SQIPrime: Yet Another Variant of SQISignHD

Max Duparc[1],Tako Boris Fouotsa[2]

[1]Nagra

[2]LASEC, EPFL

May 24, 2024

SQIPrime: A simple mechanism

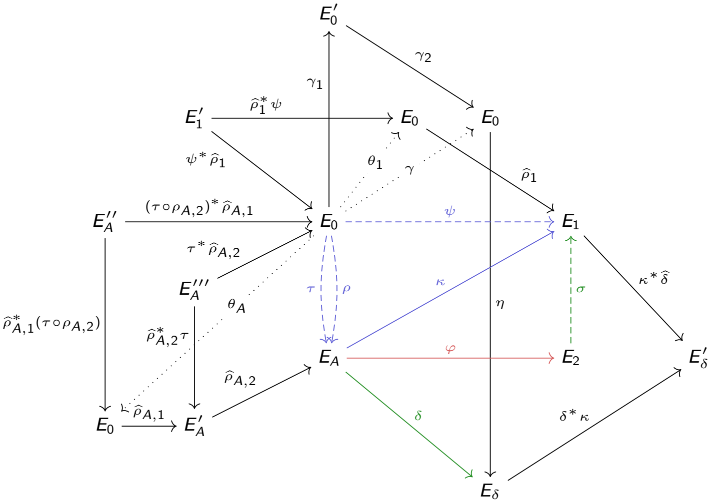
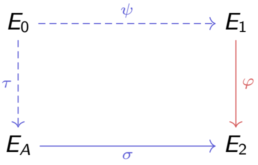


Table of Contents

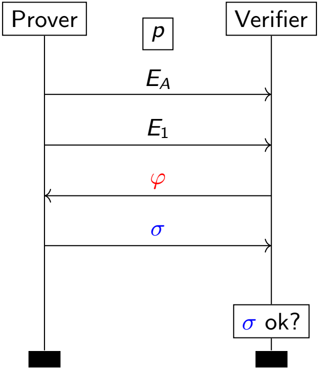
SQISign & SQISignHD

SQISign [DFKL+20, DFLLW23]:

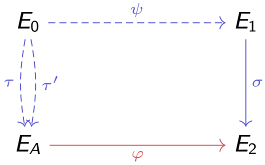


- σ long ($\simeq p^4$) smooth.
- Given in kernel representation.

\+ Compact. (177 B)
− Slow signature.
\+ Quick verification.
− Hard primes.



SQISignHD [DLRW23]:

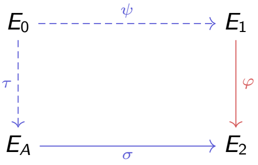


- σ short ($\simeq \sqrt{p}$).
- Given in HD representation.

\+ Very compact. (109 B)
\+ Quick signature.
− Long verification.
\+ Easy primes.

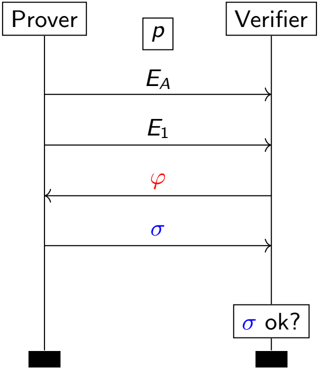
SQISign & SQISignHD

SQISign [DFKL+20, DFLLW23]:

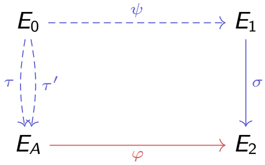


- σ long ($\simeq p^4$) smooth.
- Given in kernel representation.

\+ Compact. (177 B)
– Slow signature.
\+ Quick verification.
– Hard primes.



SQISignHD [DLRW23]:

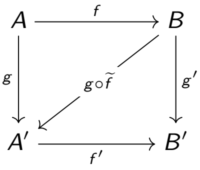


- σ short ($\simeq \sqrt{p}$).
- Given in HD representation.

\+ Very compact. (109 B)
\+ Quick signature.
– Long verification.
\+ Easy primes.

Kani's Lemma [Kan97]

Kani [CD23, MMP+23]:



- g is often an unsmooth isogeny.
- Practical in dim 2.

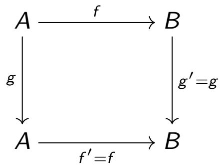
− Hard to use.
− Requires $\deg(F)$ torsion.
− Dual sensitive.
+ Quick.

$$F := \begin{pmatrix} \tilde{f} & -\tilde{g} \\ g' & f' \end{pmatrix}$$

$$F : B \times A' \to A \times B'$$

$$\deg(F) = \deg(f) + \deg(g)$$

Kani [Rob23]:

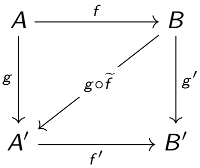


- g is trivial HD endomorphism.
- Practical in dim 4 or 8.

+ Easy to use
+ Requires $\sqrt{\deg(F)}$ torsion.
+ Not dual sensitive.
− Slow.

Kani's Lemma [Kan97]

Kani [CD23, MMP+23]:



- g is often an unsmooth isogeny.
- Practical in dim 2.

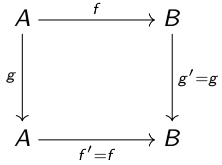
– Hard to use.
– Requires $\deg(F)$ torsion.
– Dual sensitive.
+ Quick.

$$F := \begin{pmatrix} \tilde{f} & -\tilde{g} \\ g' & f' \end{pmatrix}$$

$$F : B \times A' \to A \times B'$$

$$\deg(F) = \deg(f) + \deg(g)$$

Kani [Rob23]:



- g is trivial HD endomorphism.
- Practical in dim 4 or 8.

+ Easy to use
+ Requires $\sqrt{\deg(F)}$ torsion.
+ Not dual sensitive.
– Slow.

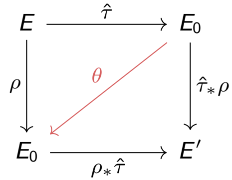
Inspiration: QFESTA [NO23]

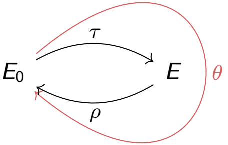
RandIsogImages:

1. Find $\theta \in \mathrm{End}(E_0)$ with $\deg(\theta) = \ell(2^\alpha - \ell) > p$.
2. Use Kani's Lemma to split θ.

$$\ker(F) = \left\{ ([\ell](P), \theta(P)) \middle| \; P \in E_0[2^\alpha] \right\}$$

   $\deg(\tau)$ and $\deg(\rho)$ coprime.
3. Finding I_τ, I_ρ from θ is easy.





- This is a **Doublepath** algorithm that:
  - is quick.
  - works with unsmooth degree.
  - Provided $2^\alpha \simeq p$, has a good distribution*.

Inspiration: QFESTA [NO23]

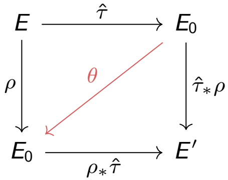
RandIsogImages:

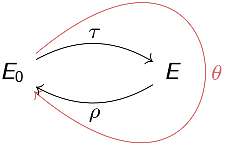
1. Find $\theta \in \text{End}(E_0)$ with $\deg(\theta) = \ell(2^\alpha - \ell) > p$.
2. Use Kani's Lemma to split θ.

$$\ker(F) = \left\{ ([\ell](P), \theta(P)) \,\middle|\, P \in E_0[2^\alpha] \right\}$$

$\deg(\tau)$ and $\deg(\rho)$ coprime.

3. Finding I_τ, I_ρ from θ is easy.





- This is a **Doublepath** algorithm that:
  - is quick.
  - works with unsmooth degree.
  - Provided $2^\alpha \simeq p$, has a good distribution*.

Inspiration: DeuringVRF [Ler23]

Problem: The KernelToIdeal only works on *smooth* isogenies.

- [Ler23] proposed an approach based on precomputed basis:

Precomputed Basis

Let E be any supersingular curve. (P, Q, ι, I_P) is special basis of $E[q]$, with:

- $P, Q \in E$ such that $\langle P, Q \rangle = E[q]$.
- $\iota \in \mathrm{End}(E)$ such that $\iota(P) = Q$.
- I_P is such that $E[I_P] = \langle P \rangle$.

1. We can perform unsmooth KernelToIdeal.

$$\ker(\varphi) = \langle [a]P + [b]Q \rangle \implies I_\varphi = [a + b\iota]_* I_P$$

2. This is preserved through isogenies.

$$\ker(\varphi) = \langle [a]\tau(P) + [b]\tau(Q) \rangle \implies I_\varphi = [(a + b\iota) I_\tau]_* I_P$$

Inspiration: DeuringVRF [Ler23]

Problem: The KernelToIdeal only works on *smooth* isogenies.

▶ [Ler23] proposed an approach based on precomputed basis:

Precomputed Basis

Let E be any supersingular curve. (P, Q, ι, I_P) is special basis of $E[q]$, with:

- $P, Q \in E$ such that $\langle P, Q \rangle = E[q]$.
- $\iota \in \mathrm{End}(E)$ such that $\iota(P) = Q$.
- I_P is such that $E[I_P] = \langle P \rangle$.

1. We can perform unsmooth KernelToIdeal.

$$\ker(\varphi) = \langle [a]P + [b]Q \rangle \implies I_\varphi = [a + b\iota]_* I_P$$

2. This is preserved through isogenies.

$$\ker(\varphi) = \langle [a]\tau(P) + [b]\tau(Q) \rangle \implies I_\varphi = [(a + b\iota)I_\tau]_* I_P$$

Inspiration: DeuringVRF [Ler23]

Problem: The KernelToIdeal only works on *smooth* isogenies.

- [Ler23] proposed an approach based on precomputed basis:

Precomputed Basis

Let E be any supersingular curve. (P, Q, ι, I_P) is special basis of $E[q]$, with:

- $P, Q \in E$ such that $\langle P, Q \rangle = E[q]$.
- $\iota \in \mathrm{End}(E)$ such that $\iota(P) = Q$.
- I_P is such that $E[I_P] = \langle P \rangle$.

1. We can perform unsmooth KernelToIdeal.

$$\ker(\varphi) = \langle [a]P + [b]Q \rangle \implies I_\varphi = [a + b\iota]_* I_P$$

2. This is preserved through isogenies.

$$\ker(\varphi) = \langle [a]\tau(P) + [b]\tau(Q) \rangle \implies I_\varphi = [(a + b\iota)I_\tau]_* I_P$$

Table of Contents

SQIPrime2D: Public parameters, KeyGen & Commit

- **Public parameters**:
  - $p + 1 = 2^\alpha f$ with $p - 1 = 2Nq$ with $q \simeq 2^\lambda$ prime with
    - $\alpha \geq \lceil \frac{\log_2(p)}{2} + \log_2(q) \rceil + 1$.
  - $\langle P_0, Q_0 \rangle = E_0[2^\alpha]$.
  - (P, Q, ι, I_P) a precomputed basis of $E_0[q]$.

- **Key Generation**:
  - Use KaniDoublePath to compute τ and $\widehat{\rho}$.
  - $\deg(\tau) = q$.
  - $\deg(\widehat{\rho}) = 2^\alpha - q$.
  - $\binom{R}{S} = \widehat{\rho}\binom{P}{Q}$.
  - $\mathrm{pk} = E_A, R, S \quad \mathrm{sk} = \tau, \widehat{\rho}$

E_0

- **Commitment**:
  - Use KaniDoublePath to compute ψ.
  - $\deg(\psi) = \ell_1 < 2^\alpha$ random prime.
  - Share E_1.

SQIPrime2D: Public parameters, KeyGen & Commit

- **Public parameters**:
  - $p+1 = 2^\alpha f$ with $p-1 = 2Nq$ with $q \simeq 2^\lambda$ prime with
    - $\alpha \geq \lceil \frac{\log_2(p)}{2} + \log_2(q) \rceil + 1$.
  - $\langle P_0, Q_0 \rangle = E_0[2^\alpha]$.
  - (P, Q, ι, I_P) a precomputed basis of $E_0[q]$.

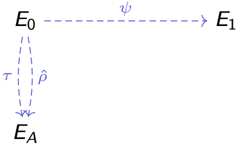
- **Key Generation**:
  - Use KaniDoublePath to compute τ and $\widehat{\rho}$.
  - $\deg(\tau) = q$.
  - $\deg(\widehat{\rho}) = 2^\alpha - q$.
  - $\binom{R}{S} = \widehat{\rho}\binom{P}{Q}$.
  - pk $= E_A, R, S$ sk $= \tau, \widehat{\rho}$

- **Commitment**:
  - Use KaniDoublePath to compute ψ.
  - $\deg(\psi) = \ell_1 < 2^\alpha$ random prime.
  - Share E_1.

SQIPrime2D: Public parameters, KeyGen & Commit

- **Public parameters**:
  - $p+1 = 2^{\alpha} f$ with $p-1 = 2Nq$ with $q \simeq 2^{\lambda}$ prime with
    - $\alpha \geq \lceil \frac{\log_2(p)}{2} + \log_2(q) \rceil + 1$.
  - $\langle P_0, Q_0 \rangle = E_0[2^{\alpha}]$.
  - (P, Q, ι, I_P) a precomputed basis of $E_0[q]$.

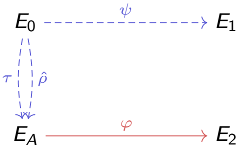
- **Key Generation**:
  - Use KaniDoublePath to compute τ and $\widehat{\rho}$.
  - $\deg(\tau) = q$.
  - $\deg(\widehat{\rho}) = 2^{\alpha} - q$.
  - $\binom{R}{S} = \widehat{\rho}\binom{P}{Q}$.
  - pk $= E_A, R, S$ sk $= \tau, \widehat{\rho}$



- **Commitment**:
  - Use KaniDoublePath to compute ψ.
  - $\deg(\psi) = \ell_1 < 2^{\alpha}$ random prime.
  - Share E_1.

SQIPrime2D: Challenge & Response[1]

- **Challenge**:
  - The challenge is $a \in \mathbb{Z}_q$.
  - a defines $C_a = R + [a]S$.
  - C_a is the kernel gen. of $\varphi : E_A \to E_2$
  - Share a.

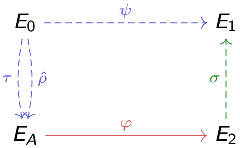


- **Response[1]**:
  1. Retrieve $I_\varphi = [(1 + a\iota)I_{\hat{\rho}}]_* I_P$.
  2. Use $\overline{I_\varphi I_\tau} I_\psi$ to compute $J \sim \sigma : E_2 \to E_1$ of degree $d \leq 2\sqrt{p}$ odd.
  3. We will consider and work with $\kappa = \sigma \circ \varphi$.

SQIPrime2D: Challenge & Response[1]

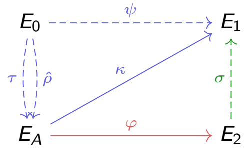
- **Challenge**:
  - The challenge is $a \in \mathbb{Z}_q$.
  - a defines $C_a = R + [a]S$.
  - C_a is the kernel gen. of $\varphi : E_A \to E_2$
  - Share a.

- **Response[1]**:
  1. Retrieve $I_\varphi = [(1 + a\iota) I_{\widehat{\rho}}]_* I_P$.
  2. Use $\overline{I_\varphi I_\tau} I_\psi$ to compute $J \sim \sigma : E_2 \to E_1$ of degree $d \leq 2\sqrt{p}$ odd.
  3. We will consider and work with $\kappa = \sigma \circ \varphi$.



SQIPrime2D: Challenge & Response[1]

- **Challenge**:
  - The challenge is $a \in \mathbb{Z}_q$.
  - a defines $C_a = R + [a]S$.
  - C_a is the kernel gen. of $\varphi : E_A \to E_2$
  - Share a.

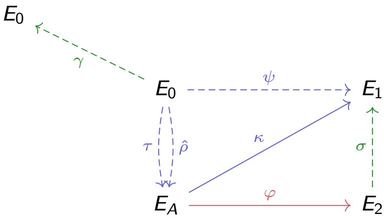


- **Response[1]**:
  1. Retrieve $I_\varphi = [(1 + a\iota)I_{\hat{\rho}}]_* I_P$.
  2. Use $\overline{I_\varphi I_\tau} I_\psi$ to compute $J \sim \sigma : E_2 \to E_1$ of degree $d \le 2\sqrt{p}$ odd.
  3. We will consider and work with $\kappa = \sigma \circ \varphi$.

$$\kappa = \sigma \circ \varphi \iff \kappa(C_a) = 0$$

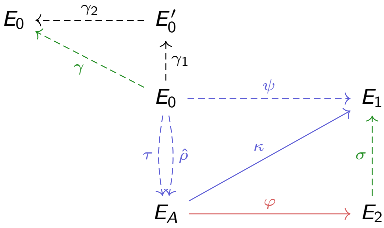
SQIPrime2D: Response[2]

- **Response[2]**:
  4. Compute $\gamma \in \mathrm{End}(E_0)$ such that $n(\gamma) = d(2^\alpha - dq)$ and evaluate $\tau \circ \widehat{\gamma}$ over $E_0[2^\alpha]$.
  5. Write $\gamma = \gamma_2 \circ \gamma_1$ where $\deg \gamma_1 = d$ and $\deg \gamma_2 = (2^\alpha - dq)$.
  6. Use Kani (where the isogenies in play are γ_2 and $\tau \circ \widehat{\gamma_1}$) to retrieve the pushforward $\delta : E_1 \to E_\delta$ of γ_2 through $\tau \circ \widehat{\gamma_1}$.
  7. Compute X, Y a deterministic basis of $E_1[2^\alpha]$.
  8. Evaluate $\binom{T}{U} = [(qd)^{-1}]\delta \circ \widehat{\kappa}\binom{X}{Y}$ and $V = \delta(C_a)$.
  9. Return E_δ, T, U, V.



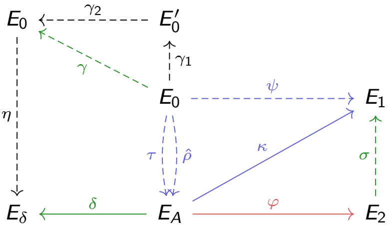
SQIPrime2D: Response[2]

- **Response[2]**:
  4. Compute $\gamma \in \mathrm{End}(E_0)$ such that $n(\gamma) = d(2^\alpha - dq)$ and evaluate $\tau \circ \widehat{\gamma}$ over $E_0[2^\alpha]$.
  5. Write $\gamma = \gamma_2 \circ \gamma_1$ where $\deg \gamma_1 = d$ and $\deg \gamma_2 = (2^\alpha - dq)$.
  6. Use Kani (where the isogenies in play are γ_2 and $\tau \circ \widehat{\gamma_1}$) to retrieve the pushforward $\delta : E_1 \to E_\delta$ of γ_2 through $\tau \circ \widehat{\gamma_1}$.
  7. Compute X, Y a deterministic basis of $E_1[2^\alpha]$.
  8. Evaluate $\binom{T}{U} = [(qd)^{-1}]\delta \circ \widehat{\kappa}\binom{X}{Y}$ and $V = \delta(C_a)$.
  9. Return E_δ, T, U, V.



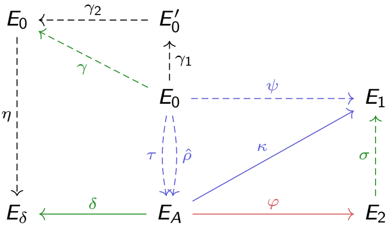
SQIPrime2D: Response[2]

- **Response[2]**:
  4. Compute $\gamma \in \mathrm{End}(E_0)$ such that $n(\gamma) = d(2^\alpha - dq)$ and evaluate $\tau \circ \widehat{\gamma}$ over $E_0[2^\alpha]$.
  5. Write $\gamma = \gamma_2 \circ \gamma_1$ where $\deg \gamma_1 = d$ and $\deg \gamma_2 = (2^\alpha - dq)$.
  6. Use Kani (where the isogenies in play are γ_2 and $\tau \circ \widehat{\gamma_1}$) to retrieve the pushforward $\delta : E_1 \to E_\delta$ of γ_2 through $\tau \circ \widehat{\gamma_1}$.
  7. Compute X, Y a deterministic basis of $E_1[2^\alpha]$.
  8. Evaluate $\binom{T}{U} = [(qd)^{-1}]\delta \circ \widehat{\kappa}\binom{X}{Y}$ and $V = \delta(C_a)$.
  9. Return E_δ, T, U, V.



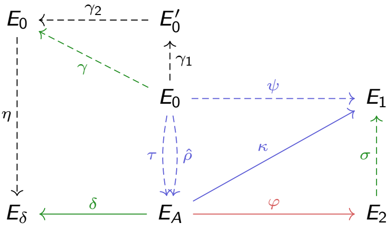
SQIPrime2D: Response[2]

- **Response[2]**:
  4. Compute $\gamma \in \mathrm{End}(E_0)$ such that $n(\gamma) = d(2^\alpha - dq)$ and evaluate $\tau \circ \widehat{\gamma}$ over $E_0[2^\alpha]$.
  5. Write $\gamma = \gamma_2 \circ \gamma_1$ where $\deg \gamma_1 = d$ and $\deg \gamma_2 = (2^\alpha - dq)$.
  6. Use Kani (where the isogenies in play are γ_2 and $\tau \circ \widehat{\gamma_1}$) to retrieve the pushforward $\delta : E_1 \to E_\delta$ of γ_2 through $\tau \circ \widehat{\gamma_1}$.
  7. Compute X, Y a deterministic basis of $E_1[2^\alpha]$.
  8. Evaluate $\binom{T}{U} = [(qd)^{-1}]\delta \circ \widehat{\kappa}\binom{X}{Y}$ and $V = \delta(C_a)$.
  9. Return E_δ, T, U, V.



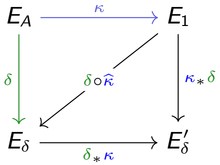
SQIPrime2D: Response[2]

- **Response[2]**:
  4. Compute $\gamma \in \mathrm{End}(E_0)$ such that $n(\gamma) = d(2^\alpha - dq)$ and evaluate $\tau \circ \widehat{\gamma}$ over $E_0[2^\alpha]$.
  5. Write $\gamma = \gamma_2 \circ \gamma_1$ where $\deg \gamma_1 = d$ and $\deg \gamma_2 = (2^\alpha - dq)$.
  6. Use Kani (where the isogenies in play are γ_2 and $\tau \circ \widehat{\gamma_1}$) to retrieve the pushforward $\delta : E_1 \to E_\delta$ of γ_2 through $\tau \circ \widehat{\gamma_1}$.
  7. Compute X, Y a deterministic basis of $E_1[2^\alpha]$.
  8. Evaluate $\binom{T}{U} = [(qd)^{-1}]\delta \circ \widehat{\kappa}\binom{X}{Y}$ and $V = \delta(C_a)$.
  9. Return E_δ, T, U, V.



SQIPrime2D: Verification

- **Verification**:
  1. Using the basis $\mathcal{B} = \{(X, T), (Y, U)\}$, define the dim 2 isogeny F.
  2. Check that $\mathrm{codomain}(F) = E_A \times -$.
  3. Compute $\binom{Z_1}{Z_2} = F\binom{0}{V}$ and check that:
     - $Z_1 = [2^\alpha](R + [a]S) = [2^\alpha]C_a$.
     - $Z_2 = 0$.
  4. Soundness comes from the equivalence $C_a \in \ker(\kappa) \iff \delta(C_a) \in \ker(\delta_* \kappa)$.

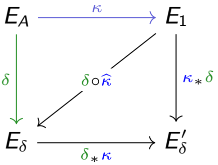


$$F : E_1 \times E_\delta \to E_A \times E'_\delta$$

$$\begin{pmatrix} \widehat{\kappa} & -\widehat{\delta} \\ \kappa_* \delta & \delta_* \kappa \end{pmatrix}$$

SQIPrime2D: Verification

- **Verification**:
  1. Using the basis $\mathcal{B} = \{(X, T), (Y, U)\}$, define the dim 2 isogeny F.
  2. Check that $\text{codomain}(F) = E_A \times -$.
  3. Compute $\binom{Z_1}{Z_2} = F\binom{0}{V}$ and check that:
     - $Z_1 = [2^\alpha](R + [a]S) = [2^\alpha]C_a$.
     - $Z_2 = 0$.
  4. Soundness comes from the equivalence $C_a \in \ker(\kappa) \iff \delta(C_a) \in \ker(\delta_*\kappa)$.



$$F : E_1 \times E_\delta \to E_A \times E'_\delta$$

$$\begin{pmatrix} \widehat{\kappa} & -\widehat{\delta} \\ \kappa_*\delta & \delta_*\kappa \end{pmatrix}$$

Table of Contents

Security

- **SQIPrime2D is a Σ-protocol**:
  - Special soundness is the same as SQISign & SQISignHD.
  - HVZK under RUCODIO+AIO.
- Key security reduces to the following problem:
  - *Let $\phi_1, \phi_2 : E_0 \to E'$ be two supersingular isogenies of degree q and $2^\alpha - q$ respectively, where $q \simeq 2^\lambda$ is a prime. Given E_0, E' and $\phi_2(E_0[q])$, retrieve ϕ_1 or ϕ_2.*
- Relies on assumptions regarding the distribution of KaniDoublePath:
  - Can only compute $\Theta(\ell/\log(p))$ isogenies of degree ℓ.
  - Assumed computational undistinguishability of the codomain.

Security

- **SQIPrime2D is a Σ-protocol**:
  - Special soundness is the same as SQISign & SQISignHD.
  - HVZK under RUCODIO+AIO.
- Key security reduces to the following problem:
  - *Let $\phi_1, \phi_2 : E_0 \to E'$ be two supersingular isogenies of degree q and $2^\alpha - q$ respectively, where $q \simeq 2^\lambda$ is a prime. Given E_0, E' and $\phi_2(E_0[q])$, retrieve ϕ_1 or ϕ_2.*
- Relies on assumptions regarding the distribution of KaniDoublePath:
  - Can only compute $\Theta(\ell/\log(p))$ isogenies of degree ℓ.
  - Assumed computational undistinguishability of the codomain.

Security

- **SQIPrime2D is a Σ-protocol**:
  - Special soundness is the same as SQISign & SQISignHD.
  - HVZK under RUCODIO+AIO.
- Key security reduces to the following problem:
  - *Let $\phi_1, \phi_2 : E_0 \to E'$ be two supersingular isogenies of degree q and $2^\alpha - q$ respectively, where $q \simeq 2^\lambda$ is a prime. Given E_0, E' and $\phi_2(E_0[q])$, retrieve ϕ_1 or ϕ_2.*
- Relies on assumptions regarding the distribution of KaniDoublePath:
  - Can only compute $\Theta(\ell/\log(p))$ isogenies of degree ℓ.
  - Assumed computational undistinguishability of the codomain.

Parameters

- On average, "SQIPrime2D-friendly" prime for security λ are of size $2\lambda + 4\log_2(\lambda)$ bits.
  - Finding *good* primes is a search for statistical anomalies:

$$p_{130} + 1 = 2^{273} \cdot 19^{2} \simeq 2^{281.50}$$
$$p_{130} - 1 = 2 \cdot 3 \cdot 5 \cdot 59 \cdot 191 \cdot 2797 \cdot 16585601 \cdot 2015747199847233809284079593\,07 \cdot q_{130}$$
$$q_{130} = 1733124013302036320718171822563477047667 \simeq 2^{130.35}$$

$$p_{117} + 1 = 2^{247} \cdot 79 \simeq 2^{253.34}$$
$$p_{117} - 1 = 2 \cdot 3 \cdot 5 \cdot 2903 \cdot 1924673583633629 \cdot 6340099406996072110\,39 \cdot q_{117}$$
$$q_{117} = 168118140144706967996895604212334429 \simeq 2^{117.01}$$

- Finding *good* "SQIPrime2D-friendly" primes requires a lot of factoring.

Parameters

- On average, "SQIPrime2D-friendly" prime for security λ are of size $2\lambda + 4\log_2(\lambda)$ bits.
  - Finding *good* primes is a search for statistical anomalies:

$$p_{130} + 1 = 2^{273} \cdot 19^2 \simeq 2^{281.50}$$
$$p_{130} - 1 = 2 \cdot 3 \cdot 5 \cdot 59 \cdot 191 \cdot 2797 \cdot 16585601 \cdot 201574719984723380928407959307 \cdot q_{130}$$
$$q_{130} = 1733124013302036320718171822563477047667 \simeq 2^{130.35}$$

$$p_{117} + 1 = 2^{247} \cdot 79 \simeq 2^{253.34}$$
$$p_{117} - 1 = 2 \cdot 3 \cdot 5 \cdot 2903 \cdot 1924673583633629 \cdot 634009940699607211039 \cdot q_{117}$$
$$q_{117} = 168118140144706967996895604212334429 \simeq 2^{117.01}$$

- Finding *good* "SQIPrime2D-friendly" primes requires a lot of factoring.

Parameters

- On average, "SQIPrime2D-friendly" prime for security λ are of size $2\lambda + 4\log_2(\lambda)$ bits.
    - Finding *good* primes is a search for statistical anomalies:

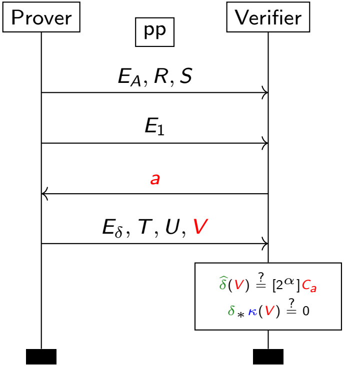
$$p_{130} + 1 = 2^{273} \cdot 19^2 \simeq 2^{281.50}$$
$$p_{130} - 1 = 2 \cdot 3 \cdot 5 \cdot 59 \cdot 191 \cdot 2797 \cdot 16585601 \cdot 2015747199847233809284079593 07 \cdot q_{130}$$
$$q_{130} = 17331240133020363207181718225634770476 67 \simeq 2^{130.35}$$

$$p_{117} + 1 = 2^{247} \cdot 79 \simeq 2^{253.34}$$
$$p_{117} - 1 = 2 \cdot 3 \cdot 5 \cdot 2903 \cdot 1924673583633629 \cdot 634009940699607211039 \cdot q_{117}$$
$$q_{117} = 168118140144706967996895604212334429 \simeq 2^{117.01}$$

- Finding *good* "SQIPrime2D-friendly" primes requires a lot of factoring.

SQIPrime as a DSA

- Using Fiat-Shamir [FS86], SQIPrime induces a quantum resistant DSA.

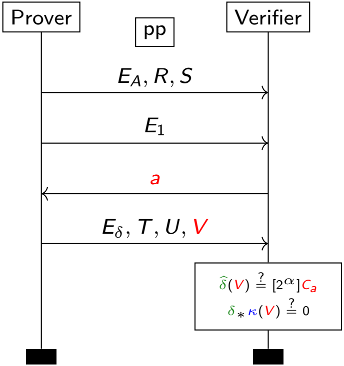


Scheme	pk	sign	sign (comp.)
SQISign	64	322	177
SQISignHD	64	208	109
SQIPrime4D	192	272	240
SQIPrime2D (p_{117})	191	320	299

- SQIPrime is not large, it is hard to compress.
- Still smaller than all non-isogeny based DSA.

SQIPrime as a DSA

- Using Fiat-Shamir [FS86], SQIPrime induces a quantum resistant DSA.



Scheme	pk	sign	sign (comp.)
SQISign	64	322	177
SQISignHD	64	208	109
SQIPrime4D	192	272	240
SQIPrime2D (p_{117})	191	320	299

- SQIPrime is not large, it is hard to compress.
- Still smaller than all non-isogeny based DSA.

Efficiency

- Following [DMPR23], it is clear that SQIPrime is quite efficient.
  - For KeyGen.
  - For Signature.
  - For Verification.
- Implementation not yet finalized.

Scheme ($\lambda = 128$)		2	3	(2,2)	(2,2,2,2)
SQISignHD	KeyGen	378	234	-	-
	Sign	252	312	-	-
	Verif	-	78	-	142
SQIPrime4D	KeyGen	-	-	241	-
	Sign	-	-	241	-
	Verif	-	-	-	263
SQIPrime2D (p_{130})	KeyGen	-	-	273	-
	Sign	-	-	546	-
	Verif	-	-	273	-
SQIPrime2D (p_{117})	KeyGen	-	-	247	-
	Sign	-	-	494	-
	Verif	-	-	247	-

Scheme ($\lambda = 128$)		End(E_0)	dim 2	dim 4
SQIPrime4D	KeyGen	4	2	-
	Sign	4	2	-
	Verif	-	-	4
SQIPrime2D	KeyGen	2	4	-
	Sign	6	5	-
	Verif	-	1	-

Efficiency

- Following [DMPR23], it is clear that SQIPrime is quite efficient.
  - For KeyGen.
  - For Signature.
  - For Verification.
- Implementation not yet finalized.

Scheme ($\lambda = 128$)		2	3	(2,2)	(2,2,2,2)
SQISignHD	KeyGen	378	234	-	-
	Sign	252	312	-	-
	Verif	-	78	-	142
SQIPrime4D	KeyGen	-	-	241	-
	Sign	-	-	241	-
	Verif	-	-	-	263
SQIPrime2D (p_{130})	KeyGen	-	-	273	-
	Sign	-	-	546	-
	Verif	-	-	273	-
SQIPrime2D (p_{117})	KeyGen	-	-	247	-
	Sign	-	-	494	-
	Verif	-	-	247	-

Scheme ($\lambda = 128$)		End(E_0)	dim 2	dim 4
SQIPrime4D	KeyGen	4	2	-
	Sign	4	2	-
	Verif	-	-	4
SQIPrime2D	KeyGen	2	4	-
	Sign	6	5	-
	Verif	-	1	-

Efficiency

- Following [DMPR23], it is clear that SQIPrime is quite efficient.
  - For KeyGen.
  - For Signature.
  - For Verification.
- Implementation not yet finalized.

Scheme ($\lambda = 128$)		2	3	(2,2)	(2,2,2,2)
SQISignHD	KeyGen	378	234	-	-
	Sign	252	312	-	-
	Verif	-	78	-	142
SQIPrime4D	KeyGen	-	-	241	-
	Sign	-	-	241	-
	Verif	-	-	-	263
SQIPrime2D (p_{130})	KeyGen	-	-	273	-
	Sign	-	-	546	-
	Verif	-	-	273	-
SQIPrime2D (p_{117})	KeyGen	-	-	247	-
	Sign	-	-	494	-
	Verif	-	-	247	-

Scheme ($\lambda = 128$)		End(E_0)	dim 2	dim 4
SQIPrime4D	KeyGen	4	2	-
	Sign	4	2	-
	Verif	-	-	4
SQIPrime2D	KeyGen	2	4	-
	Sign	6	5	-
	Verif	-	1	-

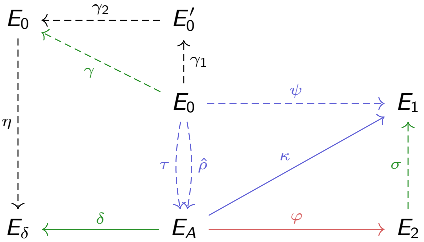
Efficiency

- Following [DMPR23], it is clear that SQIPrime is quite efficient.
  - For KeyGen.
  - For Signature.
  - For Verification.
- Implementation not yet finalized.

Scheme ($\lambda = 128$)		2	3	(2,2)	(2,2,2,2)
SQISignHD	KeyGen	378	234	-	-
	Sign	252	312	-	-
	Verif	-	78	-	142
SQIPrime4D	KeyGen	-	-	241	-
	Sign	-	-	241	-
	Verif	-	-	-	263
SQIPrime2D (p_{130})	KeyGen	-	-	273	-
	Sign	-	-	546	-
	Verif	-	-	273	-
SQIPrime2D (p_{117})	KeyGen	-	-	247	-
	Sign	-	-	494	-
	Verif	-	-	247	-

Scheme ($\lambda = 128$)		End(E_0)	dim 2	dim 4
SQIPrime4D	KeyGen	4	2	-
	Sign	4	2	-
	Verif	-	-	4
SQIPrime2D	KeyGen	2	4	-
	Sign	6	5	-
	Verif	-	1	-

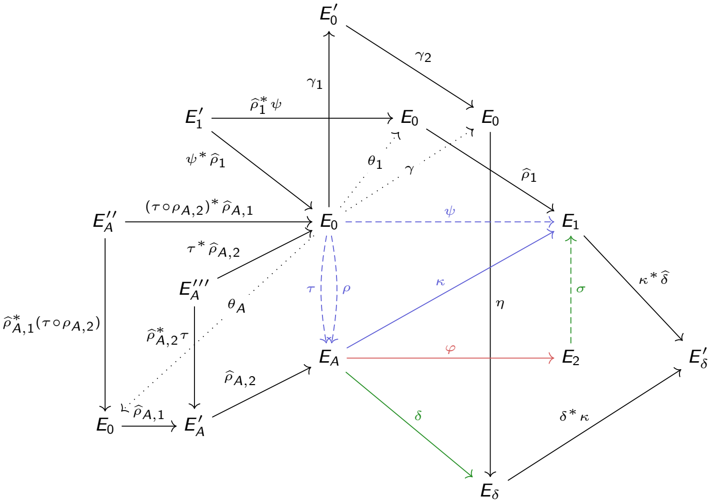
Thank you for listening !



Happy to discuss your comments and questions !!!

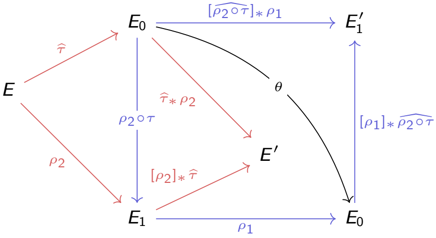
- More details in our paper (2024/773).

SQIPrime: A simple mechanism



To heuristically improve security of secret key, we propose **ExtKaniDoublePath**:

- $\theta \in \mathrm{End}(E_0)$ such that $n(\theta) = q(2^{\lceil \log_2(q) \rceil} - q)\left(2^\alpha - q(2^{\lceil \log_2(q) \rceil} - q)\right)$
- Use Kani's Lemma twice.
- $\rho = \rho_1 \circ \rho_2$ is such that $\deg(\rho) \simeq p^{3/2}$ (as opposed to p).



▶ Heuristically outputs almost all isogenies of degree q.

References I

Wouter Castryck and Thomas Decru, *An efficient key recovery attack on SIDH*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 423–447.

Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, *SQISign: compact post-quantum signatures from quaternions and isogenies*, 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 26, Springer, 2020, pp. 64–93.

Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski, *New algorithms for the deuring correspondence: towards practical and secure SQISign signatures*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 659–690.

Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski, *SQISignHD: New Dimensions in Cryptography*, Cryptology ePrint Archive, Paper 2023/436, 2023, `https://eprint.iacr.org/2023/436`.

References II

Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert, *An algorithmic approach to $(2, 2)$-isogenies in the theta model and applications to isogeny-based cryptography*, Cryptology ePrint Archive, Paper 2023/1747, 2023, https://eprint.iacr.org/2023/1747.

Amos Fiat and Adi Shamir, *How to prove yourself: Practical solutions to identification and signature problems*, Conference on the theory and application of cryptographic techniques, Springer, 1986, pp. 186–194.

Ernst Kani, *The number of curves of genus two with elliptic differentials.*, Walter de Gruyter, Berlin/New York Berlin, New York, 1997.

Antonin Leroux, *Verifiable random function from the Deuring correspondence and higher dimensional isogenies*, Cryptology ePrint Archive, Paper 2023/1251, 2023, https://eprint.iacr.org/2023/1251.

Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski, *A direct key recovery attack on SIDH*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 448–471.

References III

Kohei Nakagawa and Hiroshi Onuki, *QFESTA: Efficient Algorithms and Parameters for FESTA using Quaternion Algebras*, Cryptology ePrint Archive, Paper 2023/1468, 2023, `https://eprint.iacr.org/2023/1468`.

Damien Robert, *Breaking SIDH in polynomial time*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 472–503.