The state of the isogeny

Lorenz Panny

Technische Universität München

Workshop on the mathematics of post-quantum cryptography,
Zürich, 6 June 2025

Big picture

- Isogenies are a type of maps between elliptic curves.

Big picture

- Isogenies are a type of maps between elliptic curves.

- Sampling an isogeny *from* some curve is easy, recovering an isogeny *between* given curves seems very hard.

Big picture

- Isogenies are a type of maps between elliptic curves.
- Sampling an isogeny *from* some curve is easy, recovering an isogeny *between* given curves seems very hard.

⇝ ***Cryptography!***

Big picture

- Isogenies are a type of maps between elliptic curves.
- Sampling an isogeny *from* some curve is easy, recovering an isogeny *between* given curves seems very hard.

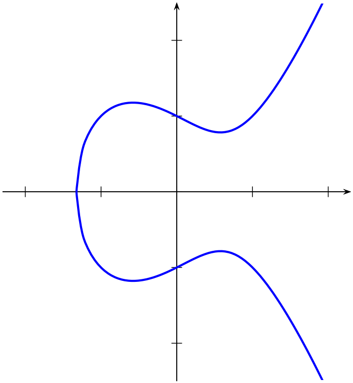
$\rightsquigarrow$ ***Cryptography!***

(Modern isogeny-based cryptography uses not just elliptic curves, but also higher-dimensional abelian varieties.)

Plan for this talk

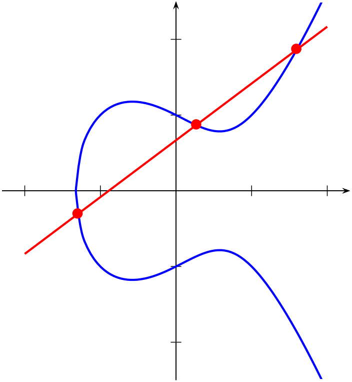
- Elliptic curves & isogenies.
- The SIKE attacks.
- Transcending to higher dimensions.
- Isogeny group actions.
- Signatures from isogenies.

Elliptic curves (picture over $\mathbb{R}$)



The elliptic curve $y^2 = x^3 - x + 1$ over $\mathbb{R}$.

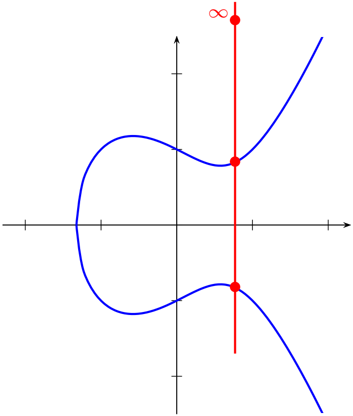
Elliptic curves (picture over $\mathbb{R}$)



Addition law:

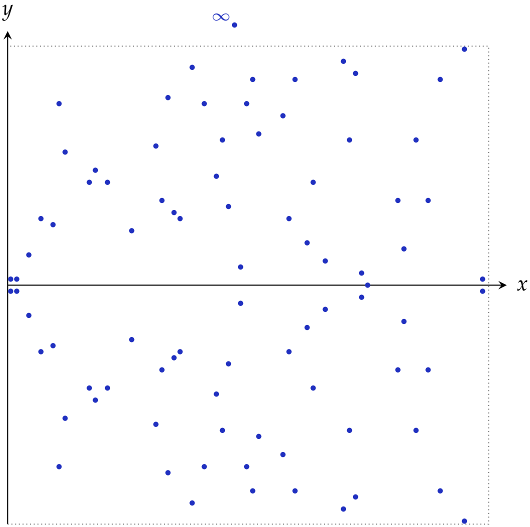
$$P + Q + R = \infty \quad \Longleftrightarrow \quad \{P, Q, R\} \text{ on a straight line.}$$

Elliptic curves (picture over $\mathbb{R}$)



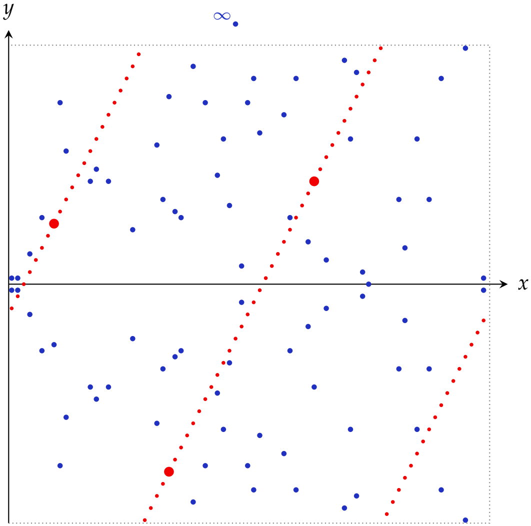
The *point at infinity* ∞ lies on every vertical line.

Elliptic curves (picture over $\mathbb{F}_p$)



The same curve $y^2 = x^3 - x + 1$ over the finite field $\mathbb{F}_{79}$.

Elliptic curves (picture over $\mathbb{F}_p$)



The addition law of $y^2 = x^3 - x + 1$ over the finite field $\mathbb{F}_{79}$.

Isogenies

Isogenies

...are just fancily-named

nice maps

between elliptic curves.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

The kernel of an isogeny $\varphi\colon E \to E'$ is $\{P \in E \,:\, \varphi(P) = \infty\}$.
The degree of a separable* isogeny is the size of its kernel.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

The kernel of an isogeny $\varphi \colon E \to E'$ is $\{P \in E : \varphi(P) = \infty\}$.
The degree of a separable* isogeny is the size of its kernel.

Generic example: $(x, y) \mapsto \left(\frac{x^3-4x^2+30x-12}{(x-2)^2}, \frac{x^3-6x^2-14x+35}{(x-2)^3} \cdot y \right)$
defines a degree-3 isogeny of the elliptic curves

$$\{y^2 = x^3 + x\} \longrightarrow \{y^2 = x^3 - 3x + 3\}$$

over $\mathbb{F}_{71}$. Its kernel is $\{(2, 9), (2, -9), \infty\}$.

Isogenies and kernels

For any finite subgroup G of E, there exists a unique* separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

Isogenies and kernels

For any finite subgroup G of E, there exists a unique* separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

Isogenies and kernels

For any finite subgroup G of E, there exists a unique* separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

Isogenies and kernels

For any finite subgroup G of E, there exists a unique* separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

- We have formulas to compute and evaluate isogenies.
  (...but they are only efficient for "small" degrees!)

Isogenies and kernels

For any finite subgroup G of E, there exists a unique* separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

- We have formulas to compute and evaluate isogenies.
  (...but they are only efficient for "small" degrees!)

$\rightsquigarrow$ Decompose large-degree isogenies into prime steps.
That is, walk in an isogeny graph.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \Bigg(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q) \big), \\ y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q) \big) \Bigg)$$

defines an isogeny of elliptic curves with kernel G.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \Bigg(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q) \big), \\ y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q) \big) \Bigg)$$

defines an isogeny of elliptic curves with kernel G.

This leads to formulas for

- computing the defining equation of E/G;
- evaluating the isogeny $E \to E/G$ at a point.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \Bigg(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q) \big), \\ y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q) \big) \Bigg)$$

defines an isogeny of elliptic curves with kernel G.

This leads to formulas for

- computing the defining equation of E/G;
- evaluating the isogeny $E \to E/G$ at a point.

Complexity: $\Theta(\#G)$ $\rightsquigarrow$ only suitable for small degrees.

The $\sqrt{\text{é}}$lu algorithm reduces the cost to $\widetilde{\mathcal{O}}(\sqrt{\#G})$.

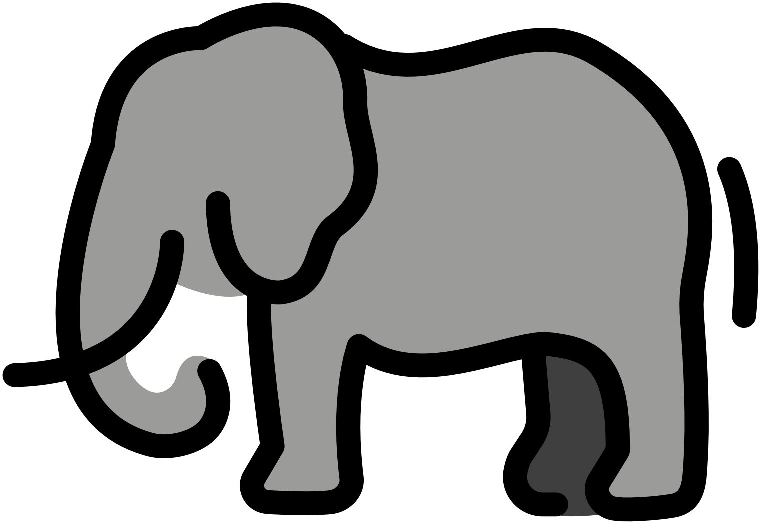
“Computing an isogeny”

“Computing an isogeny”

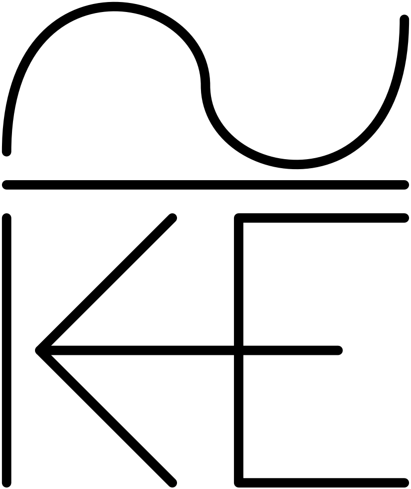
Keep in mind: Constructing isogenies $E \to \_$ is (usually) easy, constructing an isogeny $E \to E'$ given (E, E') is (usually) hard.

Plan for this talk

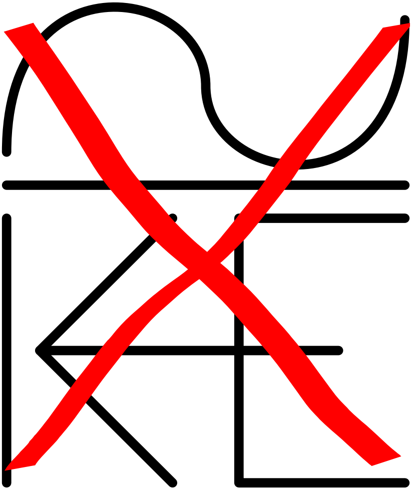
- Elliptic curves & isogenies. ✓
- The SIKE attacks.
- Transcending to higher dimensions.
- Isogeny group actions.
- Signatures from isogenies.



SIDH/SIKE



SIDH/SIKE



SIDH/SIKE

...*was* a well-known isogeny-based key exchange scheme:

- The “isogeny poster child” from ≈ 2011 to ≈ 2022.
- Part of NISTPQC, which found no security flaws.

SIDH/SIKE

...*was* a well-known isogeny-based key exchange scheme:

- The "isogeny poster child" from ≈ 2011 to ≈ 2022.
- Part of NISTPQC, which found no security flaws.

It was catastrophically broken in 2022.

Isogeny-based key exchange: High-level view

E

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E_A \\
E_B &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E_A \arrow[dl, dashed, shift left] \\
E_B \arrow[ur, dashed, shift left] &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A \colon E \to E_A$ and $\varphi_B \colon E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \arrow[ddll, dashed, shift left] \\
& & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] \arrow[uurr, dashed, shift left] & E_{BA} &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice <u>somehow</u> finds a “parallel” $\varphi_{A'} : E_B \to E_{BA}$, and
  Bob <u>somehow</u> finds $\varphi_{B'} : E_A \to E_{AB}$,

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \arrow[ddll, dashed] \\
 & & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] \arrow[uurr, dashed] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice <u>somehow</u> finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob <u>somehow</u> finds $\varphi_{B'} : E_A \to E_{AB}$, such that $E_{AB} \cong E_{BA}$.

How to find "parallel" isogenies?

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, swap, "\varphi_B"] & & E_A \arrow[d, thick, "\varphi'_B"] \\
 & & E_{AB} \\
E_B \arrow[r, thick, swap, "\varphi'_A"] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}$$

How to find “parallel” isogenies?

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \\
 & & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}$$

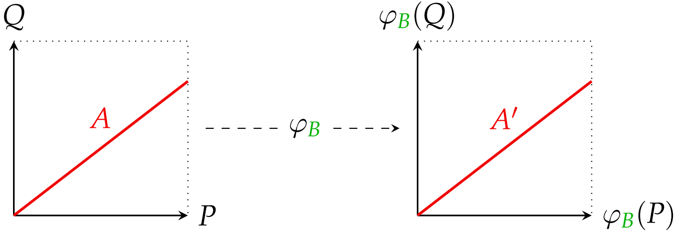
SIKE’s solution:

The isogeny φ_B is a group homomorphism! (and $A \cap B = \{\infty\}$)

How to find "parallel" isogenies?

SIKE's solution:

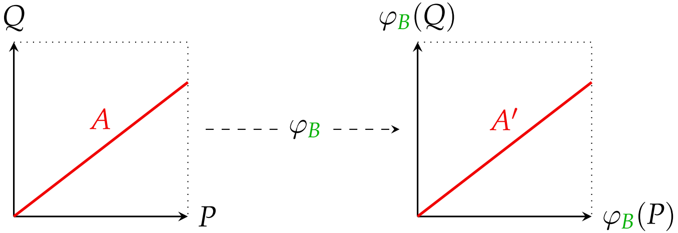
The isogeny φ_B is a group homomorphism! (and $A \cap B = \{\infty\}$)



How to find "parallel" isogenies?

SIKE's solution:

The isogeny φ_B is a group homomorphism! (and $A \cap B = \{\infty\}$)



- Alice picks A as $\langle P + [a]Q \rangle$ for fixed public $P, Q \in E$.
- Bob includes $\varphi_B(P)$ and $\varphi_B(Q)$ in his public key.

$\Longrightarrow$ Now Alice can compute A' as $\langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$.

(Similarly for Bob.)

The SIDH/SIKE attacks

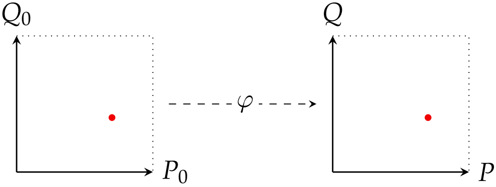
- **Not** a case of everyone overlooking something stupid.

The SIDH/SIKE attacks

- **<u>Not</u>** a case of everyone overlooking something stupid.
- The attack uses an unexpected profound new technique.

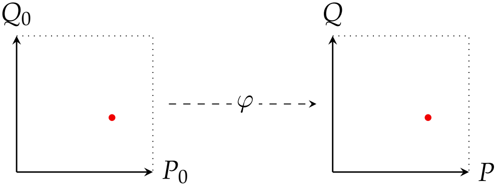
The SIDH/SIKE attacks

- **Not** a case of everyone overlooking something stupid.
- The attack uses an unexpected profound new technique.
- SIKE revealed how a secret isogeny acts on lots of points.



The SIDH/SIKE attacks

- **Not** a case of everyone overlooking something stupid.
- The attack uses an unexpected profound new technique.
- SIKE revealed how a secret isogeny acts on lots of points.

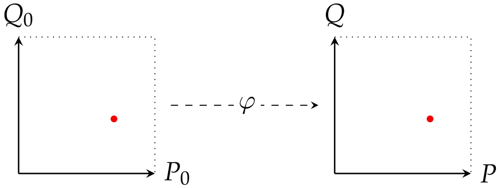


This **isogeny interpolation** problem turns out to be **easy!**

(at least in some cases — it's complicated, etc., etc.)

The SIDH/SIKE attacks

- **Not** a case of everyone overlooking something stupid.
- The attack uses an unexpected profound new technique.
- SIKE revealed how a secret isogeny acts on lots of points.



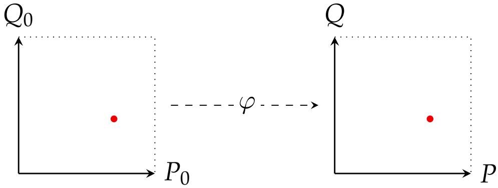
This **isogeny interpolation** problem turns out to be **easy!**

(at least in some cases — it's complicated, etc., etc.)

- It has since found groundbreaking constructive uses.
- The general **isogeny problem** is entirely unaffected!

The SIDH/SIKE attacks

- **Not** a case of everyone overlooking something stupid.
- The attack uses an unexpected profound new technique.
- SIKE revealed how a secret isogeny acts on lots of points.



This **isogeny interpolation** problem turns out to be **easy!**

(at least in some cases — it's complicated, etc., etc.)

- It has since found groundbreaking constructive uses.
- The general **isogeny problem** is entirely unaffected!

⇝ The best thing to ever happen to isogenies! 🎉

Plan for this talk

- Elliptic curves & isogenies. ✓
- The SIKE attacks. ✓
- Transcending to higher dimensions.
- Isogeny group actions.
- Signatures from isogenies.

Transcending to higher dimensions

- Fallout from the SIDH attack: New tools.

  "One man's a-ttack is another man's a-treasure."

Transcending to higher dimensions

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Main technique underlying attack:

Computing isogenies between *products* of elliptic curves

Transcending to higher dimensions

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Main technique underlying attack:

Computing isogenies between *products* of elliptic curves

- The product $E \times E'$ is an abelian *surface*.

Transcending to higher dimensions

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Main technique underlying attack:

Computing isogenies between *products* of elliptic curves

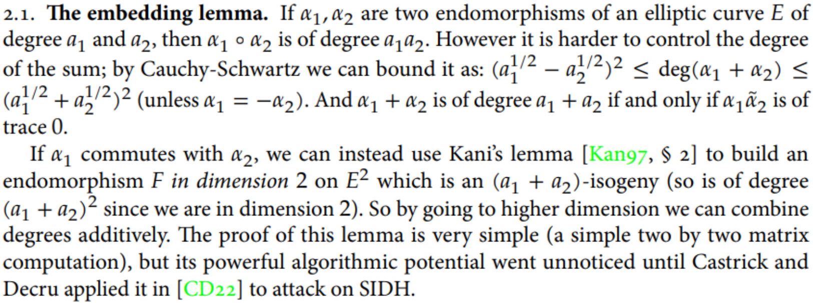
- The product $E \times E'$ is an abelian *surface*.
- Similar to elliptic curves in many ways:
  - Points form an abelian group.
  - Similar group structure, but more components.
  - Can define isogenies from kernel subgroups.

The embedding lemma

- Fallout from the SIDH attack: New tools.

The embedding lemma

- Fallout from the SIDH attack: New tools.



2.1. **The embedding lemma.** If α_1, α_2 are two endomorphisms of an elliptic curve E of degree a_1 and a_2, then $\alpha_1 \circ \alpha_2$ is of degree $a_1 a_2$. However it is harder to control the degree of the sum; by Cauchy-Schwartz we can bound it as: $(a_1^{1/2} - a_2^{1/2})^2 \leq \deg(\alpha_1 + \alpha_2) \leq (a_1^{1/2} + a_2^{1/2})^2$ (unless $\alpha_1 = -\alpha_2$). And $\alpha_1 + \alpha_2$ is of degree $a_1 + a_2$ if and only if $\alpha_1 \tilde{\alpha}_2$ is of trace 0.

If α_1 commutes with α_2, we can instead use Kani's lemma [Kan97, § 2] to build an endomorphism F *in dimension* 2 on E^2 which is an $(a_1 + a_2)$-isogeny (so is of degree $(a_1 + a_2)^2$ since we are in dimension 2). So by going to higher dimension we can combine degrees additively. The proof of this lemma is very simple (a simple two by two matrix computation), but its powerful algorithmic potential went unnoticed until Castrick and Decru applied it in [CD22] to attack on SIDH.

— Damien Robert [ePrint 2022/1704]

The embedding lemma

Consider a commutative diagram of isogenies

$$\begin{tikzcd} E \arrow[r, "\varphi"] \arrow[d, "\psi"'] & E' \arrow[d, "\psi'"] \\ E'' \arrow[r, "\varphi'"'] & E''' \end{tikzcd}$$

where $a := \deg \varphi$ and $b := \deg \psi$ are coprime, and let $N := a + b$.

Lemma. Then

$$\Phi := \begin{pmatrix} \varphi & \widehat{\psi}' \\ -\psi & \widehat{\varphi}' \end{pmatrix} \colon (P, Q) \mapsto \big(\varphi(P) + \widehat{\psi}'(Q), -\psi(P) + \widehat{\varphi}'(Q)\big)$$

defines an N-isogeny $E \times E''' \to E' \times E''$.

Its kernel is $\ker(\Phi) = \big\{(\widehat{\varphi}(T), \psi'(T)) \mid T \in E'[N]\big\}$.

The HD representation

...is an efficient representation of *any (!)* isogeny between two elliptic curves.

(Recall: Using Vélu/$\sqrt{\text{é}}$lu techniques, only smooth-degree isogenies are efficient.)

The HD representation

...is an efficient representation of *any (!)* isogeny between two elliptic curves.

(Recall: Using Vélu/√élu techniques, only smooth-degree isogenies are efficient.)

💡 Simply encode $\varphi\colon E \to E'$ as a higher-dimensional isogeny

$$\Phi := \begin{pmatrix} \varphi & \widehat{\psi'} \\ -\psi & \widehat{\varphi'} \end{pmatrix} \colon E \times E''' \to E' \times E''\,.$$

The HD representation

...is an efficient representation of *any (!)* isogeny between two elliptic curves.

(Recall: Using Vélu/$\sqrt{\text{é}}$lu techniques, only smooth-degree isogenies are efficient.)

💡 Simply encode $\varphi\colon E \to E'$ as a higher-dimensional isogeny

$$\Phi := \begin{pmatrix} \varphi & \widehat{\psi'} \\ -\psi & \widehat{\varphi'} \end{pmatrix} \colon E \times E''' \to E' \times E''\,.$$

+ For full generality, need to embed in dimension 8.

The HD representation

...is an efficient representation of *any (!)* isogeny between two elliptic curves.

(Recall: Using Vélu/$\sqrt{\text{é}}$lu techniques, only smooth-degree isogenies are efficient.)

💡 Simply encode $\varphi\colon E \to E'$ as a higher-dimensional isogeny

$$\Phi := \begin{pmatrix} \varphi & \widehat{\psi'} \\ -\psi & \widehat{\varphi'} \end{pmatrix} \colon E \times E''' \to E' \times E''\,.$$

+ For full generality, need to embed in dimension 8.

⚠ Requires isogeny formulas for principally polarized abelian varieties of dimension ≥ 2. Highly non-trivial matter, but fundamentally doable and efficient.

Plan for this talk

- Elliptic curves & isogenies. ✓
- The SIKE attacks. ✓
- Transcending to higher dimensions. ✓
- Isogeny group actions.
- Signatures from isogenies.

How to find "parallel" isogenies?

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \\
& & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}$$

How to find “parallel” isogenies?

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \\
& & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}$$

CSIDH’s solution:

Use special isogenies φ_A which can be transported to the curve E_B totally independently of the secret isogeny φ_B.

(Similarly with reversed roles, of course.)

CSIDH [ˈsiːˌsaɪd]

[Castryck–Lange–Martindale–Panny–Renes 2018]

“Special” isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

“Special” isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ For every $\ell \mid (p+1)$ exists a unique order-ℓ subgroup H_ℓ.

“Special” isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ For every $\ell \mid (p+1)$ exists a unique order-ℓ subgroup H_ℓ.

$\rightsquigarrow$ For all such E can canonically find an isogeny $\varphi_\ell \colon E \to E'$.

“Special” isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ For every $\ell \mid (p+1)$ exists a unique order-ℓ subgroup H_ℓ.

$\leadsto$ For all such E can canonically find an isogeny $\varphi_\ell \colon E \to E'$.

We consider prime ℓ and refer to φ_ℓ as a “special” isogeny.

Cycles from "special" isogenies

What happens when we iterate such a "special" isogeny?

Cycles from “special” isogenies

What happens when we iterate such a “special” isogeny?

$$\begin{tikzcd}
 & & & & & & \cdots \arrow[dll] & E_{\ell^6} \arrow[l] \\
 & & & & E_{\ell^{r-1}} \arrow[dl] & & & \\
 & & & E_{\ell^r} \arrow[dd] & & & & E_{\ell^5} \arrow[uu] \\
 & & & & & E_{\ell^4} \arrow[urr] & & \\
E \arrow[r] & E_\ell \arrow[r] & E_{\ell^2} \arrow[r] & E_{\ell^3} \arrow[urr] & & & &
\end{tikzcd}$$

Cycles from "special" isogenies

What happens when we iterate such a "special" isogeny?

$$\begin{tikzcd}
 & & & & E_{\ell^{r-1}} \arrow[dl] & \cdots \arrow[l] & E_{\ell^6} \arrow[l] \\
 & & & E_{\ell^r} \arrow[dd] & & & E_{\ell^5} \arrow[u] \\
 & & & & & E_{\ell^4} \arrow[ur] & \\
E \arrow[r] & E_{\ell} \arrow[r] & E_{\ell^2} \arrow[r] & E_{\ell^3} \arrow[urr] & & &
\end{tikzcd}$$

- Fact: Each curve has only one other rational ℓ-isogeny.

Cycles from “special” isogenies

What happens when we iterate such a “special” isogeny?

$$\begin{tikzcd}
 & & & & & \cdots \arrow[dll] & E_{\ell^6} \arrow[l] \\
 & & & E_{\ell^{r-1}} \arrow[d] & & & \\
 & & & E_{\ell^r} \arrow[dd] & & & E_{\ell^5} \arrow[uu] \\
 & & & & E_{\ell^4} \arrow[urr] & & \\
E \arrow[r] & E_{\ell} \arrow[r] & E_{\ell^2} \arrow[r] & E_{\ell^3} \arrow[ur] & & &
\end{tikzcd}$$

- Fact: Each curve has only one other rational ℓ-isogeny.

!! Reverse arrows are unique; the “tail” $E \to E_{\ell^3}$ cannot exist.

Cycles from “special” isogenies

What happens when we iterate such a “special” isogeny?

$$\begin{tikzcd}
 & & & E_{\ell^{r-1}} \arrow[dl] & & \cdots \arrow[ll] & & E_{\ell^6} \arrow[ll] \\
 & & E_{\ell^r} \arrow[ddr] & & & & & E_{\ell^5} \arrow[u] \\
 & & & & & E_{\ell^4} \arrow[urr] & & \\
E \arrow[r] & E_\ell \arrow[r] & E_{\ell^2} \arrow[r] & E_{\ell^3} \arrow[urr] & & & &
\end{tikzcd}$$

- Fact: Each curve has only one other rational ℓ-isogeny.

!! Reverse arrows are unique; the “tail” $E \to E_{\ell^3}$ cannot exist.

$\Longrightarrow$ The “special” isogenies φ_ℓ form isogeny cycles!

Compatible cycles from “special” isogenies

What happens when we compose those “special” isogenies?

Compatible cycles from “special” isogenies

What happens when we compose those “special” isogenies?

$$\begin{tikzcd}
 & & E_{m\ell} & E_{\ell m} \\
E_m \arrow[urr, "\varphi'_\ell"] & & & \\
 & & E_\ell \arrow[uur, "\varphi_m"] & \\
E \arrow[uu, "\varphi'_m"] \arrow[uurr, "\varphi_\ell"] & & &
\end{tikzcd}$$

Compatible cycles from “special” isogenies

What happens when we compose those “special” isogenies?

$$\begin{tikzcd}
 & E_{m\ell} & E_{\ell m} \\
E_m \arrow[ur, "\varphi'_\ell"] & E_\ell \arrow[ur, "\varphi_m"] & \\
E \arrow[u, "\varphi'_m"] \arrow[ur, "\varphi_\ell"] & &
\end{tikzcd}$$

- Fact: $\ker(\varphi'_\ell \circ \varphi'_m) = \ker(\varphi_m \circ \varphi_\ell) = \langle \ker \varphi_\ell, \ker \varphi'_m \rangle$.

Compatible cycles from “special” isogenies

What happens when we compose those “special” isogenies?

$$\begin{tikzcd}
E_m \arrow[r, "\varphi'_\ell"] & E_{\ell m} \\
E \arrow[u, "\varphi'_m"] \arrow[r, "\varphi_\ell"'] & E_\ell \arrow[u, "\varphi_m"']
\end{tikzcd}$$

- Fact: $\ker(\varphi'_\ell \circ \varphi'_m) = \ker(\varphi_m \circ \varphi_\ell) = \langle \ker \varphi_\ell, \ker \varphi'_m \rangle$.

!! The order cannot matter $\Longrightarrow$ cycles must be compatible.

CSIDH in one slide

CSIDH in one slide

- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.

CSIDH in one slide

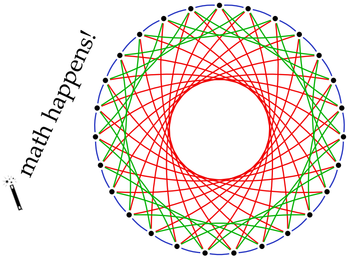
- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.

CSIDH in one slide

- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.
- Look at the "special" ℓ_i-isogenies within X.

CSIDH in one slide

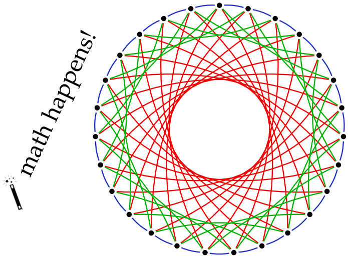
- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.
- Look at the "special" ℓ_i-isogenies within X.



$p = 419$
$\ell_1 = 3$
$\ell_2 = 5$
$\ell_3 = 7$

CSIDH in one slide

- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.
- Look at the "special" ℓ_i-isogenies within X.



$p = 419$

$\ell_1 = 3$

$\ell_2 = 5$

$\ell_3 = 7$

- Walking "left" and "right" on any ℓ_i-subgraph is efficient.

Walking in the CSIDH graph (in SageMath)

```
sage: E = EllipticCurve(GF(419^2), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
      over Finite Field in z2 of size 419^2
```

Walking in the CSIDH graph (in SageMath)

```
sage: E = EllipticCurve(GF(419^2), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
        over Finite Field in z2 of size 419^2
sage: while True:
....:     x = GF(419).random_element()
....:     try:
....:         P = E.lift_x(x)
....:     except ValueError: continue
....:     if P[1] in GF(419):   # "right" step: invert
....:         break
....:
sage: P
(218 : 403 : 1)
```

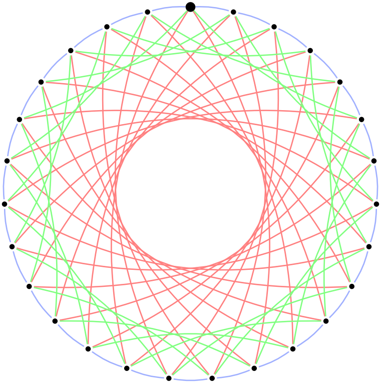
Walking in the CSIDH graph (in SageMath)

```
sage: E = EllipticCurve(GF(419^2), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
        over Finite Field in z2 of size 419^2
sage: while True:
....:     x = GF(419).random_element()
....:     try:
....:         P = E.lift_x(x)
....:     except ValueError: continue
....:     if P[1] in GF(419):   # "right" step: invert
....:         break
....:
sage: P
(218 : 403 : 1)
sage: P.order().factor()
2 * 3 * 7
sage: EE = E.isogeny_codomain(2*3*P)  # "left" 7-step
sage: EE
Elliptic Curve defined by y^2 = x^3 + 285*x + 87
        over Finite Field in z2 of size 419^2
```

CSIDH key exchange

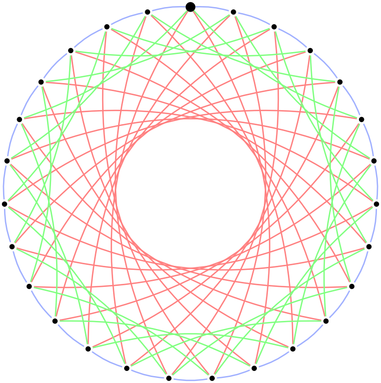
Alice

[+, +, −, −]



Bob

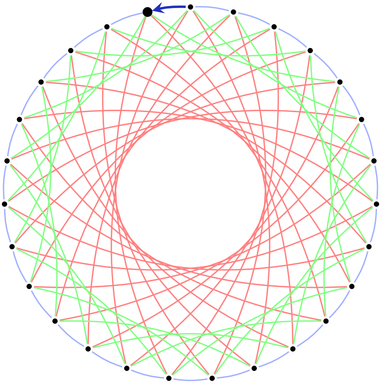
[−, +, −, −]



CSIDH key exchange

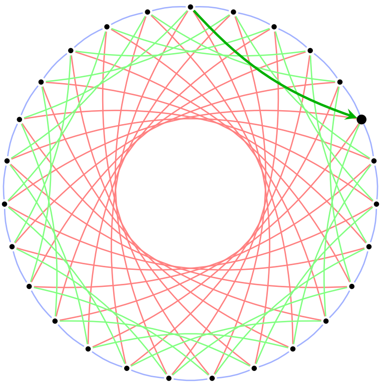
Alice

[+, +, −, −]
↑



Bob

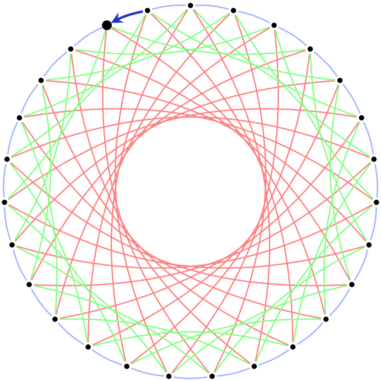
[−, +, −, −]
↑



CSIDH key exchange

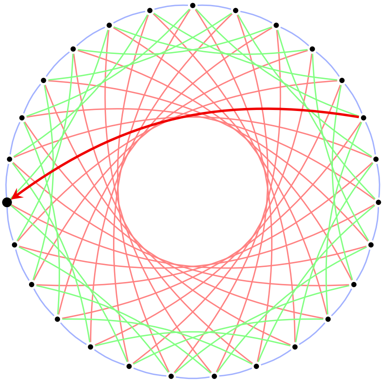
Alice

[+, +, −, −]
 $\uparrow$



Bob

[−, +, −, −]
 $\uparrow$



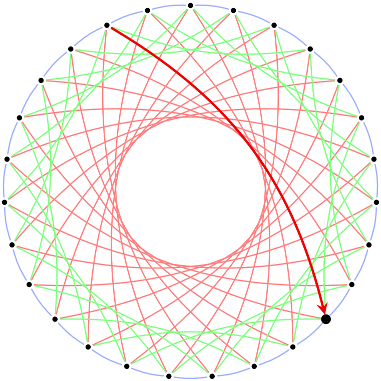
CSIDH key exchange

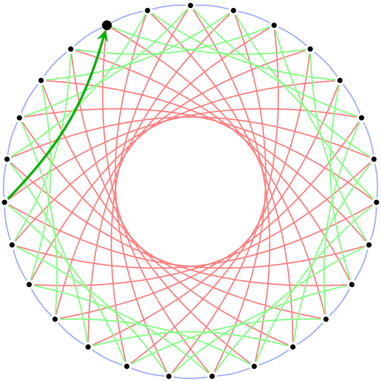
Alice

$[+, +, \underset{\uparrow}{-}, -]$

Bob

$[-, +, \underset{\uparrow}{-}, -]$

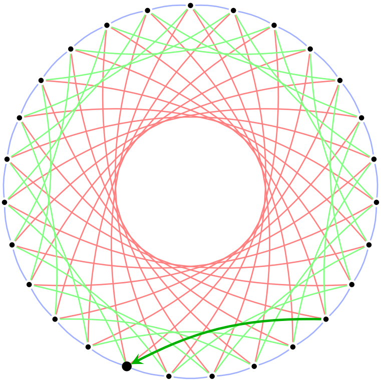




CSIDH key exchange

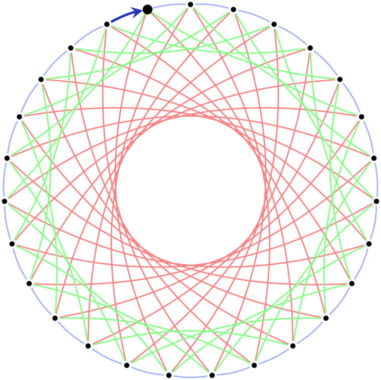
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



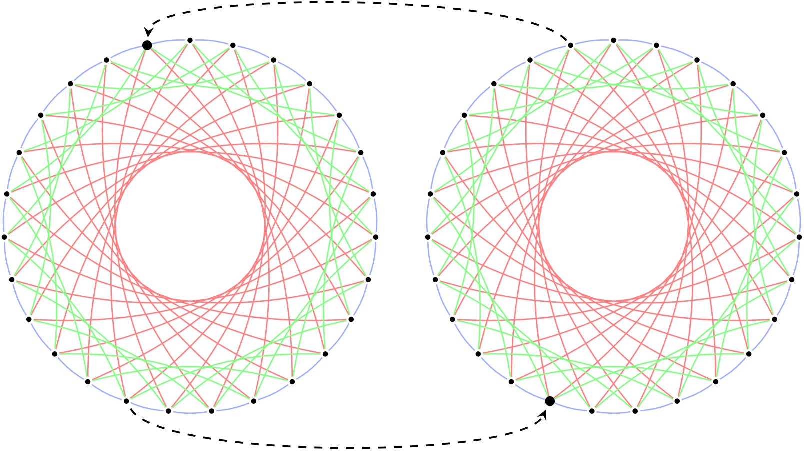
CSIDH key exchange

Alice

[+, +, −, −]

Bob

[−, +, −, −]

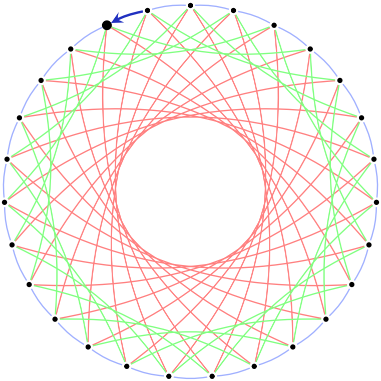


CSIDH key exchange

Alice

[+, +, −, −]

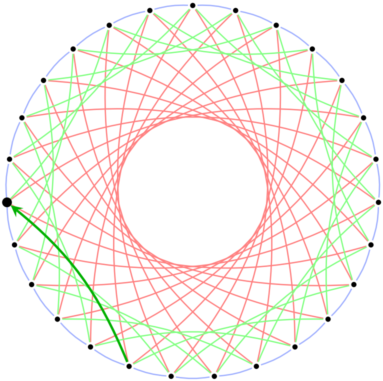
↑



Bob

[−, +, −, −]

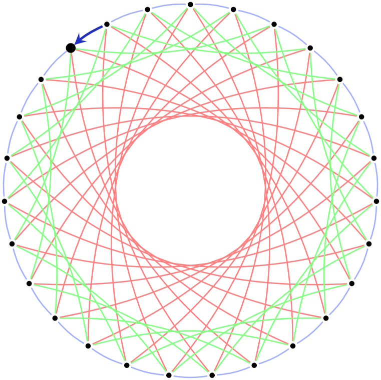
↑



CSIDH key exchange

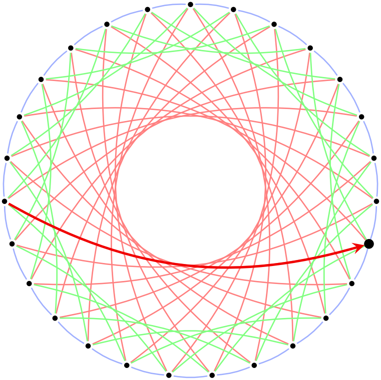
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



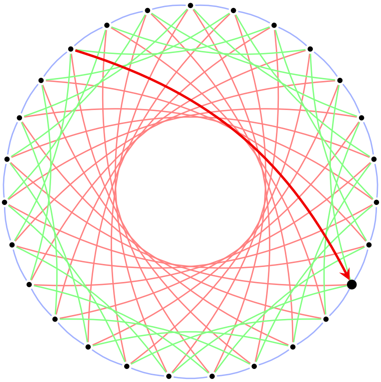
CSIDH key exchange

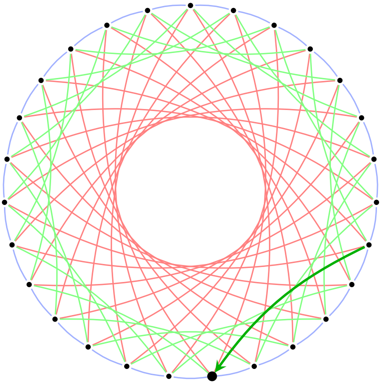
Alice

$[+, +, \underset{\uparrow}{-}, -]$

Bob

$[-, +, \underset{\uparrow}{-}, -]$





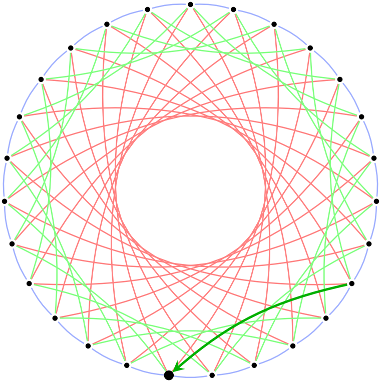
CSIDH key exchange

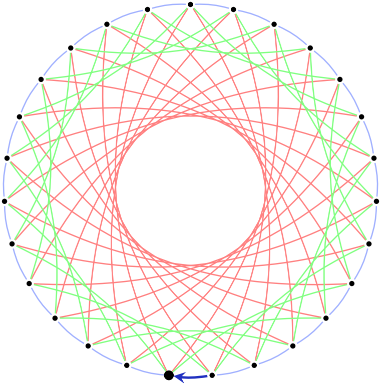
Alice

$[+, +, -, \underset{\uparrow}{-}]$

Bob

$[-, +, -, \underset{\uparrow}{-}]$

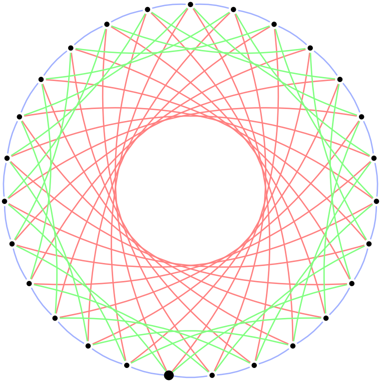




CSIDH key exchange

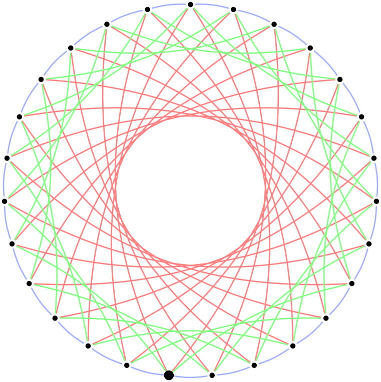
Alice

$[+, +, -, -]$



Bob

$[-, +, -, -]$



Action!

Cycles are compatible: [right then left] = [left then right]

Action! 🎬

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+, +, -, -, -, +, -, -]$ just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.

Action!

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+, +, -, -, -, +, -, -]$ just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.

There is a group action of $(\mathbb{Z}^n, +)$ on our set of curves X!

Action! 🎬

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+, +, -, -, -, +, -, -]$ just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.

There is a group action of $(\mathbb{Z}^n, +)$ on our set of curves X!

!! The set X is **finite** $\implies$ The action is **not free**.

There exist vectors $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.

Action! 🎬

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+,+,-,-,-,+,-,-]$ just becomes $(+1,\quad 0,-3) \in \mathbb{Z}^3$.

> There is a group action of $(\mathbb{Z}^n,+)$ on our set of curves X!

!! The set X is **finite** $\implies$ The action is **not free**.

There exist vectors $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.

Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

Action!

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+, +, -, -, -, +, -, -]$ just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.

There is a group action of $(\mathbb{Z}^n, +)$ on our set of curves X!

!! The set X is **finite** $\implies$ The action is **not free**.

There exist vectors $\underline{v} \in \mathbb{Z}^n \setminus \{0\}$ which act trivially.

Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

!! We understand the structure: By complex-multiplication theory, the quotient $\mathbb{Z}^n/\Lambda$ is the ideal-class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

Action!

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+,+,-,-,-,+,-,-]$ just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.

There is a group action of $(\mathbb{Z}^n, +)$ on our set of curves X!

!! The set X is **finite** $\Longrightarrow$ The action is **not free**.

There exist vectors $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.

Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

!! We understand the structure: By complex-multiplication theory, the quotient $\mathbb{Z}^n/\Lambda$ is the ideal-class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

!! This group characterizes *when two paths lead to the same curve*.

Action! 🎬

Cycles are compatible: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of total step counts for each ℓ_i.

Example: $[+,+,-,-,-,+,-,-]$ just becomes $(+1,\quad 0,-3) \in \mathbb{Z}^3$.

There is a group action of $(\mathbb{Z}^n, +)$ on our set of curves X!

!! The set X is **finite** $\Longrightarrow$ The action is **not free**.
There exist vectors $\underline{v} \in \mathbb{Z}^n \setminus \{0\}$ which act trivially.
Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

!! We understand the structure: By complex-multiplication theory, the quotient $\mathbb{Z}^n/\Lambda$ is the ideal-class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

!! This group characterizes *when two paths lead to the same curve*.

The lattice Λ is computable in subexponential time classically, and in polynomial time using a quantum computer.
It is used to construct more advanced schemes (*"CSI-FiSh"*).

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt{p})$; attacks are basically brute force.

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

$\Longrightarrow$ Key sizes: Public keys are 4λ bits for *classical* λ-bit security. (For λ-bit *quantum* security, need $\Theta(\lambda^2)$ bits.)

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

$\Longrightarrow$ Key sizes: Public keys are 4λ bits for *classical* λ-bit security.
(For λ-bit *quantum* security, need $\Theta(\lambda^2)$ bits.)

- Performance: Some tens of milliseconds per group-action evaluation at the 128-bit *classical* security level.

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

$\Longrightarrow$ Key sizes: Public keys are 4λ bits for *classical* λ-bit security.
(For λ-bit *quantum* security, need $\Theta(\lambda^2)$ bits.)

- Performance: Some tens of milliseconds per group-action evaluation at the 128-bit *classical* security level.

- 2023: "Clapoti" — a polynomial-time algorithm for arbitrary combinations of operations in the group and evaluations of the action. $\rightsquigarrow$ *"KLaPoTi"*, *"PEGASIS"*.
(Previously, only restricted sequences of operations were efficient.)

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

- The algorithm admits many different tradeoffs.
- Oracle calls are expensive.
- The sieving phase has classical *and* quantum operations.

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

- The algorithm admits many different tradeoffs.
- Oracle calls are expensive.
- The sieving phase has classical *and* quantum operations.

  How to compare costs?

  (Is one qubit operation $\approx$ one bit operation? a hundred? millions?)

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. (“oracle calls”)
2. Combine the results in a certain way. (“sieving”)

- The algorithm admits many different tradeoffs.
- Oracle calls are expensive.
- The sieving phase has classical *and* quantum operations.

  How to compare costs?

  (Is one qubit operation $\approx$ one bit operation? a hundred? millions?)

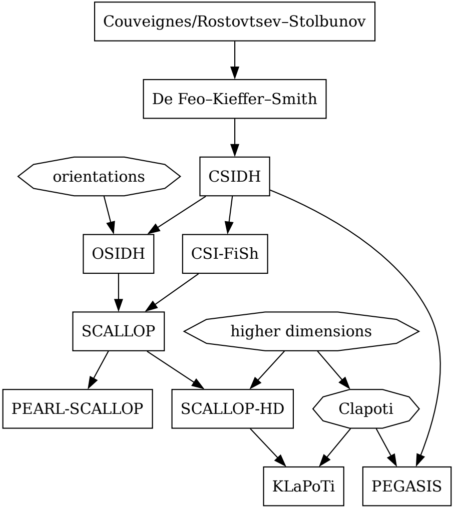
$\Longrightarrow$ Security estimates for CSIDH & friends vary wildly. ⚠

Oriented isogenies

There are many ways of building isogeny group actions.

Oriented isogenies

There are many ways of building isogeny group actions.



Plan for this talk

- Elliptic curves & isogenies. ✓
- The SIKE attacks. ✓
- Transcending to higher dimensions. ✓
- Isogeny group actions. ✓
- Signatures from isogenies.

SQIsign: What?

https://sqisign.org

SQIsign: What?

`https://sqisign.org`

- A new-ish and very hot post-quantum signature scheme.
- Based on super cool mathematics. ☺

More “special” isogenies

- Earlier: “Special” isogenies φ_ℓ with rational kernel points.

More “special” isogenies

- Earlier: “Special” isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is the Frobenius endomorphism $\pi\colon (x, y) \mapsto (x^p, y^p)$.)

More “special” isogenies

- Earlier: “Special” isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is the Frobenius endomorphism $\pi\colon (x,y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x,y) \mapsto (-x, \sqrt{-1} \cdot y)$.

More “special” isogenies

- Earlier: “Special” isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is the Frobenius endomorphism $\pi\colon (x,y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x,y) \mapsto (-x, \sqrt{-1} \cdot y)$.

- Extremely non-obvious fact in this setting:

Every isogeny $\varphi\colon E \to E'$ comes from an ideal $I_\varphi \subseteq \operatorname{End}(E)$.

More “special” isogenies

- <u>Earlier:</u> “Special” isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is the Frobenius endomorphism $\pi\colon (x, y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$.

- Extremely non-obvious fact in this setting:

<u>Every</u> isogeny $\varphi\colon E \to E'$ comes from an ideal $I_\varphi \subseteq \mathrm{End}(E)$.

☺ We understand the structure of $\mathrm{End}(E)$.

More “special” isogenies

- Earlier: “Special” isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is the Frobenius endomorphism $\pi\colon (x, y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$.

- Extremely non-obvious fact in this setting:

Every isogeny $\varphi\colon E \to E'$ comes from an ideal $I_\varphi \subseteq \mathrm{End}(E)$.

☺ We understand the structure of $\mathrm{End}(E)$.

☺ We understand how I_φ, I_ψ relate for isogenies $\varphi, \psi\colon E \to E'$.

$\rightsquigarrow$ one-sided ideal class *set* of $\mathrm{End}(E)$, etc.

The Deuring correspondence

...is the formal version of what I just said.

The Deuring correspondence

...is the formal version of what I just said.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules
  with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

The Deuring correspondence

...is the formal version of what I just said.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules
  with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

A strong connection between two $\overset{\textit{a priori}}{\curlyvee}$ very different worlds:

The Deuring correspondence

...is the formal version of what I just said.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules
  with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

A strong connection between two $\overset{\textit{a priori}}{\curlyvee}$ very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.

The Deuring correspondence

...is the formal version of what I just said.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules
  with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

A strong connection between two$\overset{\textit{a priori}}{\curlyvee}$very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

The Deuring correspondence

...is the formal version of what I just said.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules
  with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

A strong connection between two $\overset{\textit{a priori}}{\curlyvee}$ very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

Isogenies become "connecting ideals" in quaternion land.

The Deuring correspondence

...is the formal version of what I just said.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \operatorname{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\operatorname{End}(E_0)$-modules
  with nonzero left $\operatorname{End}(E_0)$-module homomorphisms.

A strong connection between two${}^{\overset{\textit{a priori}}{\curlyvee}}$very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

Isogenies become “connecting ideals” in quaternion land.

☺ One direction is easy, the other seems hard! $\rightsquigarrow$ *Cryptography!*

The Deuring correspondence (examples)

Let $p = 7799999$ and let $\mathbf{i}, \mathbf{j}$ satisfy $\mathbf{i}^2 = -1,\ \mathbf{j}^2 = -p,\ \mathbf{ji} = -\mathbf{ij}$.

The ring $\mathcal{O}_0 = \mathbb{Z} \oplus \mathbb{Z}\,\mathbf{i} \oplus \mathbb{Z}\,\frac{\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{1+\mathbf{ij}}{2}$
corresponds to the curve $E_0\colon\ y^2 = x^3 + x$.

The ring $\mathcal{O}_1 = \mathbb{Z} \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{4947\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947+32631010\mathbf{i}+\mathbf{ij}}{9894}$
corresponds to the curve $E_1\colon\ y^2 = x^3 + 1$.

The ideal $I = \mathbb{Z}\,4947 \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{598+4947\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947+598\mathbf{i}+\mathbf{ij}}{2}$
defines an isogeny $E_0 \to E_1$ of degree $4947 = 3 \cdot 17 \cdot 97$.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski ’21: “Orientations and the supersingular endomorphism ring problem”).

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- $\approx$All isogeny security reduces to the "$\Longrightarrow$" direction.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- $\approx$All isogeny security reduces to the "$\Longrightarrow$" direction.
- **SQIsign** builds on the "$\Longleftarrow$" direction constructively.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski ’21: “Orientations and the supersingular endomorphism ring problem”).

- $\approx$All isogeny security reduces to the “$\Longrightarrow$” direction.
- **SQIsign** builds on the “$\Longleftarrow$” direction constructively.
- Essential tool for *both* constructions and attacks.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- $\approx$All isogeny security reduces to the "$\Longrightarrow$" direction.
- **SQIsign** builds on the "$\Longleftarrow$" direction constructively.
- Essential tool for *both* constructions and attacks.

Constructively, *partially* known endomorphism rings are useful.
$\rightsquigarrow$ **Oriented curves** and **isogeny group actions**. 🏝

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd} E_0 \arrow[r, dashed, "secret"] & E_{pk} \end{tikzcd}$$

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "secret"] \arrow[d, dashed, "commitment"'] & E_{pk} \\
E_{com} &
\end{tikzcd}$$

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "secret"] \arrow[d, dashed, "commitment"'] & E_{pk} \\
E_{com} \arrow[r, "challenge"'] & E_{chl}
\end{tikzcd}$$

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "secret"] \arrow[d, dashed, "commitment"'] & E_{pk} \arrow[d, very thick, "response"] \\
E_{com} \arrow[r, "challenge"'] & E_{chl}
\end{tikzcd}$$

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, dashed, "\textit{commitment}"'] & E_{pk} \arrow[d, thick, "\textit{response}"] \\
E_{com} \arrow[r, "\textit{challenge}"'] & E_{chl}
\end{tikzcd}$$

- Easy signature: $E_{pk} \to E_0 \to E_{com} \to E_{chl}$. *Obviously broken.*

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, dashed, "\textit{commitment}"'] & E_{pk} \arrow[d, thick, "\textit{response}"] \\
E_{com} \arrow[r, "\textit{challenge}"'] & E_{chl}
\end{tikzcd}$$

- Easy signature: $E_{pk} \to E_0 \to E_{com} \to E_{chl}$. *Obviously broken.*
- **SQIsign**: Construct new path $E_{pk} \to E_{chl}$ (using *secret*).

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "secret"] \arrow[d, dashed, "commitment"'] & E_{pk} \arrow[d, thick, "response"] \\
E_{com} \arrow[r, "challenge"'] & E_{chl}
\end{tikzcd}$$

- Easy signature: $E_{pk} \to E_0 \to E_{com} \to E_{chl}$. *Obviously broken.*
- **SQIsign**: Construct new path $E_{pk} \to E_{chl}$ (using *secret*).
- It relies on an explicit form of the Deuring correspondence.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can find *smooth* isogeny $E \to E'$.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can find *smooth* isogeny $E \to E'$.

$\rightsquigarrow$ SQIsign rewrites the “broken” signature

$$E_{pk} \to E_0 \to E_{com} \to E_{chl}$$

into a random (smooth) isogeny $E_{pk} \to E_{chl}$.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can find *smooth* isogeny $E \to E'$.

$\rightsquigarrow$ SQIsign rewrites the “broken” signature

$$E_{pk} \to E_0 \to E_{com} \to E_{chl}$$

into a random (smooth) isogeny $E_{pk} \to E_{chl}$.

> *“If you have KLPT implemented very nicely as a black box, then anyone can implement SQIsign.”* — Yan Bo Ti

SQIsign: Why?

+ It’s extremely small compared to the competition.

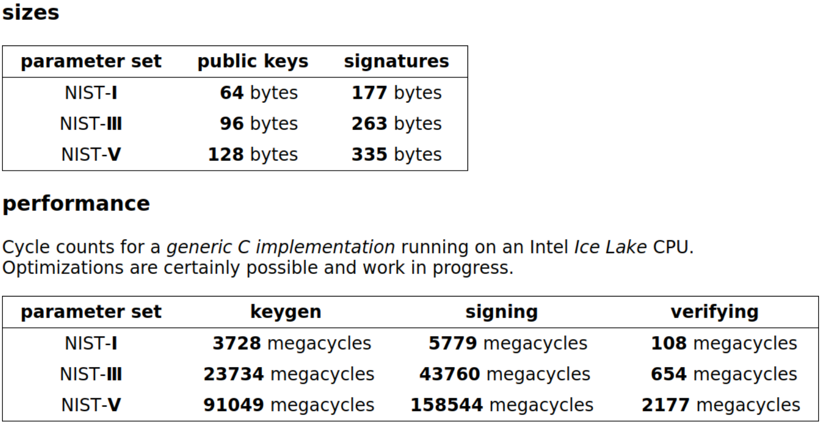
– It’s relatively slow compared to the competition.

+ ...but performance only gets better!

SQIsign: Why?

+ It’s extremely small compared to the competition.
– It’s relatively slow compared to the competition.
+ ...but performance only gets better!

SQIsign (original version): Numbers



sizes

parameter set	public keys	signatures
NIST-**I**	**64** bytes	**177** bytes
NIST-**III**	**96** bytes	**263** bytes
NIST-**V**	**128** bytes	**335** bytes

performance

Cycle counts for a *generic C implementation* running on an Intel *Ice Lake* CPU.
Optimizations are certainly possible and work in progress.

parameter set	keygen	signing	verifying
NIST-**I**	**3728** megacycles	**5779** megacycles	**108** megacycles
NIST-**III**	**23734** megacycles	**43760** megacycles	**654** megacycles
NIST-**V**	**91049** megacycles	**158544** megacycles	**2177** megacycles

Source: `https://sqisign.org` (2023–2024)

SQIsign (current version): Dramatically improved!

- The $\geq 20\times$ speedup over the original version of SQIsign comes from the new tools underlying the SIKE attacks.
- Also, it has even smaller signatures.

SQIsign (current version): Dramatically improved!

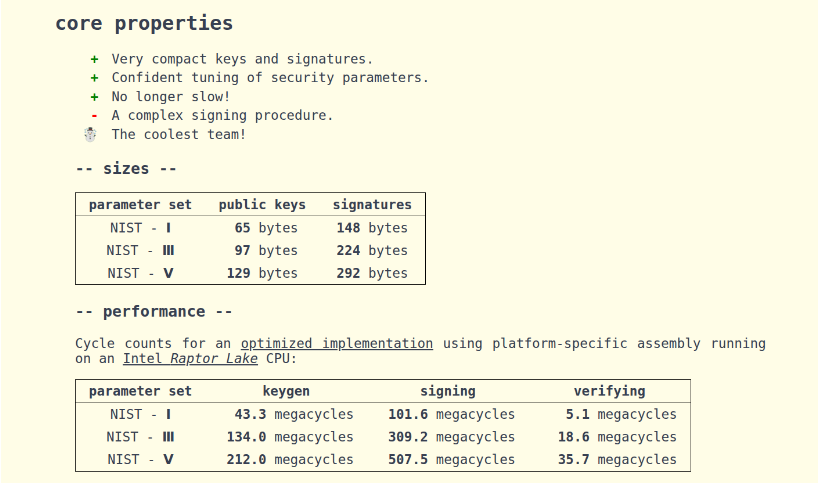
- The $\geq 20\times$ speedup over the original version of SQIsign comes from the new tools underlying the SIKE attacks.
- Also, it has even smaller signatures.

Main <u>idea</u> (from "SQIsign[H2]D" papers): Use HD representation.

$$\begin{tikzcd}
E_0 \arrow[r, Rightarrow, dashed, "secret"] \arrow[d, Rightarrow, dashed, "commitment"'] & E_{pk} \arrow[d, Rightarrow, "response"] \\
E_{com} \arrow[r, "challenge"'] & E_{chl}
\end{tikzcd}$$

$\longrightarrow$ 1-dimensional isogeny $\Longrightarrow$ 2-dimensional isogeny

SQIsign (current version): Numbers



core properties

- + Very compact keys and signatures.
- + Confident tuning of security parameters.
- + No longer slow!
- - A complex signing procedure.
- ☃ The coolest team!

-- sizes --

parameter set	public keys	signatures
NIST - **I**	**65** bytes	**148** bytes
NIST - **III**	**97** bytes	**224** bytes
NIST - **V**	**129** bytes	**292** bytes

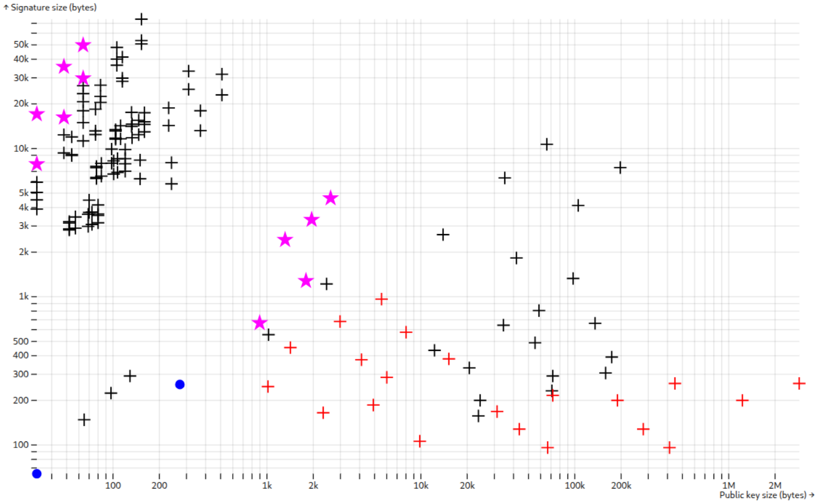
-- performance --

Cycle counts for an optimized implementation using platform-specific assembly running on an Intel *Raptor Lake* CPU:

parameter set	keygen	signing	verifying
NIST - **I**	**43.3** megacycles	**101.6** megacycles	**5.1** megacycles
NIST - **III**	**134.0** megacycles	**309.2** megacycles	**18.6** megacycles
NIST - **V**	**212.0** megacycles	**507.5** megacycles	**35.7** megacycles

Source: `https://sqisign.org` (2025–?)

SQIsign (current version): Comparison



Source: `https://pqshield.github.io/nist-sigs-zoo`

Signing with isogenies — another way

Signing with isogenies — another way

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

We don't know how to instantiate such an oracle.

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

We don't know how to instantiate such an oracle.

"One man's gap-in-security-proof is another man's treasure."

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

We don't know how to instantiate such an oracle.

"One man's gap-in-security-proof is another man's treasure."

PRISM builds a *two-round* identification scheme as follows:

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

We don't know how to instantiate such an oracle.

"One man's gap-in-security-proof is another man's treasure."

PRISM builds a *two-round* identification scheme as follows:

- **Public key:** Random supersingular elliptic curve E;
  prover knows a secret isogeny $E_0 \to E$.

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

We don't know how to instantiate such an oracle.

"One man's gap-in-security-proof is another man's treasure."

PRISM builds a *two-round* identification scheme as follows:

- **Public key:** Random supersingular elliptic curve E;
  prover knows a secret isogeny $E_0 \to E$.
- **Challenge:** A large prime q.

Signing with isogenies — another way

Issue: Original security proofs for HD variants of SQIsign require access to an oracle for producing random isogenies of bounded degrees.

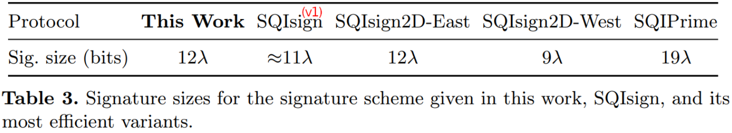
We don't know how to instantiate such an oracle.

"One man's gap-in-security-proof is another man's treasure."

PRISM builds a *two-round* identification scheme as follows:

- **Public key:** Random supersingular elliptic curve E;
  prover knows a secret isogeny $E_0 \to E$.
- **Challenge:** A large prime q.
- **Response:** An isogeny $\varphi\colon E \to \_$ of degree q.
  How? Create HD representation of φ using knowledge of $\mathrm{End}(E)$!

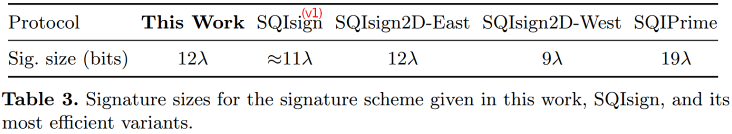
PRISM: Parameters



Protocol	**This Work**	SQIsign$^{(v1)}$	SQIsign2D-East	SQIsign2D-West	SQIPrime
Sig. size (bits)	12λ	$\approx 11\lambda$	12λ	9λ	19λ

Table 3. Signature sizes for the signature scheme given in this work, SQIsign, and its most efficient variants.

PRISM: Parameters



Protocol	**This Work**	SQIsign$^{(v1)}$	SQIsign2D-East	SQIsign2D-West	SQIPrime
Sig. size (bits)	12λ	$\approx 11\lambda$	12λ	9λ	19λ

Table 3. Signature sizes for the signature scheme given in this work, SQIsign, and its most efficient variants.

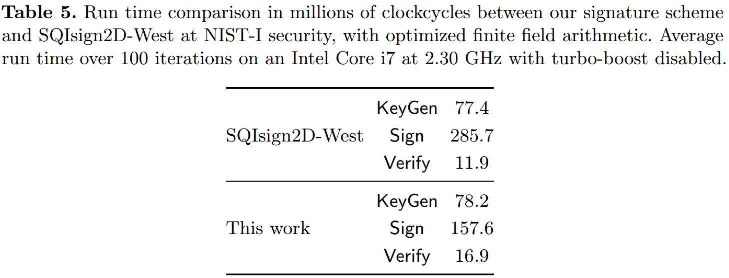


Table 5. Run time comparison in millions of clockcycles between our signature scheme and SQIsign2D-West at NIST-I security, with optimized finite field arithmetic. Average run time over 100 iterations on an Intel Core i7 at 2.30 GHz with turbo-boost disabled.

SQIsign2D-West	KeyGen	77.4
	Sign	285.7
	Verify	11.9
This work	KeyGen	78.2
	Sign	157.6
	Verify	16.9

Plan for this talk

- Elliptic curves & isogenies. ✓
- The SIKE attacks. ✓
- Transcending to higher dimensions. ✓
- Isogeny group actions. ✓
- Signatures from isogenies. ✓

Ad break

THE isogeny club

Seminar Sessions

A seminar session for young isogenists.

https://isogeny.club

Questions?

(Also feel free to email me: `lorenz@yx7.cc`)