texttt{com}_L = \mathcal{H}\,(\texttt{com}_2 \mid\mid r_L)$ and $\texttt{com}_R = \mathcal{H}\,(\texttt{com}_3 \mid\mid r_R)$;
  - He **rejects** if $\mathcal{H}(\texttt{pk} \mid\mid m \mid\mid \texttt{com}) \neq \texttt{com}_{\mathcal{H}}$ where $\texttt{com} = (\texttt{com}_L, \texttt{com}_R, \texttt{com}')$;
  - Finally, Victor **accepts** if and only if $E_3 = E_3'$, otherwise **rejects**.
- If $\texttt{chall} = 0$,

  - He takes $((b, \varDelta), r)$, $\texttt{com}_L$, and r_R from $\texttt{resp}$;
  - He reconstructs $\texttt{com}_3 = (E_3, P_3, Q_3)$ from $\texttt{resp}$;
  - He calculates $\texttt{com}' = \mathcal{H}\left((b, \varDelta) \mid\mid r\right)$ and $\texttt{com}_R = \mathcal{H}\left(\texttt{com}_3 \mid\mid r_R\right)$;
  - He **rejects** if $\mathcal{H}(\mathsf{pk} \mid\mid m \mid\mid \texttt{com}) \neq \texttt{com}_{\mathcal{H}}$ where $\texttt{com} = (\texttt{com}_L, \texttt{com}_R, \texttt{com}')$;
  - He calculates $K_{\psi'}$ using P_3, Q_3, and $(b, \varDelta)$;
  - He computes the 3^b-isogeny $\psi' \colon E_3 \to E_1'$ with kernel generator $K_{\psi'}$;
  - Finally, Victor **accepts** if and only if $E_1 = E_1'$, otherwise **rejects**.
- If $\texttt{chall} = -1$,
  - He takes $((b, \varDelta), r)$, $\texttt{com}_R$, and r_L from $\texttt{resp}$;
  - He reconstructs $\texttt{com}_2 = (E_2, P_2, Q_2)$ from $\texttt{resp}$;
  - He calculates $\texttt{com}' = \mathcal{H}\left((b, \varDelta) \mid\mid r\right)$ and $\texttt{com}_L = \mathcal{H}\left(\texttt{com}_2 \mid\mid r_L\right)$;
  - He **rejects** if $\mathcal{H}(\mathsf{pk} \mid\mid m \mid\mid \texttt{com}) \neq \texttt{com}_{\mathcal{H}}$ where $\texttt{com} = (\texttt{com}_L, \texttt{com}_R, \texttt{com}')$;
  - He calculates K_{ψ} using P_2, Q_2, and $(b, \varDelta)$;
  - He computes the 3^b-isogeny $\psi \colon E_2 \to E_0'$ with kernel generator K_{ψ};
  - Finally, Victor **accepts** if and only if $E_0 = E_0'$, otherwise **rejects**.

Notice, if $\texttt{chall} = 1$ then the response $\texttt{resp}$ in the above recoverable Sigma protocol has $\frac{8\lambda + 13\log_2(p)}{2}$ bits; otherwise, it has $\frac{8\lambda + 9\log_2(p)}{2}$ bits. Therefore, in average the response $\texttt{resp}$ has $\frac{24\lambda + 31\log_2(p)}{6} \approx (4\lambda + 5\log_2(p))$ bits. As the last optimization, we suggest taking

$$(\texttt{chall}_0, \ldots, \texttt{chall}_{\kappa-1}) \leftarrow \mathcal{RO}\left(\mathcal{H}'(\texttt{com}_{\mathcal{H},0}, \ldots, \texttt{com}_{\mathcal{H},\kappa-1})\right)$$

as κ challenges for κ repetitions of the above recoverable Sigma protocol, where $\mathcal{H}'$ is a hash function return λ-bits and $\mathcal{RO}$ is a random oracle that uniformly samples from $\{-1, 0, 1\}^{\kappa}$. After that, we get a signature

$$\sigma = \left(\mathcal{H}'(\texttt{com}_{\mathcal{H}',0}, \ldots, \texttt{com}_{\mathcal{H},\kappa-1}), \texttt{resp}_0, \ldots, \texttt{resp}_{\kappa-1}\right)$$

of $(\lambda + \kappa\,(4\lambda + 5\log_2(p)))$-bits. We list the expected sizes according to [1,36] in Table 1.

$\log_2(p)$	λ	κ	Security Level	Private key	Public Key	Signature
377	128	219	NIST Level 1	24 B	96 B	66.592 KB
546	192	329	NIST Level 3	35 B	138 B	145.113 KB
697	256	438	NIST Level 5	44 B	176 B	248.816 KB
434	128	219	NIST Level 1	28 B	110 B	74.257 KB
503	160	274	NIST Level 2	32 B	126 B	108.250 KB
610	192	329	NIST Level 3	39 B	154 B	158.273 KB
751	256	438	NIST Level 5	47 B	188 B	261.956 KB

Table 1: Byte sizes. Signature sizes correspond with the average case. Private keys correspond to integer coefficients sk in $\mathbb{Z}_{2^a}$, while public keys are elliptic curves $E\colon y^2 = x^3 + Ax^2 + x$ described by the element A in $\mathbb{F}_{p^2}$. Since $2^a \approx \sqrt{p}$, public keys are 4x larger than private keys.

3.3 To the quadratic twist to reduce sizes

Following B-SIDH construction [14,15], we can still reduce the signature sizes using the quadratic twist curve. For instance, according the parameter sets from [15], we can use primes of 256-bits (NIST Level 1), 384-bits (NIST Level 3), and 512-bits (NIST Level 5). The idea is to choose a prime number p with $M \mid (p+1)$ and $N \mid (p-1)$ being smooth integer numbers close to p and

- replace order-2^a points and 2^a-isogenies by order-M points and M-isogenies, and
- replace order-3^b points and 3^b-isogenies with order-N points and N-isogenies.

On the other hand, [17, Theorem 4] also holds if we repeat κ times the Sigma protocol described in [17, §5.3] and replace 2^a and 3^b with M and N, respectively. It becomes an SHVZK PoK with soundness $(2/3)^\kappa$, assuming the DSPP is computationally hard. Table 2 illustrates the respective signature sizes based on Section 3.2 under the B-SIDH setup [15].

$\log_2(p)$	λ	κ	Security Level	Private key	Public Key	Signature
256	128	219	NIST Level 1	32 B	64 B	49.072 KB
384	192	329	NIST Level 3	48 B	96 B	110.568 KB
512	256	438	NIST Level 5	64 B	128 B	196.256 KB

Table 2: Byte sizes. Signature sizes correspond with the average case. Private keys correspond to integer coefficients sk in $\mathbb{Z}_M$, while public keys are elliptic curves $E\colon y^2 = x^3 + Ax^2 + x$ described by the element A in $\mathbb{F}_{p^2}$. Since $M \approx p$, public keys are 2x larger than private keys.

3.4 To Jacobian of genus-two curves to keep reducing sizes

Following G2SIDH construction [26,33], we have another way to reduce sizes by working with Jacobian of genus two hyperelliptic curves [3]. This time the idea is replace 2^a-isogenies and 3^b-isogenies with $(2^a, 2^a)$-isogenies and $(3^b, 3^b)$-isogenies. One crucial difference between SIDH and G2SIDH is that we do not have only two generators for the torsion subgroups; we have four generators instead, and the isogeny kernels are generated by two elements. For instance, given a public $(2^a, 2^a)$-isogenous Jacobian J_1 to J_0. This time Peggy wants to convince Victor that she knows the secret $(2^a, 2^a)$-isogeny $\phi\colon J_0 \to J_1$, which implies knowing $\ker \phi = \langle K_{\phi,0}, K_{\phi,1}\rangle$. Here, J_0 is a public and fixed Jacobian of a genus two curve H_0, similarly J_1 (the public key) comes from a genus two hyperelliptic curve H_1.

Similarly to Section 3.3, [17, Theorem 4] also extends if we repeat κ times the Sigma protocol described in [17, §5.3] and replace 2^a-isogenies and 3^b-isogenies

---

[3] For a deeper understanding of isogenies in the context of G2SIDH, we strongly suggest reading [26,10,33,32]

with $(2^a, 2^a)$-isogenies and $(3^b, 3^b)$-isogenies, respectively. It becomes an SHVZK PoK with soundness $(2/3)^\kappa$, assuming the G2DSPP (described by Definition 3) is computationally hard.

Definition 3 (Genus two Decisional Supersingular Product Problem (G2DSPP): Alice's case). *Let J_0 be a Jacobian of genus two curve H_0 as in the G2SIDH setting. Given a $(2^a, 2^a)$-isogeny $\phi\colon J_0 \to J_1$ with kernel $\langle K_{\phi,0}, K_{\phi,1}\rangle$, the Genus two Decisional Supersingular Product Problem, labeled as G2DSPP, asks to distinguish between the following two distributions:*

- *(J_2, J_3, ϕ') is the bottom of a random G2SIDH-square. That is, for a randomly chosen order-$(3^b, 3^b)$ kernel $\langle K_{\psi,0}, K_{\psi,1}\rangle$, we have J_2 is the codomain of the $(3^b, 3^b)$-isogeny ψ with kernel $\langle K_{\psi,0}, K_{\psi,1}\rangle$, J_3 is the codomain of the $(3^b, 3^b)$-isogeny ψ' with kernel $\langle \phi(K_{\psi,0}), \phi(K_{\psi,1})\rangle$, and $\phi'\colon J_2 \to J_3$ is the $(2^a, 2^a)$-isogeny with kernel $\langle \psi(K_{\phi,0}), \psi(K_{\phi,1})\rangle$.*
- *(J_2, J_3, ϕ') such that J_2 is a randomly chosen Jacobian with same cardinality as J_0, and $\phi'\colon J_2 \to J_3$ is a random $(2^a, 2^a)$-isogeny with kernel $\langle R_0, R_1\rangle$ for some order-2^a elements $R_0, R_1 \in J_2[2^a]$.*

Essentially, the genus-two recoverable Sigma protocol remains the same flow as in Section 3.2, but we need to consider that it requires double generators and isogeny evaluations. Additionally, we have that the kernel generators of the $(2^a, 2^a)$-isogenies and $(3^b, 3^b)$-isogenies can be expressed by linear combinations determined with three integer coefficients c, d, and e of $\frac{\log_2(p)}{2}$-bits. In summary, we need double of $\frac{\log_2(p)}{2}$-bits integer coefficients to represent $\texttt{com}_2$ and $\texttt{com}_3$, and three coefficients to represent the kernel generators of ϕ', ψ and ψ'. To be more precise, if $\texttt{chall} = 1$, then `resp` has $(4\lambda + 9\log_2(p))$-bits. Otherwise, we have `resp` of $(4\lambda + 7\log_2(p))$-bits. Consequently, in average we get a response `resp` with $\frac{12\lambda + 23\log_2(p)}{3} \approx (4\lambda + 8\log_2(p))$-bits. Since the best algorithm to find an isogeny is $\widetilde{O}(p)$ (classically) and $\widetilde{O}(\sqrt{p})$ (quantumly) [16], we can work with primes of 128 (NIST Level 1), 192 (NIST Leve 3), and 256 (NIST Level 5). Table 3 lists the expected sizes of the signature over Jacobian of genus two curves.

$\log_2(p)$	λ	κ	Security Level	Private key	Public Key	Signature
128	128	219	NIST Level 1	24 B	192 B	42.064 KB
192	192	329	NIST Level 3	36 B	288 B	94.776 KB
256	256	438	NIST Level 5	48 B	384 B	168.224 KB

Table 3: Byte sizes. Signature sizes correspond with the average case. Private keys correspond to 3-tuples of integer coefficients $(\mathsf{sk}_c, \mathsf{sk}_d, \mathsf{sk}_e)$ in $\mathbb{Z}_{2^a}^3$, while public keys are genus two hyperelliptic curves $H\colon y^2 = f(x)$ described by the degree-6 polynomial $f(x)$ over $\mathbb{F}_{p^2}$. Since $2^a \approx \sqrt{p}$, public keys are 8x larger than private keys.

3.5 Size comparisons against isogeny-based signatures

As mentioned in Section 1, the short keys are the most significant selling point of isogeny-based signature construction. In contrast, isogeny construction has a high latency in practice, which seems to be much easier to improve. This section compares state-of-the-art isogeny-based signatures that remain secure against Castryck-Decru family attacks in terms of byte lengths. Currently, there are different families of isogeny-based sigma protocols, such as:

- CSIDH-based: Sea-sign [18,22], CSI-FiSh [6] and the Lossy CSI-FiSh [24];
- SIDH-based: [17, §5.3]; and
- Quaternion-based: SQI-sign [20,21] and [29].

Since all CSIDH-based proposals are initially based over a 512-bits prime field, we compare them by moving into a 2048-bits prime field (as suggested in [8,39,12])). Using a 2048-bits CSIDH-prime impacts public-key sizes and timing efficiency; signature sizes stay fixed as in CSIDH-512. Now, due to the extended variety of CSIDH-based configurations determined by

- the number n of different isogeny degrees,
- the number B of isogenies per isogeny degree, and
- the number S of multiple public-key curves as CSIDH-base public-keys.

We try to englobe a fair comparison assuming $n = 74$, $B = 5$, and $S = 2^6$, which gives a good trade-off between small signature sizes and timings. We used the script from [22] to get sizes concerning the improved Sea-sign over a 2048-bits prime field. Table 4 lists all analyzed isogeny-based signature sizes in bytes.

Scheme	Private key	Public Key	Signature
[29, §4] with Fiat-Shamir transform	32 B	96 B	11.264 KB
Original SQI-sign [20]	16 B	64 B	204 B
SQI-sign improvement from [21]			
Sea-sign [18]	16 B	16.384 KB	720 B
Sea-sign improvement from [22]	16 B	16.128 KB	7.220 KB
Simple variant of CSI-FiSh [6]	16 B	16.384 KB	560 B
Lossy CSI-FiSh [24]	16 B	16.896 KB	560 B
Optimized [17, §5.3] according to Section 3.2	24 B	96 B	66.592 KB
Twist quadratic variant of [17, §5.3] according to Section 3.3	32 B	64 B	49.072 KB
Genus two variant of [17, §5.3] according to Section 3.4	24 B	192 B	42.064 KB

Table 4: Byte sizes concerning state-of-the-art isogeny-based signatures with close to NIST security Level 1. For a fair comparison, we set all CSIDH-based construction in [18,22,6,24] over a 2048-bits prime field (as suggested in [8,39,12]). Large CSIDH primes only impact public-key sizes and timing efficiency; signature sizes stay fixed as in CSIDH-512.

4 Concluding remarks

After the wave of Castryck-Decru attacks, it could be hard to stand for using some isogeny constructions. Therefore, we list all flavors of isogeny-based signatures for which the Castryck-Decru attack does not apply (see Section 3.5). We also estimate the expected optimized sizes for the 3-special soundness Sigma protocol from [17, §5.3] and discuss its extensions on the B-SIDH and G2SIDH context.

Open problems. As pointed out in [17], there is no 2-special soundness construction under the Fixed degree relation: can we construct it for SIDH-squares? A 2-special soundness protocol would considerably reduce sizes and, thus, the number of repetitions (e.g., $\kappa = 128$ instead of 219).

A recent new proposal by LeGrow, Ti, and Zobernig suggests using the supersingular non-superspecial abelian surface [34], where the Costello-Smith attack from [16] does not apply and allows working with 87-bit primes (concerning NIST Security Level 1). Is it possible to build a shorter Sigma protocol using the proposals from [34]?

We learn more from failure than from success. Do not let it stop us. Failure could build new isogeny schemes.

Acknowledgements. We thank Víctor Mateu and Lucas Pandolfo Perin for encouraging us to make this work public. We thank Luca De Feo, Steven D. Galbraith, Jana Sotakova, Kaizhan Lin, and anonymous reviewers for their comments and discussions on [13] that we consider when writing this paper. We thank Andrea Basso for his comments about optimizing sizes when challenge `chall` is one.

References

1. Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalali, A., Jao, D., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Pereira, G., Renes, J., Soukharev, V., Urbanik, D.: Supersingular Isogeny Key Encapsulation. Third Round Candidate of the NIST's post-quantum cryptography standardization process (2020), available at: https://sike.org/
2. Basso, A., Kutas, P., Merz, S., Petit, C., Weitkämper, C.: On Adaptive Attacks Against Jao-Urbanik's Isogeny-Based Protocol. In: Nitaj, A., Youssef, A.M. (eds.) Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12174, pp. 195–213. Springer (2020). https://doi.org/10.1007/978-3-030-51938-4_10, https://doi.org/10.1007/978-3-030-51938-4_10
3. Bernhard, D., Pereira, O., Warinschi, B.: How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In: Wang, X., Sako, K.

(eds.) Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 626–643. Springer (2012). https://doi.org/10.1007/978-3-642-34961-4_38, https://doi.org/10.1007/978-3-642-34961-4_38

4. Beullens, W.: Week 4: Signatures based on SIDH and CSIDH. Isogeny-based cryptography school pp. 1–23 (2021), https://homes.esat.kuleuven.be/~wbeullen/week4_1.pdf, last online access on June 1st, 2022: https://homes.esat.kuleuven.be/~wbeullen/week4_1.pdf
5. Beullens, W., Dobson, S., Katsumata, S., Lai, Y., Pintore, F.: Group Signatures and More from Isogenies and Lattices: Generic, Simple, and Efficient. IACR Cryptol. ePrint Arch. p. 1366 (2021), https://eprint.iacr.org/2021/1366
6. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11921, pp. 227–247. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_9, https://doi.org/10.1007/978-3-030-34578-5_9
7. Biasse, J., Bonnetain, X., Pring, B., Schrottenloher, A., Youmans, W.: A trade-off between classical and quantum circuit size for an attack against CSIDH. Journal of Mathematical Cryptology **15**(1), 4–17 (2021). https://doi.org/10.1515/jmc-2020-0070, https://doi.org/10.1515/jmc-2020-0070
8. Bonnetain, X., Schrottenloher, A.: Quantum Security Analysis of CSIDH. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 493–522. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_17, https://doi.org/10.1007/978-3-030-45724-2_17
9. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH (preliminary version). IACR Cryptol. ePrint Arch. p. 975 (2022), https://eprint.iacr.org/2022/975
10. Castryck, W., Decru, T., Smith, B.: Hash functions from superspecial genus-2 curves using richelot isogenies. J. Math. Cryptol. **14**(1), 268–292 (2020). https://doi.org/10.1515/jmc-2019-0021, https://doi.org/10.1515/jmc-2019-0021
11. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An Efficient Post-Quantum Commutative Group Action. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11274, pp. 395–427. Springer (2018). https://doi.org/10.1007/978-3-030-03332-3_15, https://doi.org/10.1007/978-3-030-03332-3_15
12. Chávez-Saab, J., Chi-Domínguez, J., Jaques, S., Rodríguez-Henríquez, F.: The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. Journal of Cryptographic Engineering (2021). https://doi.org/10.1007/s13389-021-00271-w, https://doi.org/10.1007/s13389-021-00271-w
13. Chi-Domínguez, J., Mateu, V., Perin, L.P.: SIDH-sign: an efficient SIDH PoK-based signature (2022), https://eprint.iacr.org/2022/475

14. Costello, C.: B-SIDH: supersingular isogeny diffie-hellman using twisted torsion. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12492, pp. 440–463. Springer (2020). https://doi.org/10.1007/978-3-030-64834-3_15, https://doi.org/10.1007/978-3-030-64834-3_15
15. Costello, C., Meyer, M., Naehrig, M.: Sieving for twin smooth integers with solutions to the prouhet-tarry-escott problem. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12696, pp. 272–301. Springer (2021). https://doi.org/10.1007/978-3-030-77870-5_10, https://doi.org/10.1007/978-3-030-77870-5_10
16. Costello, C., Smith, B.: The Supersingular Isogeny Problem in Genus 2 and Beyond. In: Ding, J., Tillich, J. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12100, pp. 151–168. Springer (2020). https://doi.org/10.1007/978-3-030-44223-1_9, https://doi.org/10.1007/978-3-030-44223-1_9
17. De Feo, L., Dobson, S., Galbraith, S.D., Zobernig, L.: SIDH Proof of Knowledge. IACR Cryptol. ePrint Arch. p. 1023 (2021), https://eprint.iacr.org/2021/1023, to appear in ASIACRYPT 2022
18. De Feo, L., Galbraith, S.D.: SeaSign: Compact Isogeny Signatures from Class Group Actions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11478, pp. 759–789. Springer (2019). https://doi.org/10.1007/978-3-030-17659-4_26, https://doi.org/10.1007/978-3-030-17659-4_26
19. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology **8**(3), 209–247 (2014). https://doi.org/10.1515/jmc-2012-0015, https://doi.org/10.1515/jmc-2012-0015
20. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 64–93. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_3, https://doi.org/10.1007/978-3-030-64837-4_3
21. De Feo, L., Leroux, A., Wesolowski, B.: New algorithms for the Deuring correspondence: SQISign twice as fast. IACR Cryptol. ePrint Arch. p. 234 (2022), https://eprint.iacr.org/2022/234
22. Decru, T., Panny, L., Vercauteren, F.: Faster SeaSign Signatures Through Improved Rejection Sampling. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers. Lecture Notes in Computer Science, vol. 11505, pp. 271–285. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_15, https://doi.org/10.1007/978-3-030-25510-7_15

23. Dobson, S., Galbraith, S.D., LeGrow, J.T., Ti, Y.B., Zobernig, L.: An adaptive attack on 2-SIDH. International Journal of Computer Mathematics: Computer Systems Theory **5**(4), 282–299 (2020). https://doi.org/10.1080/23799927.2020.1822446, https://doi.org/10.1080/23799927.2020.1822446
24. El Kaafarani, A., Katsumata, S., Pintore, F.: Lossy CSI-FiSh: Efficient Signature Scheme with Tight Reduction to Decisional CSIDH-512. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12111, pp. 157–186. Springer (2020). https://doi.org/10.1007/978-3-030-45388-6_6, https://doi.org/10.1007/978-3-030-45388-6_6
25. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer (1986). https://doi.org/10.1007/3-540-47721-7_12, https://doi.org/10.1007/3-540-47721-7_12
26. Flynn, E.V., Ti, Y.B.: Genus two isogeny cryptography. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers. Lecture Notes in Computer Science, vol. 11505, pp. 286–306. Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_16, https://doi.org/10.1007/978-3-030-25510-7_16
27. Fouotsa, T.B., Petit, C.: A New Adaptive Attack on SIDH. In: Galbraith, S.D. (ed.) Topics in Cryptology - CT-RSA 2022 - Cryptographers' Track at the RSA Conference 2022, Virtual Event, March 1-2, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13161, pp. 322–344. Springer (2022). https://doi.org/10.1007/978-3-030-95312-6_14, https://doi.org/10.1007/978-3-030-95312-6_14
28. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the Security of Supersingular Isogeny Cryptosystems. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 63–91 (2016). https://doi.org/10.1007/978-3-662-53887-6_3, https://doi.org/10.1007/978-3-662-53887-6_3
29. Galbraith, S.D., Petit, C., Silva, J.: Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. Journal of Cryptology **33**(1), 130–175 (2020). https://doi.org/10.1007/s00145-019-09316-0, https://doi.org/10.1007/s00145-019-09316-0
30. Ghantous, W., Katsumata, S., Pintore, F., Veroni, M.: Collisions in Supersingular Isogeny Graphs and the SIDH-based Identification Protocol. IACR Cryptol. ePrint Arch. p. 1051 (2021), https://eprint.iacr.org/2021/1051
31. Jao, D., De Feo, L.: Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In: Yang, B. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7071, pp. 19–34. Springer (2011). https://doi.org/10.1007/978-3-642-25405-5_2, https://doi.org/10.1007/978-3-642-25405-5_2

32. Kunzweiler, S.: Efficient Computation of $(2^n, 2^n)$-Isogenies (2022), https://eprint.iacr.org/2022/990
33. Kunzweiler, S., Ti, Y.B., Weitkämper, C.: Secret keys in genus-2 SIDH. In: AlTawy, R., Hülsing, A. (eds.) Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13203, pp. 483–507. Springer (2021). https://doi.org/10.1007/978-3-030-99277-4_23, https://doi.org/10.1007/978-3-030-99277-4_23
34. LeGrow, J.T., Ti, Y.B., Zobernig, L.: Supersingular Non-Superspecial Abelian Surfaces in Cryptography (2022), https://eprint.iacr.org/2022/650
35. Leonardi, C.: A note on the ending elliptic curve in SIDH. IACR Cryptol. ePrint Arch. p. 262 (2020), https://eprint.iacr.org/2020/262
36. Longa, P.: Efficient Algorithms for Large Prime Characteristic Fields and Their Application to Bilinear Pairings and Supersingular Isogeny-Based Protocols. IACR Cryptol. ePrint Arch. p. 367 (2022), https://ia.cr/2022/367
37. Maino, L., Martindale, C.: An attack on SIDH with arbitrary starting curve. IACR Cryptol. ePrint Arch. p. 1026 (2022), https://eprint.iacr.org/2022/1026
38. Oudompheng, R., Pope, G.: A Note on Reimplementing the Castryck-Decru Attack and Lessons Learned for SageMath (2022), https://eprint.iacr.org/2022/1283
39. Peikert, C.: He Gives C-Sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 463–492. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_16, https://doi.org/10.1007/978-3-030-45724-2_16
40. Robert, D.: Breaking SIDH in polynomial time. IACR Cryptol. ePrint Arch. p. 1038 (2022), https://eprint.iacr.org/2022/1038
41. Stolbunov, A.: Cryptographic Schemes Based on Isogenies. Ph.D. thesis, Norwegian University of Science and Technology Faculty of Information Technology, Mathematics and Electrical Engineering Department of Telematics (01 2012). https://doi.org/10.13140/RG.2.2.20826.44488
42. Unruh, D.: Post-quantum Security of Fiat-Shamir. In: International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT) (2017). https://doi.org/10.1007/978-3-319-70694-8_3
43. Yoo, Y., Azarderakhsh, R., Jalali, A., Jao, D., Soukharev, V.: A Post-quantum Digital Signature Scheme Based on Supersingular Isogenies. In: Kiayias, A. (ed.) Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10322, pp. 163–181. Springer (2017). https://doi.org/10.1007/978-3-319-70972-7_9, https://doi.org/10.1007/978-3-319-70972-7_9