SQISignHD: signing with higher dimensional isogenies

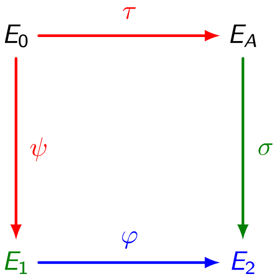
Pierrick Dartois

Joint work with Antonin Leroux, Damien Robert and Benjamin Wesolowski
Acknowledgements to Luca De Feo

28 march 2023

SQISign

The SQISign identification scheme (DFKLPW, 2020)

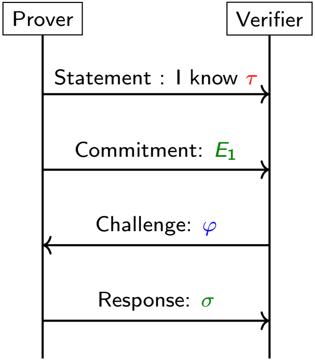


public

Prover's secret

published by Verifier

published by Prover



Accept if

$\widehat{\varphi} \circ \sigma$ is cyclic

Fiat-shamir transform

$$
\begin{tikzcd}
E_0 \arrow[r, "\tau"] \arrow[d, "\psi"'] & E_A \arrow[d, "\sigma"] \\
E_1 \arrow[r, "\varphi"] & E_2
\end{tikzcd}
$$

public

Signer's secret

published by Signer

Signature: message m, public key E_A, secret key τ.

- Commitment $\psi : E_0 \longrightarrow E_1$.
- Challenge $\varphi := H(E_1, m)$ (where H is a hash function).
- Compute and send signature (E_1, σ) to the verifier.

Fiat-shamir transform

$$\begin{tikzcd} E_0 \arrow[r, "\tau"] \arrow[d, "\psi"'] & E_A \arrow[d, "\sigma"] \\ E_1 \arrow[r, "\varphi"] & E_2 \end{tikzcd}$$

public
Signer's secret
published by Signer

Signature: message m, public key E_A, secret key τ.

- Commitment $\psi : E_0 \longrightarrow E_1$.
- Challenge $\varphi := H(E_1, m)$ (where H is a hash function).
- Compute and send signature (E_1, σ) to the verifier.

Verification: $E_A, m, (E_1, \sigma)$.

- Recompute $\varphi := H(E_1, m)$.
- Verify that $\widehat{\varphi} \circ \sigma$ is cyclic.

NB: Not necessary to send E_1.

How to compute the signature σ ?

The Deuring correspondence

Supersingular elliptic curves	Quaternions
$j(E)$ or $j(E)^p$ supersingular	$\mathcal{O} \cong \mathrm{End}(E)$ maximal order in $\mathcal{B}_{p,\infty}$
$\varphi : E \longrightarrow E'$	left $\mathcal{O}$-ideal and right $\mathcal{O}'$-ideal I_φ
$\varphi, \psi : E \longrightarrow E'$	$I_\varphi \sim I_\psi$ $(\exists \alpha \in \mathcal{B}_{p,\infty},\ I_\psi = I_\varphi \alpha)$
$\widehat{\varphi}$	$\overline{I_\varphi}$
$\varphi \circ \psi$	$I_\psi \cdot I_\varphi$
$\theta \in \mathrm{End}(E)$	Principal ideal $\mathcal{O}\theta$
$\deg(\varphi)$	$\mathrm{nrd}(I_\varphi)$

How to compute the signature σ ?

$$\begin{tikzcd}
E_0 \arrow[r, "\tau", "I_\tau"'] \arrow[d, "I_\psi"', "\psi"] & E_A \arrow[d, dashed, "I"', "\sigma"] \\
E_1 \arrow[r, "\varphi", "I_\varphi"'] & E_2
\end{tikzcd}$$

- Compute $J := \overline{I_\tau} \cdot I_\psi \cdot I_\varphi$.
- Find $I \sim J$ of norm $\mathrm{nrd}(I) = \ell^e$ (KLPT, 2014 & FKLPW, 2020).
- Compute σ associated to I.

How to compute the signature σ ?

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
E_0 \arrow[r, "\tau", "I_\tau"'] \arrow[d, "I_\psi"', "\psi"] & E_A \arrow[d, dashed, "I"', "\sigma"] \\
E_1 \arrow[r, "\varphi", "I_\varphi"'] & E_2
\end{tikzcd}
$$

- Compute $J := \overline{I_\tau} \cdot I_\psi \cdot I_\varphi$.
- Find $I \sim J$ of norm $\mathrm{nrd}(I) = \ell^e$ (KLPT, 2014 & FKLPW, 2020).
- Compute σ associated to I.

Ideal-to-isogeny translation:

- $\mathrm{nrd}(I) = \deg(\sigma) = p^{15/4}$ (DFLW, 2022).
- Not enough $\ell^\bullet$-torsion accessible.
- σ is computed piecewise (DFLW, 2022). **It is slow.**

Representing an isogeny in higher dimension

Efficient isogeny representation

Definition

An **efficient representation** of an isogeny $\varphi : E \longrightarrow E'$ is a couple $(D, \mathscr{A})$, where:

1. D is polynomial size data (in $\log(p)$, $\log(\deg(\varphi))$).
2. $\mathscr{A}$ is an algorithm computing $\varphi(P)$ on input D and $P \in E(\mathbb{F}_{p^k})$ in polynomial time (in $\log(p)$, $\log(\deg(\varphi))$ and k).

Efficient isogeny representation

Example (efficient representation): If $\varphi : E \longrightarrow E'$ is an ℓ^e-isogeny:

- D: chain of ℓ-isogenies $E = E_0 \xrightarrow{\varphi_0} E_1 \cdots E_{e-1} \xrightarrow{\varphi_{e-1}} E_e = E'$.
- $\mathscr{A}$: evaluate each isogeny of the chain successively.

Efficient isogeny representation

Example (efficient representation): If $\varphi : E \longrightarrow E'$ is an ℓ^e-isogeny:

- D: chain of ℓ-isogenies $E = E_0 \xrightarrow{\varphi_0} E_1 \cdots E_{e-1} \xrightarrow{\varphi_{e-1}} E_e = E'$.
- $\mathscr{A}$: evaluate each isogeny of the chain successively.

State of the art :

- All efficient representations are essentially equivalent to this one.
- Only smooth degree isogenies can be represented (explains the use of KLPT in SQISign).
- In SQISign, the conversion of an ideal I into such a representation (isogeny-chain) is costly.

d-isogenies

- Let $\varphi : (A, \lambda_A) \longrightarrow (B, \lambda_B)$ be an isogeny between PPAV[1].
- Consider $\widetilde{\varphi} : (B, \lambda_A) \longrightarrow (A, \lambda_B)$ the isogeny

$$B \xrightarrow{\lambda_B} \widehat{B} \xrightarrow{\widehat{\varphi}} \widehat{A} \xrightarrow{\lambda_A^{-1}} A$$

[1] Principally polarized abelian varieties.

d-isogenies

- Let $\varphi : (A, \lambda_A) \longrightarrow (B, \lambda_B)$ be an isogeny between PPAV[1].
- Consider $\widetilde{\varphi} : (B, \lambda_A) \longrightarrow (A, \lambda_B)$ the isogeny

$$B \xrightarrow{\lambda_B} \widehat{B} \xrightarrow{\widehat{\varphi}} \widehat{A} \xrightarrow{\lambda_A^{-1}} A$$

Definition

φ is a d-isogeny if $\widetilde{\varphi} \circ \varphi = [d]_A$ or equivalently $\varphi \circ \widetilde{\varphi} = [d]_B$.

[1] Principally polarized abelian varieties.

d-isogenies

- Let $\varphi : (A, \lambda_A) \longrightarrow (B, \lambda_B)$ be an isogeny between PPAV[1].
- Consider $\widetilde{\varphi} : (B, \lambda_A) \longrightarrow (A, \lambda_B)$ the isogeny

$$B \xrightarrow{\lambda_B} \widehat{B} \xrightarrow{\widehat{\varphi}} \widehat{A} \xrightarrow{\lambda_A^{-1}} A$$

Definition

φ is a d-isogeny if $\widetilde{\varphi} \circ \varphi = [d]_A$ or equivalently $\varphi \circ \widetilde{\varphi} = [d]_B$.

Lemma

If $d = \ell^e$, then φ can be decomposed as a product of ℓ-isogenies

$$A = A_0 \xrightarrow{\varphi_0} A_1 \xrightarrow{\varphi_1} \cdots \xrightarrow{\varphi_{e-2}} A_{e-1} \xrightarrow{\varphi_{e-1}} A_e = B$$

[1] Principally polarized abelian varieties.

Kani's lemma (K, 1997)

An (a, b)-**isogeny diamond** is a commutative diagram:

$$\begin{tikzcd} A' \arrow[r, "\varphi'"] & B' \\ A \arrow[u, "\psi"] \arrow[r, "\varphi"] & B \arrow[u, "\psi'"'] \end{tikzcd}$$

with:

- $\deg(\varphi) = \deg(\varphi') = a$.
- $\deg(\psi) = \deg(\psi') = b$.

Lemma (K, 1997)

Kani's lemma (K, 1997)

An (a, b)-**isogeny diamond** is a commutative diagram:

$$\begin{tikzcd}
A' \arrow[r, "\varphi'"] & B' \\
A \arrow[u, "\psi"] \arrow[r, "\varphi"] & B \arrow[u, "\psi'"']
\end{tikzcd}$$

with:

- $\deg(\varphi) = \deg(\varphi') = a$.
- $\deg(\psi) = \deg(\psi') = b$.

Lemma (K, 1997)

(i) *Assume a and b coprime with p. Then*

$$F := \begin{pmatrix} \varphi & \widetilde{\psi'} \\ -\psi & \widetilde{\varphi'} \end{pmatrix} : A \times B' \longrightarrow B \times A'$$

is a d-isogeny (with $d := a + b$).

Kani's lemma (K, 1997)

An (a, b)-**isogeny diamond** is a commutative diagram:

$$\begin{tikzcd}
A' \arrow[r, "\varphi'"] & B' \\
A \arrow[u, "\psi"'] \arrow[r, "\varphi"] & B \arrow[u, "\psi'"']
\end{tikzcd}$$

with:

- $\deg(\varphi) = \deg(\varphi') = a$.
- $\deg(\psi) = \deg(\psi') = b$.

Lemma (K, 1997)

(i) *Assume a and b coprime with p. Then*

$$F := \begin{pmatrix} \varphi & \widetilde{\psi'} \\ -\psi & \widetilde{\varphi'} \end{pmatrix} : A \times B' \longrightarrow B \times A'$$

is a d-isogeny (with $d := a + b$).
(ii) *If $a \wedge b = 1$, we have*

$$\ker(F) = \{(\widetilde{\varphi}(x), \psi'(x)) \mid x \in B[d]\}.$$

Representation in dimension 4

We can now represent isogenies of non-smooth degrees!

Let:

- $\sigma : E_A \longrightarrow E_2$ of degree $q < \ell^e$.
- $\Sigma := \mathrm{Diag}(\sigma, \sigma) : E_A^2 \longrightarrow E_2^2$.
- $a_1, a_2 \in \mathbb{Z}$, s.t.

$$a_1^2 + a_2^2 + q = \ell^e.$$

- $\alpha \in \mathrm{End}(E_A^2)$ given by:

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

- and $\alpha' \in \mathrm{End}(E_2^2)$ defined as α.

Representation in dimension 4

We can now represent isogenies of non-smooth degrees!

Let:

- $\sigma : E_A \longrightarrow E_2$ of degree $q < \ell^e$.
- $\Sigma := \mathrm{Diag}(\sigma, \sigma) : E_A^2 \longrightarrow E_2^2$.
- $a_1, a_2 \in \mathbb{Z}$, s.t.

$$a_1^2 + a_2^2 + q = \ell^e.$$

- $\alpha \in \mathrm{End}(E_A^2)$ given by:

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

- and $\alpha' \in \mathrm{End}(E_2^2)$ defined as α.

The $(q, a_1^2 + a_2^2)$-isogeny diamond

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

yields an ℓ^e-isogeny

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha'} \end{pmatrix} \in \mathrm{End}(E_A^2 \times E_2^2).$$

Representation in dimension 4

We can now represent isogenies of non-smooth degrees!

The isogeny diamond

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

yields an ℓ^e-isogeny

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix} \in \mathrm{End}(E_A^2 \times E_2^2).$$

- We have

$$\ker(F) = \{(\widetilde{\alpha}(P), \Sigma(P)) \mid P \in E_A^2[\ell^e]\}.$$

Representation in dimension 4

We can now represent isogenies of non-smooth degrees!

The isogeny diamond

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

yields an ℓ^e-isogeny

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha'} \end{pmatrix} \in \mathrm{End}(E_A^2 \times E_2^2).$$

- We have
$$\ker(F) = \{(\widetilde{\alpha}(P), \Sigma(P)) \mid P \in E_A^2[\ell^e]\}.$$
- It suffices to compute $\sigma(E_A[\ell^e])$ to compute $\ker(F)$.
- F can then be computed as a chain of ℓ-isogenies.
- Knowing F, we can evaluate σ everywhere.

Representation in dimension 4

The bad news: We have to compute isogenies in dimension 4.

Representation in dimension 4

The bad news: We have to compute isogenies in dimension 4.

The good news:

- Much more freedom on $q = \deg(\sigma)$.
- Smaller degree $q = \mathrm{nrd}(I) \simeq \sqrt{p}$.
- Recall that $q = \ell^e \simeq p^{15/4}$ in SQISign.

Representation in dimension 4

The bad news: We have to compute isogenies in dimension 4.

The good news:

- Much more freedom on $q = \deg(\sigma)$.
- Smaller degree $q = \mathrm{nrd}(I) \simeq \sqrt{p}$.
- Recall that $q = \ell^e \simeq p^{15/4}$ in SQISign.

How much freedom on the choice of q?

- Constraint: we can find $a_1, a_2 \in \mathbb{Z}$ such that $a_1^2 + a_2^2 + q = \ell^e$.
- In practice: $\ell^e - q$ is a prime $\equiv 1 \mod 4$.

Representation in dimension 8

Representing σ of any degree $q < \ell^e$

- Find $a_1, a_2, a_3, a_4 \in \mathbb{Z}$ such that

$$a_1^2 + a_2^2 + a_3^2 + a_4^2 + q = \ell^e.$$

Representation in dimension 8

Representing σ of any degree $q < \ell^e$

- Find $a_1, a_2, a_3, a_4 \in \mathbb{Z}$ such that

$$a_1^2 + a_2^2 + a_3^2 + a_4^2 + q = \ell^e.$$

- Let $\Sigma := \mathrm{Diag}(\sigma, \cdots, \sigma) : E_A^4 \longrightarrow E_2^4$,

$$\alpha := \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix} \in \mathrm{End}(E_A^4)$$

  and $\alpha' \in \mathrm{End}(E_2^4)$ defined as α.

- Instead, we can represent σ with the ℓ^e-isogeny

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix} \in \mathrm{End}(E_A^4 \times E_2^4).$$

Algorithms for response and verification

The response algorithm

Overview of the response algorithm

$$\begin{tikzcd}[column sep=huge, row sep=huge]
E_0 \arrow[r, "\tau", "I_\tau"'] \arrow[d, shift right=3, "\psi' \; I_{\psi'}"'] \arrow[d, shift left=3, "\psi \; I_\psi"] & E_A \arrow[d, dashed, "I"', "\sigma"] \\
E_1 \arrow[r, "\varphi", "I_\varphi"'] & E_2
\end{tikzcd}$$

- Compute $J := \overline{I_\tau} \cdot I_\psi \cdot I_\varphi$.
- Find $I \sim J$ random of norm $q < \ell^e$ s.t. $\ell^e - q$ is prime $\equiv 1 \mod 4$.

The response algorithm

Overview of the response algorithm

$$\begin{tikzcd}[row sep=huge, column sep=huge]
E_0 \arrow[r, "\tau", "I_\tau"'] \arrow[d, shift right=2, "\psi'"', "I_{\psi'}"] \arrow[d, shift left=2, "\psi", "I_\psi"'] & E_A \arrow[d, dashed, "I"', "\sigma"] \\
E_1 \arrow[r, "\varphi", "I_\varphi"'] & E_2
\end{tikzcd}$$

- Compute $J := \overline{I_\tau} \cdot I_\psi \cdot I_\varphi$.
- Find $I \sim J$ <u>random</u> of norm $q < \ell^e$ s.t. $\ell^e - q$ is prime $\equiv 1 \mod 4$.
- Generate (P_1, P_2) a canonical basis of $E_A[\ell^e]$.
- Compute $(\sigma(P_1), \sigma(P_2))$ using $\varphi \circ \psi \circ \widehat{\tau}$.
- Send $(q, \sigma(P_1), \sigma(P_2))$.

The response algorithm

Evaluating σ

$$\begin{tikzcd}
E_0 \arrow[rr, "\tau", "I_\tau"'] \arrow[d, shift right=3, "I_{\psi'}"', "\psi'"] \arrow[d, "\psi", "I_\psi"'] & & E_A \arrow[d, dashed, "I"', "\sigma"] \\
E_1 \arrow[rr, "\varphi", "I_\varphi"'] & & E_2
\end{tikzcd}$$

- Let $\gamma := \widehat{\psi} \circ \widehat{\varphi} \circ \sigma \circ \tau \in \mathrm{End}(E_0)$.
- We have $\mathcal{O}_0\gamma = I_\tau \cdot I \cdot \overline{I_\varphi} \cdot \overline{I_\psi}$ so we can compute γ.

The response algorithm

Evaluating σ

$$\begin{tikzcd}
E_0 \arrow[r, "\tau", "I_\tau"'] \arrow[d, shift right=2, "\psi'"', "I_{\psi'}"'] \arrow[d, shift left=2, "\psi", "I_\psi"] & E_A \arrow[d, dashed, "I"', "\sigma"] \\
E_1 \arrow[r, "\varphi", "I_\varphi"'] & E_2
\end{tikzcd}$$

- Let $\gamma := \widehat{\psi} \circ \widehat{\varphi} \circ \sigma \circ \tau \in \mathrm{End}(E_0)$.
- We have $\mathcal{O}_0\gamma = I_\tau \cdot I \cdot \overline{I_\varphi} \cdot \overline{I_\psi}$ so we can compute γ.
- Then:

$$[D_\psi D_\varphi D_\tau]\sigma = \varphi \circ \psi \circ \gamma \circ \widehat{\tau}$$

The response algorithm

Evaluating σ

$$\begin{tikzcd}
E_0 \arrow[r, "\tau", "I_\tau"'] \arrow[d, shift right=2, "\psi' \; I_{\psi'}"'] \arrow[d, shift left=2, "\psi \; I_\psi"] & E_A \arrow[d, dashed, "\sigma", "I"'] \\
E_1 \arrow[r, "\varphi", "I_\varphi"'] & E_2
\end{tikzcd}$$

- Let $\gamma := \widehat{\psi} \circ \widehat{\varphi} \circ \sigma \circ \tau \in \mathrm{End}(E_0)$.
- We have $\mathcal{O}_0\gamma = I_\tau \cdot I \cdot \overline{I_\varphi} \cdot \overline{I_\psi}$ so we can compute γ.
- Then:

$$[D_\psi D_\varphi D_\tau]\sigma = \varphi \circ \psi \circ \gamma \circ \widehat{\tau}$$

- We can evaluate σ on $P \in E_A[\ell^e]$ provided $(D_\psi D_\varphi D_\tau) \wedge \ell = 1$:

$$\sigma(P) = [\lambda]\varphi \circ \psi \circ \gamma \circ \widehat{\tau}(P),$$

with $\lambda D_\psi D_\varphi D_\tau \equiv 1 \mod \ell^e$.

The response algorithm

Evaluating σ

$$\begin{tikzcd}
E_0 \arrow[rr, "\tau", "I_\tau"'] \arrow[d, shift right=2, "I_{\psi'}"', "\psi'"] \arrow[d, shift left=2, "\psi", "I_\psi"'] & & E_A \arrow[d, dashed, "\sigma", "I"'] \\
E_1 \arrow[rr, "\varphi", "I_\varphi"'] & & E_2
\end{tikzcd}$$

- Let $\gamma := \widehat{\psi} \circ \widehat{\varphi} \circ \sigma \circ \tau \in \mathrm{End}(E_0)$.
- We have $\mathcal{O}_0\gamma = I_\tau \cdot I \cdot \overline{I_\varphi} \cdot \overline{I_\psi}$ so we can compute γ.
- Then:

$$[D_\psi D_\varphi D_\tau]\sigma = \varphi \circ \psi \circ \gamma \circ \widehat{\tau}$$

- We can evaluate σ on $P \in E_A[\ell^e]$ provided $(D_\psi D_\varphi D_\tau) \wedge \ell = 1$:

$$\sigma(P) = [\lambda]\varphi \circ \psi \circ \gamma \circ \widehat{\tau}(P),$$

with $\lambda D_\psi D_\varphi D_\tau \equiv 1 \mod \ell^e$.

- In SQISignHD, D_ψ, D_φ and D_τ are powers of a prime $\ell' \neq \ell$.

Compute the challenge ideal

Goal: Given $\varphi : E_1 \longrightarrow E_2$ of degree $D_\varphi = \ell'^\bullet$, compute I_φ.

Compute the challenge ideal

Goal: Given $\varphi : E_1 \longrightarrow E_2$ of degree $D_\varphi = \ell'^\bullet$, compute I_φ.

Step 1: Compute a basis $\mathcal{B}_1$ of $\mathcal{O}_1 \cong \mathrm{End}(E_1)$ that can be evaluated on $E_1[D_\varphi]$.

- We know a basis $\mathcal{B}_0$ of $\mathcal{O}_0 \cong \mathrm{End}(E_0)$.
- Push it through $\psi' : E_0 \longrightarrow E_1$ ($D_{\psi'} = \ell^\bullet$ coprime with $D_\varphi = \ell'^\bullet$).

Compute the challenge ideal

Goal: Given $\varphi : E_1 \longrightarrow E_2$ of degree $D_\varphi = \ell'^\bullet$, compute I_φ.

Step 1: Compute a basis $\mathcal{B}_1$ of $\mathcal{O}_1 \cong \mathrm{End}(E_1)$ that can be evaluated on $E_1[D_\varphi]$.

- We know a basis $\mathcal{B}_0$ of $\mathcal{O}_0 \cong \mathrm{End}(E_0)$.
- Push it through $\psi' : E_0 \longrightarrow E_1$ ($D_{\psi'} = \ell^\bullet$ coprime with $D_\varphi = \ell'^\bullet$).

Why do we need two paths $\psi, \psi' : E_0 \longrightarrow E_1$? Because $D_\psi = \ell'^\bullet$ is not coprime with $D_\varphi = \ell'^\bullet$.

Compute the challenge ideal

Goal: Given $\varphi : E_1 \longrightarrow E_2$ of degree $D_\varphi = \ell'^\bullet$, compute I_φ.

Step 1: Compute a basis $\mathcal{B}_1$ of $\mathcal{O}_1 \cong \mathrm{End}(E_1)$ that can be evaluated on $E_1[D_\varphi]$.

- We know a basis $\mathcal{B}_0$ of $\mathcal{O}_0 \cong \mathrm{End}(E_0)$.
- Push it through $\psi' : E_0 \longrightarrow E_1$ ($D_{\psi'} = \ell^\bullet$ coprime with $D_\varphi = \ell'^\bullet$).

Why do we need two paths $\psi, \psi' : E_0 \longrightarrow E_1$ **?** Because $D_\psi = \ell'^\bullet$ is not coprime with $D_\varphi = \ell'^\bullet$.

Step 2: Evaluate $\mathcal{B}_1$ on $\ker(\varphi)$ and solve DLPs to find a basis of I_φ.

Compute the challenge ideal

Goal: Given $\varphi : E_1 \longrightarrow E_2$ of degree $D_\varphi = \ell'^{\bullet}$, compute I_φ.

Step 1: Compute a basis $\mathcal{B}_1$ of $\mathcal{O}_1 \cong \mathrm{End}(E_1)$ that can be evaluated on $E_1[D_\varphi]$.

- We know a basis $\mathcal{B}_0$ of $\mathcal{O}_0 \cong \mathrm{End}(E_0)$.
- Push it through $\psi' : E_0 \longrightarrow E_1$ ($D_{\psi'} = \ell^{\bullet}$ coprime with $D_\varphi = \ell'^{\bullet}$).

Why do we need two paths $\psi, \psi' : E_0 \longrightarrow E_1$ **?** Because $D_\psi = \ell'^{\bullet}$ is not coprime with $D_\varphi = \ell'^{\bullet}$.

Step 2: Evaluate $\mathcal{B}_1$ on $\ker(\varphi)$ and solve DLPs to find a basis of I_φ.

- Let $\mathcal{B}_1 := (\beta_1, \cdots, \beta_4)$ and $\ker(\varphi) := \langle P \rangle$ (assuming φ cyclic).
- Compute $\beta_i(P)$ for $1 \leq i \leq 4$.
- Find i, j such that $(\beta_i(P), \beta_j(P))$ generate $E_1[D_\varphi]$.
- Let $k \neq i, j$. Find $a, b \in \mathbb{Z}/D_\varphi\mathbb{Z}$ s.t. $\beta_k(P) = a\beta_i(P) + b\beta_j(P)$.
- Let $\gamma := \beta_k - a\beta_i - b\beta_j$ and return $I_\varphi := \mathcal{O}_1\gamma + \mathcal{O}_1 D_\varphi$.

The choice of prime p

As in SIDH: $p = c\ell^f \ell'^{f'} - 1$ with $\ell^f \simeq \ell'^{f'} \simeq \sqrt{p}$.

The choice of prime p

As in SIDH: $p = c\ell^f \ell'^{f'} - 1$ with $\ell^f \simeq \ell'^{f'} \simeq \sqrt{p}$.

Size of p**:** $p = \Theta(2^{2\lambda})$ for λ bits of security (DG, 2016).

Example: $p = 2^{128}3^{81} - 1$ for NIST-1 level of security.

The choice of prime p

As in SIDH: $p = c\ell^f \ell'^{f'} - 1$ with $\ell^f \simeq \ell'^{f'} \simeq \sqrt{p}$.

Size of p: $p = \Theta(2^{2\lambda})$ for λ bits of security (DG, 2016).

Example: $p = 2^{128}3^{81} - 1$ for NIST-1 level of security.

$$\begin{tikzcd}[row sep=huge, column sep=huge]
E_0 \arrow[r, "\tau", "D_\tau"'] \arrow[d, shift right=2, "\psi'"', "D_{\psi'}"'] \arrow[d, shift left=2, "\psi", "D_\psi"'] & E_A \arrow[d, dashed, "q"', "\sigma"] \\
E_1 \arrow[r, "\varphi", "D_\varphi"'] & E_2
\end{tikzcd}$$

Degree choices:

- $D_\psi = D_\tau = \ell'^{2f'} \simeq p$.
- $D_{\psi'} = \ell^{2f} \simeq p$.
- $D_\varphi = \ell'^{f'} \simeq \sqrt{p}$.

The verification algorithm

Notations:

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

and idem for α'. $\Sigma := \mathrm{Diag}(\sigma, \sigma)$

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix}$$

Entry: $(q, \sigma(P_1), \sigma(P_2))$.

The verification algorithm

Notations:

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

and idem for α'. $\Sigma := \mathrm{Diag}(\sigma, \sigma)$

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix}$$

Entry: $(q, \sigma(P_1), \sigma(P_2))$.

- Cornacchia: find $a_1, a_2 \in \mathbb{Z}$ s.t.

$$a_1^2 + a_2^2 + q = \ell^e.$$

The verification algorithm

Notations:

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

and idem for α'. $\Sigma := \mathrm{Diag}(\sigma, \sigma)$

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix}$$

Entry: $(q, \sigma(P_1), \sigma(P_2))$.

- Cornacchia: find $a_1, a_2 \in \mathbb{Z}$ s.t.

$$a_1^2 + a_2^2 + q = \ell^e.$$

- Generate (P_1, P_2) and compute

$$\ker(F) := \langle(\widetilde{\alpha}(P_i, P_j), \Sigma(P_i, P_j))\rangle_{i,j}$$

The verification algorithm

Notations:

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

and idem for α'. $\Sigma := \mathrm{Diag}(\sigma, \sigma)$

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix}$$

Entry: $(q, \sigma(P_1), \sigma(P_2))$.

- Cornacchia: find $a_1, a_2 \in \mathbb{Z}$ s.t.

$$a_1^2 + a_2^2 + q = \ell^e.$$

- Generate (P_1, P_2) and compute

$$\ker(F) := \langle (\widetilde{\alpha}(P_i, P_j), \Sigma(P_i, P_j)) \rangle_{i,j}$$

- Compute F as an ℓ-isogeny chain

$$\mathcal{A}_0 \xrightarrow{F_1} \mathcal{A}_1 \cdots \mathcal{A}_{e-1} \xrightarrow{F_e} \mathcal{A}_e$$

with $\mathcal{A}_0 := E_2^2 \times E_A^2$.

The verification algorithm

Notations:

$$\begin{tikzcd} E_2^2 \arrow[r, "\alpha'"] & E_2^2 \\ E_A^2 \arrow[u, "\Sigma"] \arrow[r, "\alpha"] & E_A^2 \arrow[u, "\Sigma"'] \end{tikzcd}$$

$$\alpha := \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$$

and idem for α'. $\Sigma := \mathrm{Diag}(\sigma, \sigma)$

$$F := \begin{pmatrix} \alpha & \widetilde{\Sigma} \\ -\Sigma & \widetilde{\alpha}' \end{pmatrix}$$

Entry: $(q, \sigma(P_1), \sigma(P_2))$.

- Cornacchia: find $a_1, a_2 \in \mathbb{Z}$ s.t.

$$a_1^2 + a_2^2 + q = \ell^e.$$

- Generate (P_1, P_2) and compute

$$\ker(F) := \langle (\widetilde{\alpha}(P_i, P_j), \Sigma(P_i, P_j)) \rangle_{i,j}$$

- Compute F as an ℓ-isogeny chain

$$\mathcal{A}_0 \xrightarrow{F_1} \mathcal{A}_1 \cdots \mathcal{A}_{e-1} \xrightarrow{F_e} \mathcal{A}_e$$

with $\mathcal{A}_0 := E_2^2 \times E_A^2$.

- Accept if $\mathcal{A}_e = \mathcal{A}_0$.

Verifying with less torsion

- We can divide F into

$$\mathcal{A} \xrightarrow{F_1} \mathcal{B} \xrightarrow{F_2} \mathcal{A}$$

where $\mathcal{A} := E_2^2 \times E_A^2$, F_i is an ℓ^{e_i}-isogeny (for $i = 1, 2$) and $e := e_1 + e_2$.

Verifying with less torsion

- We can divide F into

$$\mathcal{A} \xrightarrow{F_1} \mathcal{B} \xrightarrow{F_2} \mathcal{A}$$

where $\mathcal{A} := E_2^2 \times E_A^2$, F_i is an ℓ^{e_i}-isogeny (for $i = 1, 2$) and $e := e_1 + e_2$.

- $\ker(F_1) = \ker(F) \cap \mathcal{A}[\ell^{e_1}]$ and $\ker(\widetilde{F_2}) = F(\mathcal{A}[\ell^{e_2}])$.

Verifying with less torsion

- We can divide F into

$$\mathcal{A} \xrightarrow{F_1} \mathcal{B} \xrightarrow{F_2} \mathcal{A}$$

  where $\mathcal{A} := E_2^2 \times E_A^2$, F_i is an ℓ^{e_i}-isogeny (for $i = 1, 2$) and $e := e_1 + e_2$.
- $\ker(F_1) = \ker(F) \cap \mathcal{A}[\ell^{e_1}]$ and $\ker(\widetilde{F_2}) = F(\mathcal{A}[\ell^{e_2}])$.
- Let (P_1, P_2) be a basis of $E_A[\ell^{f_1}]$ with $f \geq f_1 \geq \max(e_1, e_2)$.

Verifying with less torsion

- We can divide F into

$$\mathcal{A} \xrightarrow{F_1} \mathcal{B} \xrightarrow{F_2} \mathcal{A}$$

  where $\mathcal{A} := E_2^2 \times E_A^2$, F_i is an ℓ^{e_i}-isogeny (for $i = 1, 2$) and $e := e_1 + e_2$.
- $\ker(F_1) = \ker(F) \cap \mathcal{A}[\ell^{e_1}]$ and $\ker(\widetilde{F_2}) = F(\mathcal{A}[\ell^{e_2}])$.
- Let (P_1, P_2) be a basis of $E_A[\ell^{f_1}]$ with $f \geq f_1 \geq \max(e_1, e_2)$.
- Knowing $(P_1, P_2, \sigma(P_1), \sigma(P_2))$ is sufficient to compute F_1 and $\widetilde{F_2}$.

Verifying with less torsion

- We can divide F into

$$\mathcal{A} \xrightarrow{F_1} \mathcal{B} \xrightarrow{F_2} \mathcal{A}$$

where $\mathcal{A} := E_2^2 \times E_A^2$, F_i is an ℓ^{e_i}-isogeny (for $i = 1, 2$) and $e := e_1 + e_2$.

- $\ker(F_1) = \ker(F) \cap \mathcal{A}[\ell^{e_1}]$ and $\ker(\widetilde{F_2}) = F(\mathcal{A}[\ell^{e_2}])$.
- Let (P_1, P_2) be a basis of $E_A[\ell^{f_1}]$ with $f \geq f_1 \geq \max(e_1, e_2)$.
- Knowing $(P_1, P_2, \sigma(P_1), \sigma(P_2))$ is sufficient to compute F_1 and $\widetilde{F_2}$.
- The verifier accepts if the codomains of F_1 and $\widetilde{F_2}$ match.

Verifying with less torsion

Advantages:

- Use ℓ^{f_1}-torsion with $f_1 \simeq e/2$ instead of ℓ^e-torsion ($f_1 \leq f$).
- $q < \ell^e$ is not constained by the accessible torsion (more freedom on the choice of I).
- Makes signature communications $\sigma(P_1), \sigma(P_2)$ twice more compact.

Higher dimensional isogeny computation

Goal: Compute an ℓ^e-isogeny $F : \mathcal{A} \longrightarrow \mathcal{B}$.

- Let $\mathscr{B}_0$ be a basis of $\ker(F)$.
- Decompose:

$$\mathcal{A} = \mathcal{A}_0 \xrightarrow{F_1} \mathcal{A}_1 \cdots \mathcal{A}_{e-1} \xrightarrow{F_e} \mathcal{A}_e = \mathcal{B}$$

- Let $\mathscr{B}_i := F_i \circ \cdots \circ F_1(\mathscr{B}_0)$.
- Then $\ker(F_i) = \langle [\ell^{e-i}]\mathscr{B}_{i-1} \rangle$.

Higher dimensional isogeny computation

Goal: Compute an ℓ^e-isogeny $F : \mathcal{A} \longrightarrow \mathcal{B}$.

- Let $\mathscr{B}_0$ be a basis of $\ker(F)$.
- Decompose:

$$\mathcal{A} = \mathcal{A}_0 \xrightarrow{F_1} \mathcal{A}_1 \cdots \mathcal{A}_{e-1} \xrightarrow{F_e} \mathcal{A}_e = \mathcal{B}$$

- Let $\mathscr{B}_i := F_i \circ \cdots \circ F_1(\mathscr{B}_0)$.
- Then $\ker(F_i) = \langle [\ell^{e-i}]\mathscr{B}_{i-1} \rangle$.
- Descend the computation tree to compute the F_i (DFJP, 11).

Computation tree for $e = 5$

$$\begin{tikzcd}
 & & & & \mathscr{B}_0 \arrow[dl] \arrow[dr, dashed, "F_1"] & & & & \\
 & & & [\ell]\mathscr{B}_0 \arrow[dl] \arrow[dr, dashed] & & \mathscr{B}_1 \arrow[dl] \arrow[dr, dashed, "F_2"] & & & \\
 & & [\ell^2]\mathscr{B}_0 \arrow[dl] \arrow[dr, dashed] & & [\ell]\mathscr{B}_1 \arrow[dl] \arrow[dr, dashed] & & \mathscr{B}_2 \arrow[dl] \arrow[dr, dashed, "F_3"] & & \\
 & [\ell^3]\mathscr{B}_0 \arrow[dl] \arrow[dr, dashed] & & [\ell^2]\mathscr{B}_1 \arrow[dl] \arrow[dr, dashed] & & [\ell]\mathscr{B}_2 \arrow[dl] \arrow[dr, dashed] & & \mathscr{B}_3 \arrow[dl] \arrow[dr, dashed, "F_4"] & \\
[\ell^4]\mathscr{B}_0 & & [\ell^3]\mathscr{B}_1 & & [\ell^2]\mathscr{B}_2 & & [\ell]\mathscr{B}_3 & & \mathscr{B}_4
\end{tikzcd}$$

Higher dimensional isogeny computation

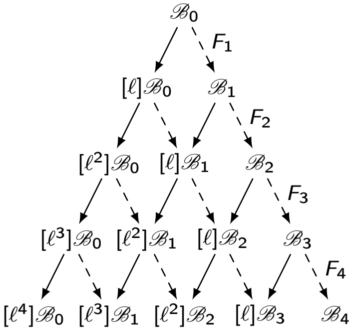
Goal: Compute an ℓ^e-isogeny $F : \mathcal{A} \longrightarrow \mathcal{B}$.

- Let $\mathscr{B}_0$ be a basis of $\ker(F)$.
- Decompose:

$$\mathcal{A} = \mathcal{A}_0 \xrightarrow{F_1} \mathcal{A}_1 \cdots \mathcal{A}_{e-1} \xrightarrow{F_e} \mathcal{A}_e = \mathcal{B}$$

- Let $\mathscr{B}_i := F_i \circ \cdots \circ F_1(\mathscr{B}_0)$.
- Then $\ker(F_i) = \langle [\ell^{e-i}]\mathscr{B}_{i-1} \rangle$.
- Descend the computation tree to compute the F_i (DFJP, 11).
- Each F_i is computed with the Θ model (level $\ell = 2$).

Computation tree for $e = 5$



Commitment (and key generation)

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Accessible torsion: $E_0[\ell^f \ell'^{f'}]$ $(p = c\ell^f \ell'^{f'} - 1)$.

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Accessible torsion: $E_0[\ell^f \ell'^{f'}]$ ($p = c\ell^f \ell'^{f'} - 1$).

$$\begin{tikzcd}
 & E_2' \arrow[d, dashed, "{[\theta_2]_*\theta_1'}"] & \\
E_0 \arrow[ur, "\theta_1"] \arrow[dr, "\theta_2'"'] & E_1 & E' \arrow[ul, "\theta_2"'] \arrow[dl, "\theta_1'"] \\
 & E_1' \arrow[u, dashed, "{[\theta_1']_*\theta_2}"] &
\end{tikzcd}$$

- Compute $\gamma \in \mathcal{O}_0 \cong \mathrm{End}(E_0)$ s.t. $\mathrm{nrd}(\gamma) = \ell^{2f} \ell'^{2f'}$.

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Accessible torsion: $E_0[\ell^f \ell'^{f'}]$ $(p = c\ell^f \ell'^{f'} - 1)$.

$$\begin{tikzcd}
 & E_2' \arrow[d, dashed, "{[\theta_2]_*\theta_1'}"] & \\
E_0 \arrow[ur, "\theta_1"] \arrow[dr, "\theta_2'"'] & E_1 & E' \arrow[ul, "\theta_2"'] \arrow[dl, "\theta_1'"] \\
 & E_1' \arrow[u, dashed, "{[\theta_1']_*\theta_2}"'] &
\end{tikzcd}$$

- Compute $\gamma \in \mathcal{O}_0 \cong \mathrm{End}(E_0)$ s.t. $\mathrm{nrd}(\gamma) = \ell^{2f} \ell'^{2f'}$.
- Factor $\gamma = \widehat{\theta_2'} \circ \theta_1' \circ \widehat{\theta_2} \circ \theta_1$, with $\deg(\theta_1) = \deg(\theta_1') = \ell^f$ and $\deg(\theta_2) = \deg(\theta_2') = \ell'^{f'}$.

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Accessible torsion: $E_0[\ell^f \ell'^{f'}]$ $(p = c\ell^f \ell'^{f'} - 1)$.

$$\begin{tikzcd}
 & E_2' \arrow[d, dashed, "{[\theta_2]_*\theta_1'}"] & \\
E_0 \arrow[ur, "\theta_1"] \arrow[dr, "\theta_2'"'] & E_1 & E' \arrow[ul, "\theta_2"'] \arrow[dl, "\theta_1'"] \\
 & E_1' \arrow[u, dashed, "{[\theta_1']_*\theta_2}"'] &
\end{tikzcd}$$

- Compute $\gamma \in \mathcal{O}_0 \cong \mathrm{End}(E_0)$ s.t. $\mathrm{nrd}(\gamma) = \ell^{2f}\ell'^{2f'}$.
- Factor $\gamma = \widehat{\theta_2'} \circ \theta_1' \circ \widehat{\theta_2} \circ \theta_1$, with $\deg(\theta_1) = \deg(\theta_1') = \ell^f$ and $\deg(\theta_2) = \deg(\theta_2') = \ell'^{f'}$.
- Compute $[\theta_1']_*\theta_2$ of kernel $\theta_1'(\ker \theta_2)$.
- Compute $[\theta_2]_*\theta_1'$ of kernel $\theta_2(\ker \theta_1')$.

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Accessible torsion: $E_0[\ell^f \ell'^{f'}]$ $(p = c\ell^f \ell'^{f'} - 1)$.

$$
\begin{tikzcd}
 & & E_2' \arrow[d, dashed, "{[\theta_2]_*\theta_1'}"] & & \\
E_0 \arrow[urr, "\theta_1"] \arrow[drr, "\theta_2'"'] & & E_1 & & E' \arrow[ull, "\theta_2"'] \arrow[dll, "\theta_1'"] \\
 & & E_1' \arrow[u, dashed, "{[\theta_1']_*\theta_2}"'] & &
\end{tikzcd}
$$

- Compute $\gamma \in \mathcal{O}_0 \cong \mathrm{End}(E_0)$ s.t. $\mathrm{nrd}(\gamma) = \ell^{2f} \ell'^{2f'}$.
- Factor $\gamma = \widehat{\theta_2'} \circ \theta_1' \circ \widehat{\theta_2} \circ \theta_1$, with $\deg(\theta_1) = \deg(\theta_1') = \ell^f$ and $\deg(\theta_2) = \deg(\theta_2') = \ell'^{f'}$.
- Compute $[\theta_1']_*\theta_2$ of kernel $\theta_1'(\ker \theta_2)$.
- Compute $[\theta_2]_*\theta_1'$ of kernel $\theta_2(\ker \theta_1')$.
- $\psi := [\theta_1']_*\theta_2 \circ \theta_2'$ and $\psi' := [\theta_2]_*\theta_1' \circ \theta_1$.

Double path to the commitment

Goal: Compute $\psi, \psi' : E_0 \longrightarrow E_1$ and $I_\psi, I_{\psi'}$ with $\deg(\psi) = \ell^{2f}$ and $\deg(\psi) = \ell'^{2f'}$.

Accessible torsion: $E_0[\ell^f \ell'^{f'}]$ $(p = c\ell^f \ell'^{f'} - 1)$.

$$\begin{tikzcd}
 & E_2' \arrow[d, dashed, "{[\theta_2]_*\theta_1'}"] & \\
E_0 \arrow[ur, "\theta_1"] \arrow[dr, "\theta_2'"'] & E_1 & E' \arrow[ul, "\theta_2"'] \arrow[dl, "\theta_1'"] \\
 & E_1' \arrow[u, dashed, "{[\theta_1']_*\theta_2}"'] &
\end{tikzcd}$$

- Compute $\gamma \in \mathcal{O}_0 \cong \mathrm{End}(E_0)$ s.t. $\mathrm{nrd}(\gamma) = \ell^{2f}\ell'^{2f'}$.
- Factor $\gamma = \widehat{\theta_2'} \circ \theta_1' \circ \widehat{\theta_2} \circ \theta_1$, with $\deg(\theta_1) = \deg(\theta_1') = \ell^f$ and $\deg(\theta_2) = \deg(\theta_2') = \ell'^{f'}$.
- Compute $[\theta_1']_*\theta_2$ of kernel $\theta_1'(\ker \theta_2)$.
- Compute $[\theta_2]_*\theta_1'$ of kernel $\theta_2(\ker \theta_1')$.
- $\psi := [\theta_1']_*\theta_2 \circ \theta_2'$ and $\psi' := [\theta_2]_*\theta_1' \circ \theta_1$.
- $I_\psi := \mathcal{O}_0\overline{\gamma} + \mathcal{O}_0\ell'^{2f'}$, $I_{\psi'} := \mathcal{O}_0\gamma + \mathcal{O}_0\ell^{2f}$.

Security analysis

Outline of the security analysis

Theorem (Fiat-Shamir, 1986)

Let ID be an identification protocol that is:

- ***Complete:*** *a honest execution is always accepted by the verifier.*
- ***Sound:*** *an attacker cannot "guess" a response.*
- ***Zero-knowledge:*** *the response does not leak any information on the secret key.*

Then the Fiat-Shamir transform of ID is a universally unforgeable signature under chosen message attacks in the random oracle model.

Soundness

Similar to SQISign

Proposition (Special soundness)

Assume $q \wedge \ell' = 1$.
Then given two transcripts $(E_1, \varphi, R), (E_1, \varphi', R')$ with the same commitment E_1 and $\varphi \neq \varphi'$, we can extract $\alpha \in \mathrm{End}(E_A)$ non-scalar.

Proof.

- Exctract σ from R and σ' from R'.
- Then $\alpha := \widehat{\varphi}' \circ \widehat{\sigma}' \circ \sigma \circ \varphi \in \mathrm{End}(E_A)$ is non-scalar.

□

Zero-knowledge

Definition

We say that an integer q is **good** if:

- $\ell^e - q$ is a prime $\equiv 1 \mod 4$.
- $q \wedge \ell' = 1$.

Zero-knowledge

Definition

We say that an integer q is **good** if:

- $\ell^e - q$ is a prime $\equiv 1 \mod 4$.
- $q \wedge \ell' = 1$.

Definition (RUGDIO)

A random uniform good degree isogeny oracle (RUGDIO):
Input: A supersingular elliptic curve $E/\mathbb{F}_{p^2}$.
Output: An isogeny $\sigma : E \longrightarrow E'$ of good degree q s.t.

- E' is uniform in the supersingular isogeny graph.
- Given E', σ is uniform among isogenies of good degree $E \longrightarrow E'$.

Zero-knowledge

Theorem

Assume that:

- E_1 *is computationally close to uniform.*
- *We have access to a RUGDIO.*

Then SQISignHD is computationally honest-verifier zero-knowledge.

Zero-knowledge

Theorem

Assume that:

- *E_1 is computationally close to uniform.*
- *We have access to a RUGDIO.*

Then SQISignHD is computationally honest-verifier zero-knowledge.

Proof.

We build a simulator $\mathcal{S}$ of protocol transcripts:

- $\mathcal{S}$ calls the RUGDIO to generate an efficient representation R of $\sigma : E_A \longrightarrow E_2$.
- $\mathcal{S}$ generates a random challenge $\widehat{\varphi} : E_2 \longrightarrow E_1$.
- $\mathcal{S}$ outputs (E_1, φ, R).

$\square$

Zero-knowledge: comparison with SQISign

Heuristic assumptions to prove the zero-knowledge property

In SQISign:

- $\sigma : E_A \longrightarrow E_2$ is computationally indistinguishable from a random isogeny of degree ℓ^e.

In SQISignHD:

- E_1 is computationally close to uniform.
- We have access to a RUGDIO.

Zero-knowledge: the interest of dimension 8

In SQISign:

- $\sigma : E_A \longrightarrow E_2$ is computationally close to a random isogeny of degree ℓ^e.

Zero-knowledge: the interest of dimension 8

In SQISign:

- $\sigma : E_A \longrightarrow E_2$ is computationally close to a random isogeny of degree ℓ^e.

In RigorousSQISignHD:

- We have access to a RADIO.

Definition (RADIO)

A random any degree isogeny oracle (RADIO):
Input: A supersingular elliptic curve $E/\mathbb{F}_{p^2}$.
Output: An efficient representation of a uniformly random isogeny $\sigma : E \longrightarrow E'$ of degree $q < \ell^e$.

Performance

Compact signatures

In SQISign: $\sigma : E_A \longrightarrow E_2$ of degree $\ell^e \simeq p^{15/4}$.

In SQISignHD: $(j(E_1), q, \sigma(P_1), \sigma(P_2))$, where:

- (P_1, P_2) is a basis of $E_A[\ell^{f_1}]$.
  NB: no need to transmit (P_1, P_2).
- $\ell^{f_1} \simeq p^{1/4}$.
- $q \simeq \sqrt{p}$.
- $j(E_1) \in \mathbb{F}_{p^2}$ has size $2\log_2(p)$ bits.

Compact signatures

In SQISign: $\sigma : E_A \longrightarrow E_2$ of degree $\ell^e \simeq p^{15/4}$.

In SQISignHD: $(j(E_1), q, \sigma(P_1), \sigma(P_2))$, where:

- (P_1, P_2) is a basis of $E_A[\ell^{f_1}]$.
  NB: no need to transmit (P_1, P_2).
- $\ell^{f_1} \simeq p^{1/4}$.
- $q \simeq \sqrt{p}$.
- $j(E_1) \in \mathbb{F}_{p^2}$ has size $2\log_2(p)$ bits.

Signature size (in bits)

	In SQISign	In SQISignHD
In general	$\sim 15/4 \log_2(p)$	$\sim 13/4 \log_2(p)$
NIST-1 security level	1060	840

Fast signatures at the expense of verification

Fast signature:

- Only 1-dimensional isogenies: ψ, ψ', φ.
- Evaluating $(\sigma(P_1), \sigma(P_2))$ with $\varphi \circ \psi \circ \widehat{\tau}$ is fast.
- Preliminary implementation way faster than SQISign.

Fast signatures at the expense of verification

Fast signature:

- Only 1-dimensional isogenies: ψ, ψ', φ.
- Evaluating $(\sigma(P_1), \sigma(P_2))$ with $\varphi \circ \psi \circ \widehat{\tau}$ is fast.
- Preliminary implementation way faster than SQISign.

Comparison with SQISign signatures:

- Slow ideal-to-isogeny translation: $I \longmapsto \sigma$.
- Piecewise computation involving 30 T-isogenies with $T \simeq p^{5/4}$.

Fast signatures at the expense of verification

Fast signature:

- Only 1-dimensional isogenies: ψ, ψ', φ.
- Evaluating $(\sigma(P_1), \sigma(P_2))$ with $\varphi \circ \psi \circ \widehat{\tau}$ is fast.
- Preliminary implementation way faster than SQISign.

Comparison with SQISign signatures:

- Slow ideal-to-isogeny translation: $I \longmapsto \sigma$.
- Piecewise computation involving 30 T-isogenies with $T \simeq p^{5/4}$.

Verification:

- Isogenies in dimension 4 to compute.
- Known algorithms.
- But to be implemented...

Conclusion

Comparison of SQISignHD with SQISign

	SQISign	SQISignHD
Security	Ad-hoc heuristic: • Distribution of σ.	Simpler heuristics: • RUGDIO; • Distribution of E_1.

Comparison of SQISignHD with SQISign

	SQISign	SQISignHD
Security	Ad-hoc heuristic: • Distribution of σ.	Simpler heuristics: • RUGDIO; • Distribution of E_1.
Prime p	$\ell^f T \mid p^2 - 1$ with $T \simeq p^{5/4}$ • Slow isogeny computations • Not certain if it scales	$p = c\ell^f \ell'^{f'} - 1$ • Fast isogeny computations • Scales well

Comparison of SQISignHD with SQISign

	SQISign	SQISignHD
Security	Ad-hoc heuristic: • Distribution of σ.	Simpler heuristics: • RUGDIO; • Distribution of E_1.
Prime p	$\ell^f T \mid p^2-1$ with $T \simeq p^{5/4}$ • Slow isogeny computations • Not certain if it scales	$p = c\ell^f \ell'^{f'} - 1$ • Fast isogeny computations • Scales well
Signature	σ with $\deg(\sigma) = \ell^e \simeq p^{15/4}$ • Ideal-to-isogeny translation • 30 T-isogenies involved	$(q, \sigma(P_1), \sigma(P_2))$ • Fast via $\varphi \circ \psi \circ \widehat{\tau}$

Comparison of SQISignHD with SQISign

	SQISign	SQISignHD
Security	Ad-hoc heuristic: • Distribution of σ.	Simpler heuristics: • RUGDIO; • Distribution of E_1.
Prime p	$\ell^f T \mid p^2-1$ with $T \simeq p^{5/4}$ • Slow isogeny computations • Not certain if it scales	$p = c\ell^f \ell'^{f'} - 1$ • Fast isogeny computations • Scales well
Signature	σ with $\deg(\sigma) = \ell^e \simeq p^{15/4}$ • Ideal-to-isogeny translation • 30 T-isogenies involved	$(q, \sigma(P_1), \sigma(P_2))$ • Fast via $\varphi \circ \psi \circ \widehat{\tau}$
Verification	• Recompute σ as a chain of ℓ-isogenies of known kernels • $\deg(\sigma) = \ell^e \simeq p^{15/4}$	• Compute F an ℓ^e-isogeny of dimension 4 • $\deg(\sigma) = \ell^e \simeq \sqrt{p}$

Thank you for listening.

Find our pre-print here: https://eprint.iacr.org/2023/436