Advances in isogeny-based cryptography

Lorenz Panny

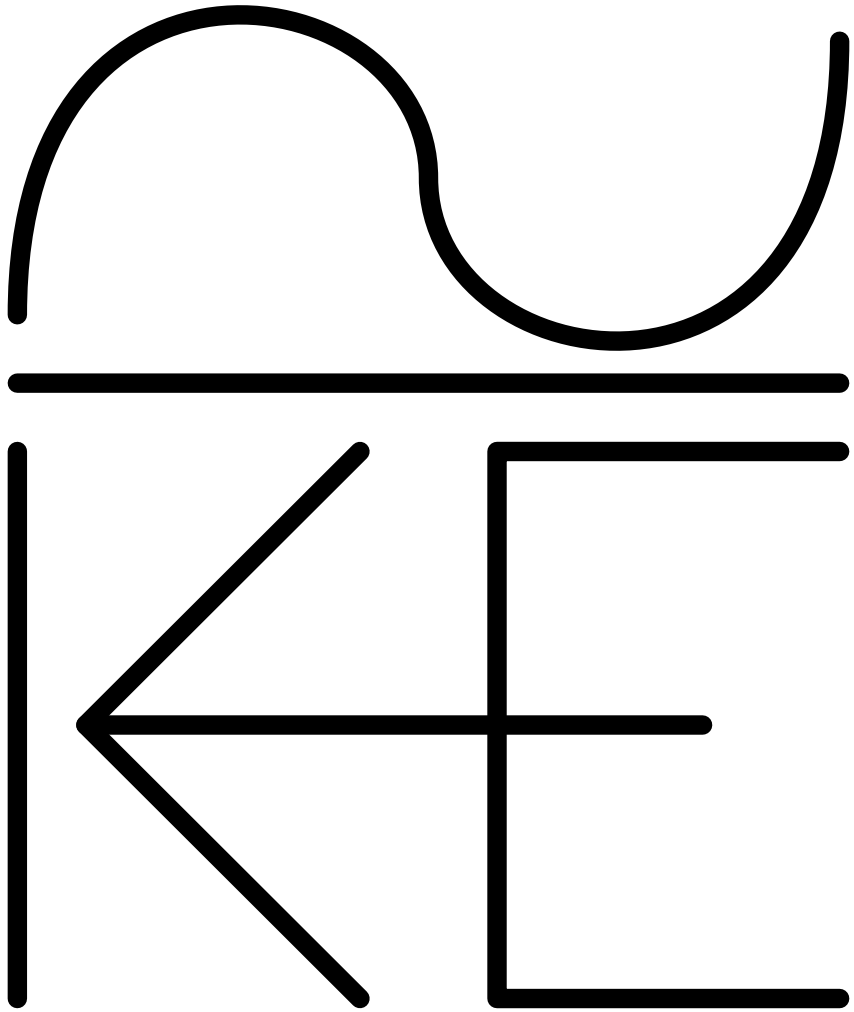
Technische Universität München

PQCSA summer school “PQC fundamentals”,
Albena, 20 June 2025

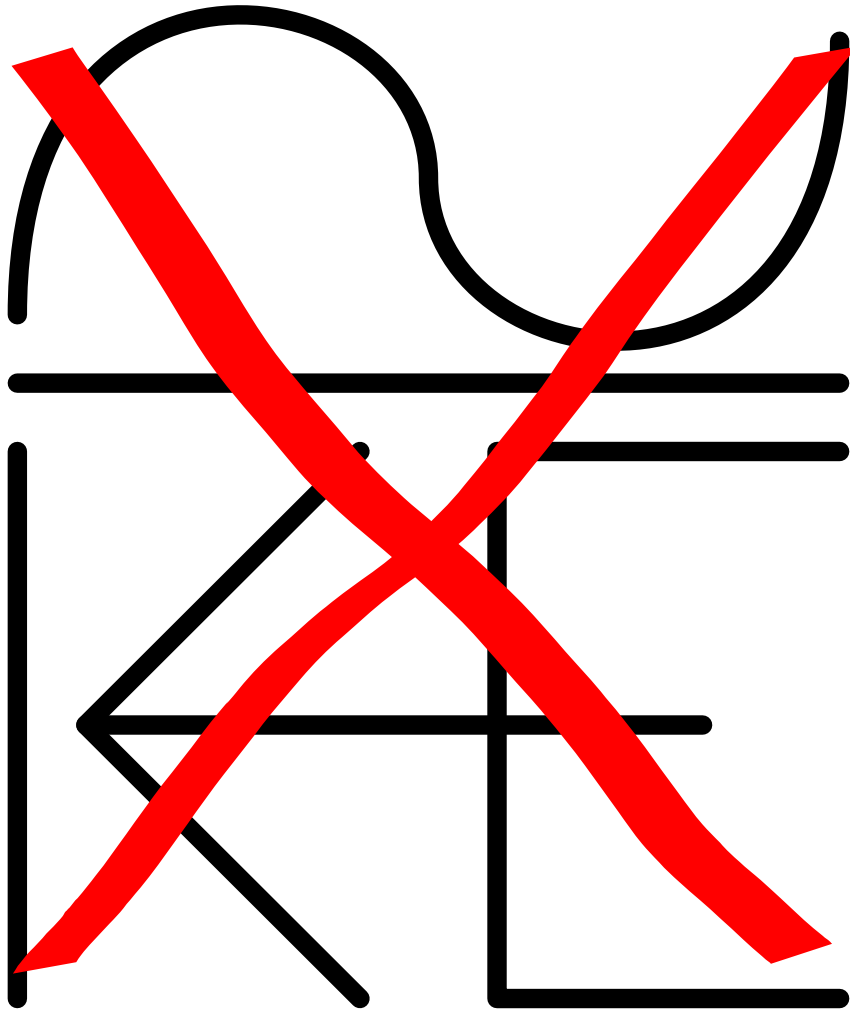
Plan for this talk

- The SIKE attacks.
- Transcending to higher dimensions.
- Isogeny group actions (+ HD).
- Signatures from isogenies (+ HD).

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.

The SIDH/SIKE attacks

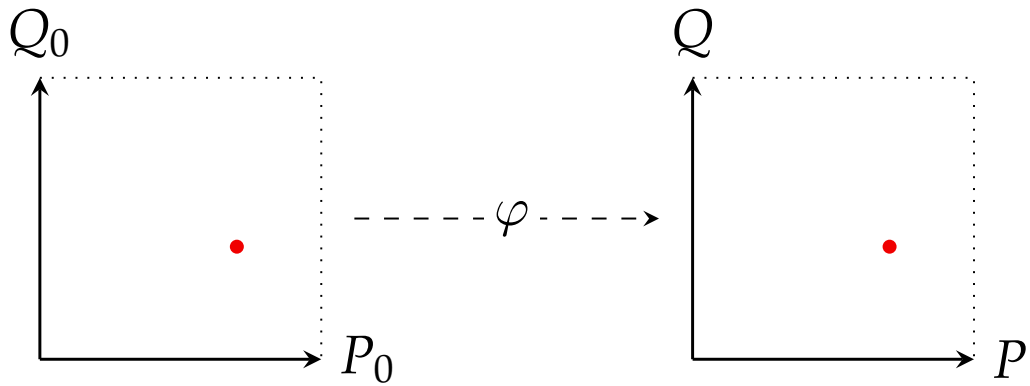
- **Not** a case of everyone overlooking something stupid.

The SIDH/SIKE attacks

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

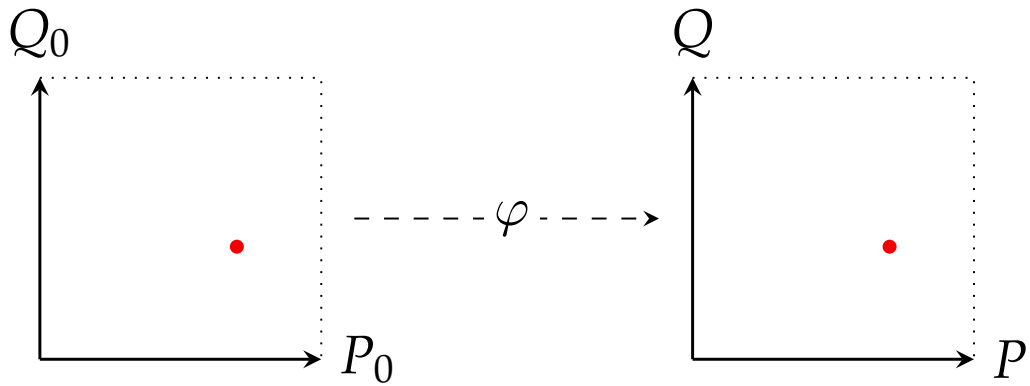
The SIDH/SIKE attacks

- **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.

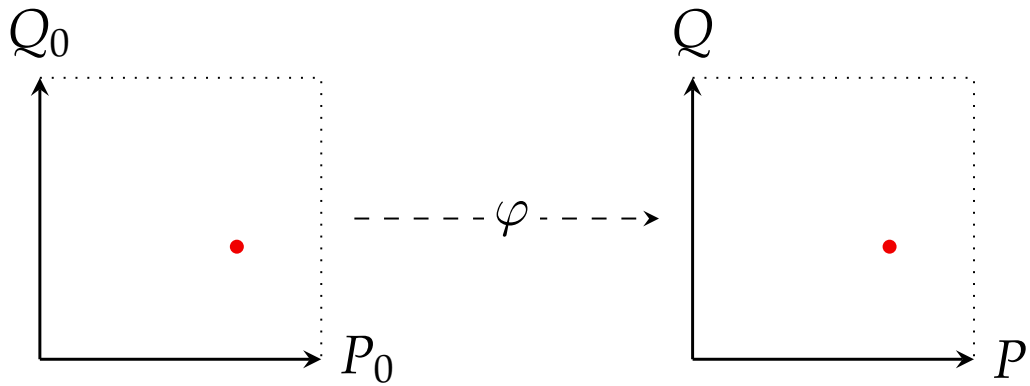


This **isogeny interpolation** problem turns out to be **easy!**

(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.



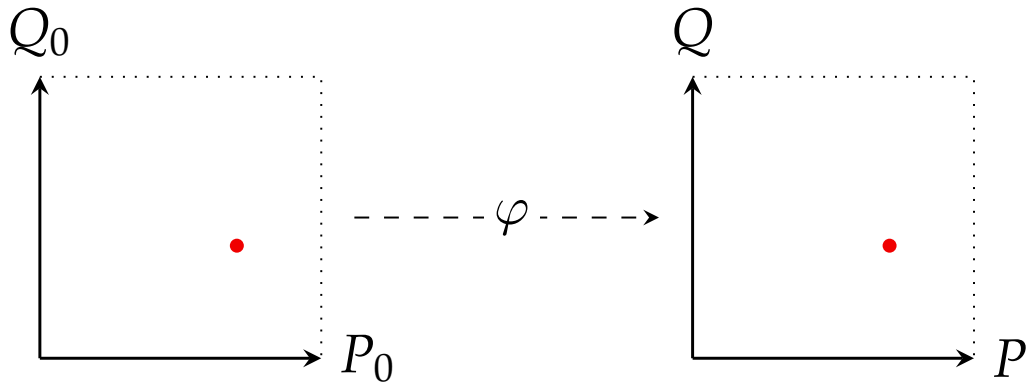
This **isogeny interpolation** problem turns out to be **easy!**

(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.



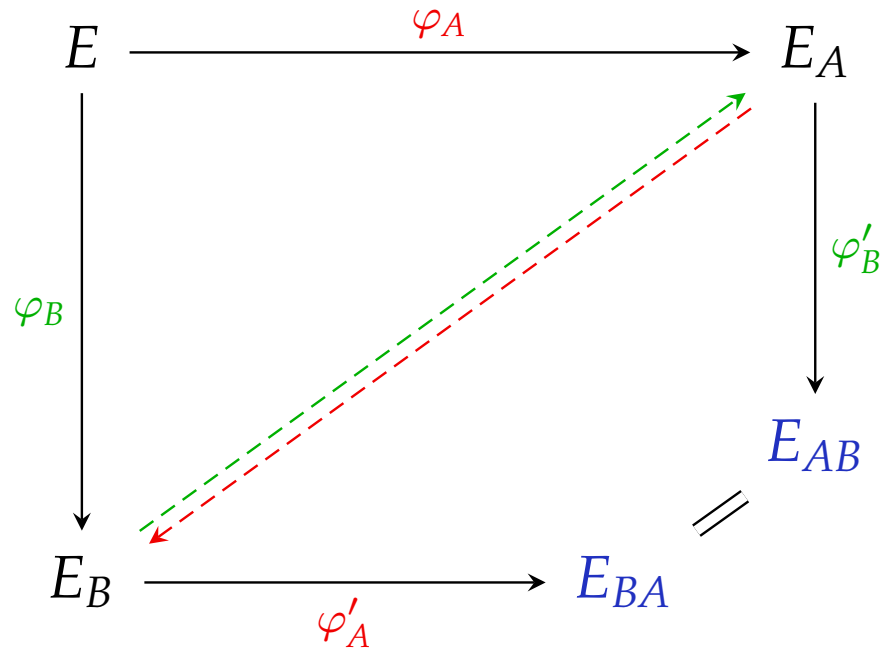
This **isogeny interpolation** problem turns out to be **easy!**

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

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

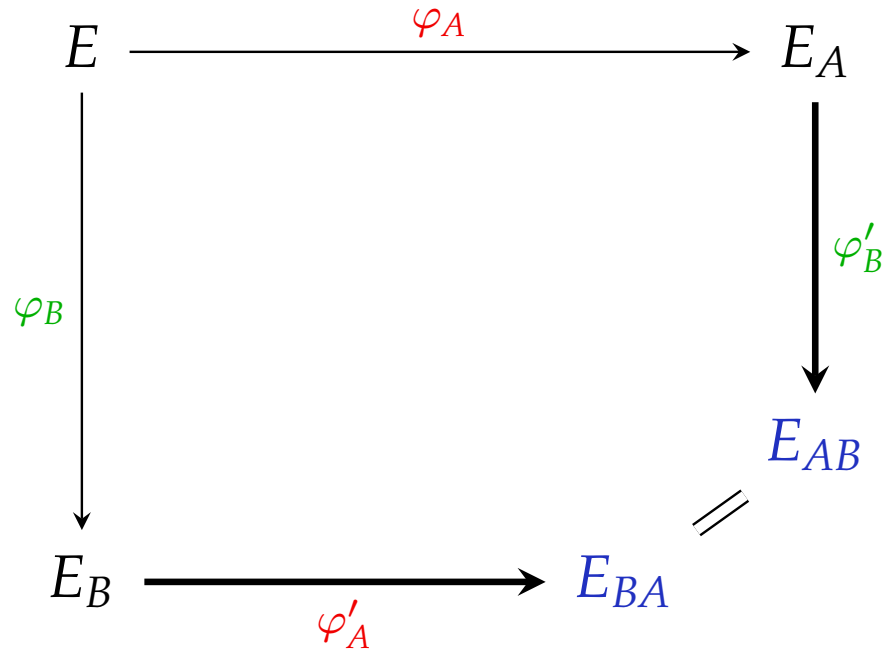
$\rightsquigarrow$ The best thing to ever happen to isogenies! 🎉

Isogeny-based key exchange: High-level view

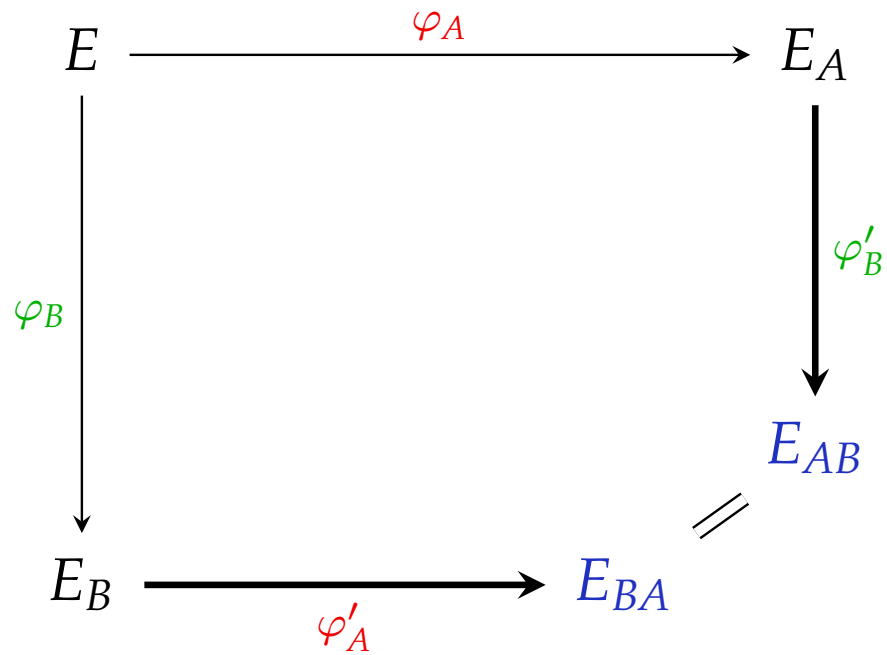


- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice somehow finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob somehow finds $\varphi_{B'} :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

How to find “parallel” isogenies?



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?

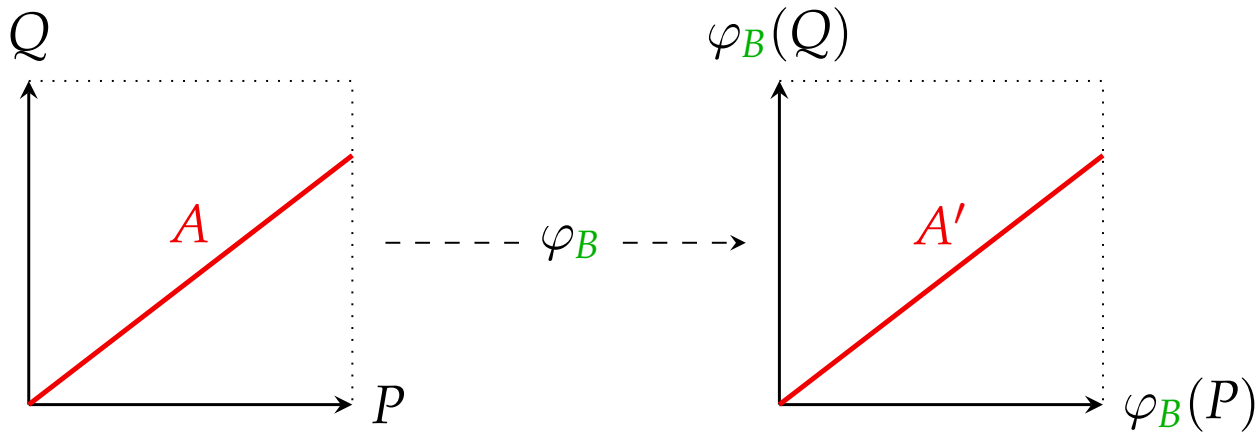
SIKE's solution:

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

How to find "parallel" isogenies?

SIKE's solution:

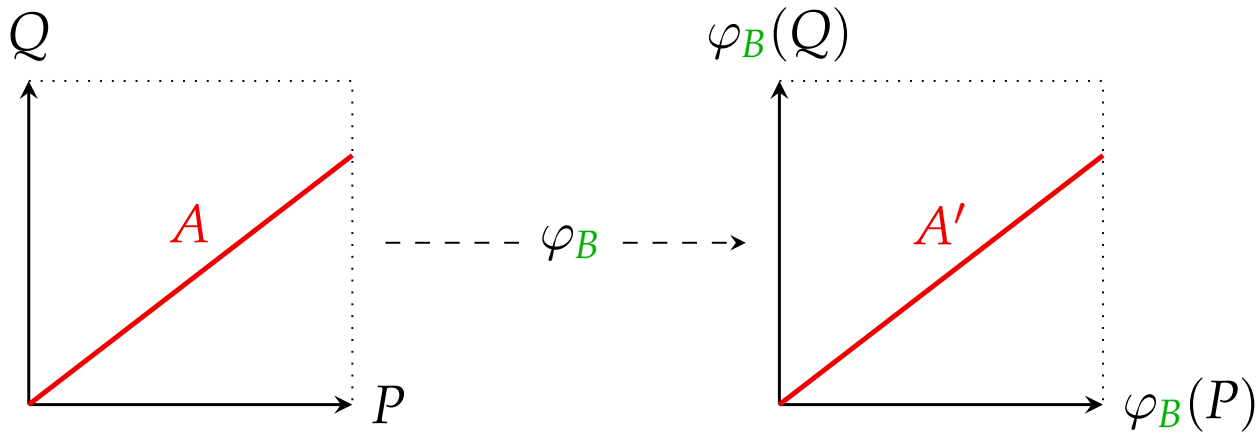
The isogeny φ_B is a group homomorphism! (and $A \cap B = \{\infty\}$)



How to find "parallel" isogenies?

SIKE's solution:

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

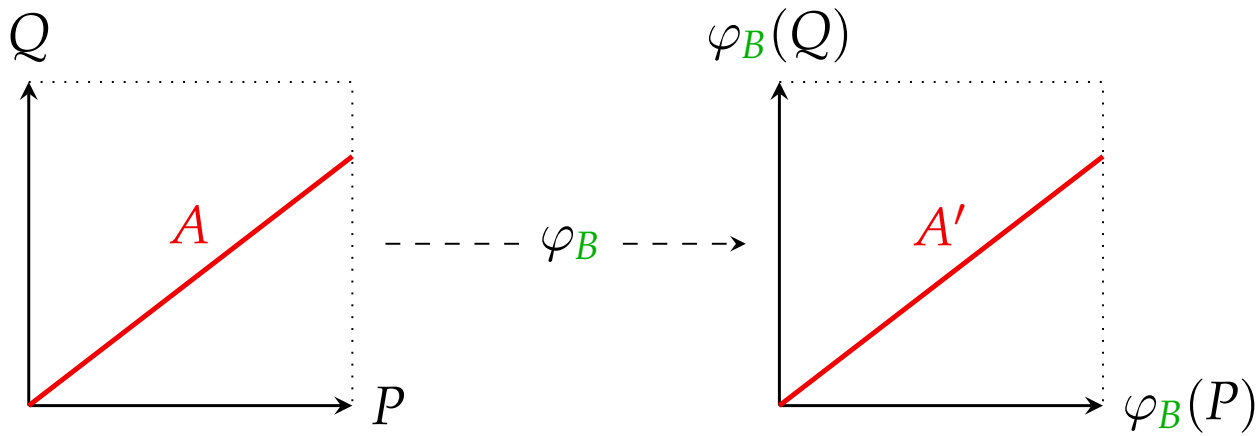


- Alice picks A as $\langle P + [a]Q \rangle$ for fixed public $P, Q \in E$.

How to find "parallel" isogenies?

SIKE's solution:

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

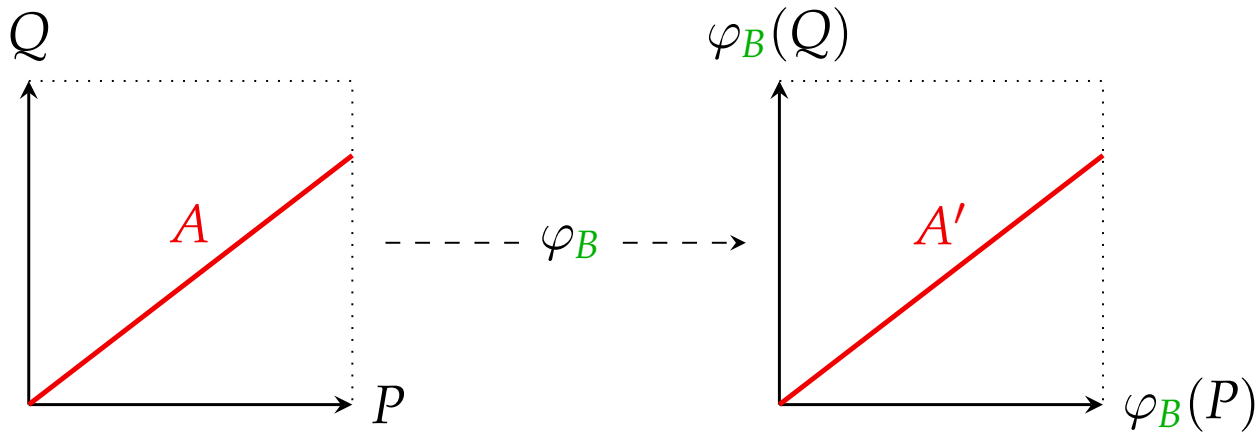


- Alice picks A as $\langle P + [a]Q \rangle$ for fixed public $P, Q \in E$.
- Bob includes $\varphi_B(P)$ and $\varphi_B(Q)$ in his public key.

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.

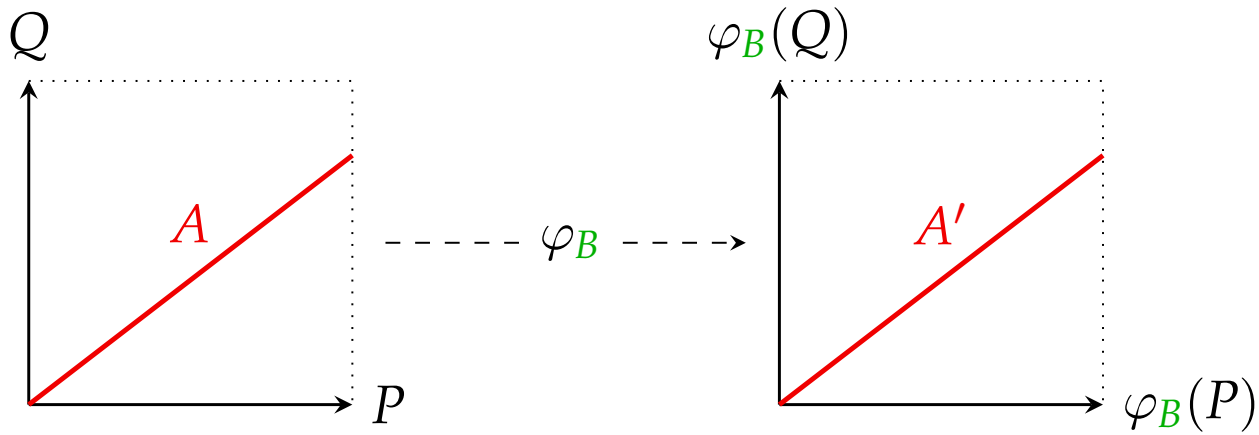
$\Longrightarrow$ Now Alice can compute $A' = \varphi_B(A)$ as $\langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$.

(Similarly for Bob.)

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 $A' = \varphi_B(A)$ as $\langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$.

(Similarly for Bob.)

⚠ This reveals the restriction of φ_B to $\langle P, Q \rangle$!

($\rightsquigarrow$ *Two-dimensional discrete-logarithm computation modulo* $\deg(\varphi_A)$*, which is smooth.*)

Higher-dimensional isogenies

Main technique underlying attack:

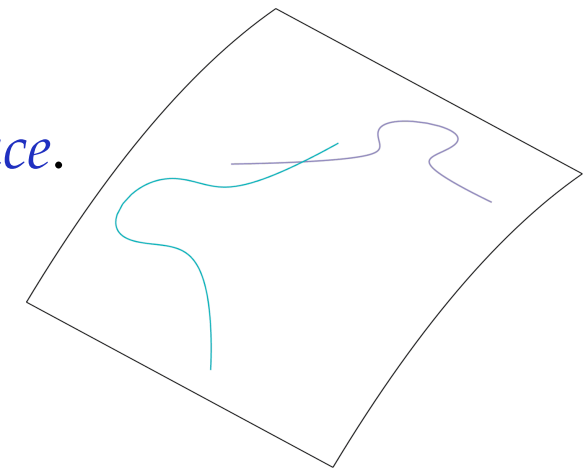
Computing isogenies between *products* of elliptic curves

Higher-dimensional isogenies

Main technique underlying attack:

Computing isogenies between *products* of elliptic curves

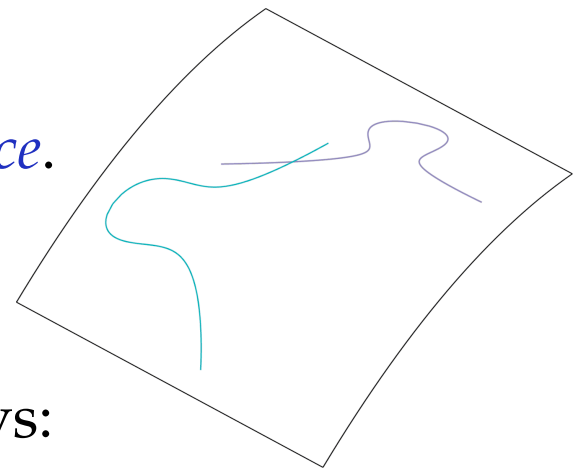
- The product $E \times E'$ is an abelian *surface*.
  Compare: A product of two lines is a plane!



Higher-dimensional isogenies

Main technique underlying attack:

Computing isogenies between *products* of elliptic curves



- The product $E \times E'$ is an abelian *surface*.
  Compare: A product of two lines is a plane!

- Similar to elliptic curves in many ways:
  - Points form an abelian group.
  - Similar group structure, but more components.
  - Can define isogenies from kernel subgroups.

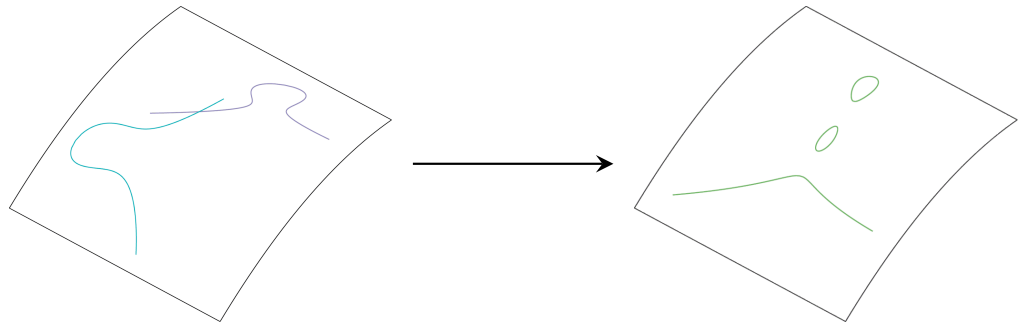
Kani's lemma

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

Kani's lemma

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

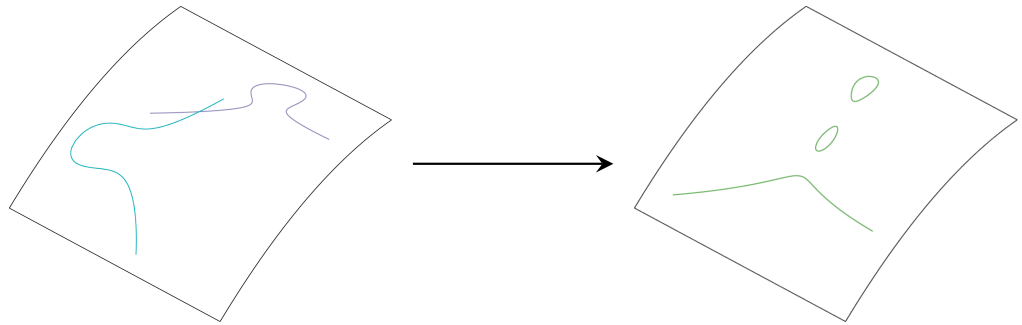
? Generic case: Codomain is a Jacobian of a genus-2 curve.



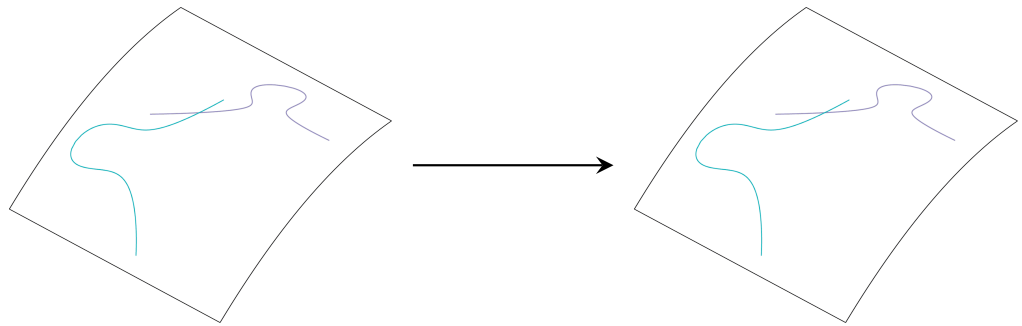
Kani's lemma

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

? Generic case: Codomain is a Jacobian of a genus-2 curve.



!! "Kani" case: Codomain is a product of two elliptic curves.



The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

Castryck–Decru (2022): Kani's criterion can be used to check whether SIDH public keys are *valid*.

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

Castryck–Decru (2022): Kani's criterion can be used to check whether SIDH public keys are *valid*.

\+ known reduction from private-key search to public-key validation

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

Castryck–Decru (2022): Kani's criterion can be used to check whether SIDH public keys are *valid*.

+ known reduction from private-key search to public-key validation
+ later generalizations & improvements (Maino–Martindale, Wesolowski, Robert, etc.)

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

Castryck–Decru (2022): Kani's criterion can be used to check whether SIDH public keys are *valid*.

+ known reduction from private-key search to public-key validation
+ later generalizations & improvements (Maino–Martindale, Wesolowski, Robert, etc.)

$\rightsquigarrow$ Unconditional polynomial-time attack.

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

Castryck–Decru (2022): Kani's criterion can be used to check whether SIDH public keys are *valid*.

+ known reduction from private-key search to public-key validation
+ later generalizations & improvements (Maino–Martindale, Wesolowski, Robert, etc.)

$\rightsquigarrow$ Unconditional polynomial-time attack.

Original Magma attack code breaks *SIKEp751* in $<$ 21 *hours* on a *single laptop core*.
Subsequent *SageMath implementation* (Pope–Oudompheng–...) takes $<$ 2 *hours*.

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

💥 **Castryck–Decru (2022):** Kani's criterion can be used to check whether SIDH public keys are *valid*.

+ known reduction from private-key search to public-key validation
+ later generalizations & improvements (Maino–Martindale, Wesolowski, Robert, etc.)

$\rightsquigarrow$ Unconditional polynomial-time attack.

Original Magma attack code breaks *SIKEp751* in $<$ 21 *hours* on a *single laptop core*.
Subsequent *SageMath implementation* (Pope–Oudompheng–...) takes $<$ 2 *hours*.

⚠ The attack crucially depends on knowing $\varphi_B(P), \varphi_B(Q)$.

The Castryck–Decru attack

Kani (1997): Under which circumstances does an isogeny $E \times E''' \to \_\_$ lead to another product $E' \times E''$?

💥 **Castryck–Decru (2022):** Kani's criterion can be used to check whether SIDH public keys are *valid*.

+ known reduction from private-key search to public-key validation
+ later generalizations & improvements (Maino–Martindale, Wesolowski, Robert, etc.)

$\rightsquigarrow$ Unconditional polynomial-time attack.

Original Magma attack code breaks *SIKEp751* in $<$ 21 *hours* on a *single laptop core*.
Subsequent *SageMath implementation* (Pope–Oudompheng–...) takes $<$ 2 *hours*.

⚠ The attack crucially depends on knowing $\varphi_B(P), \varphi_B(Q)$.

$\rightsquigarrow$ ☺ The general **isogeny problem** is entirely unaffected!

Plan for this talk

- The SIKE attacks. ✓
- Transcending to higher dimensions.
- Isogeny group actions (+ HD).
- Signatures from isogenies (+ HD).

The embedding lemma

- Fallout from the SIDH attack: New tools.

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

The embedding lemma

- Fallout from the SIDH attack: New tools.

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

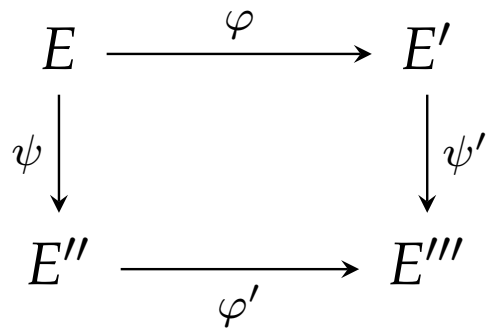
2.1. **The embedding lemma.** If α_1, α_2 are two endomorphisms of an elliptic curve E of degree a_1 and a_2, then $\alpha_1 \circ \alpha_2$ is of degree $a_1 a_2$. However it is harder to control the degree of the sum; by Cauchy-Schwartz we can bound it as: $(a_1^{1/2} - a_2^{1/2})^2 \leq \deg(\alpha_1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

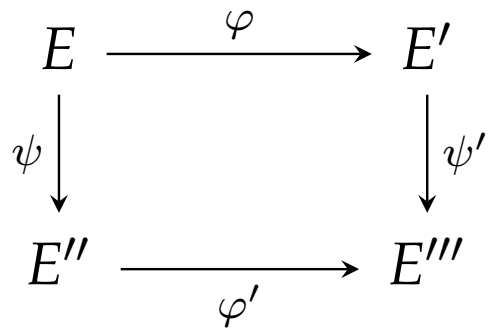
Consider a commutative diagram of isogenies



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

The embedding lemma

Consider a commutative diagram of isogenies



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

Lemma. Then

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

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

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

The HD representation

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

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

The HD representation

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

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

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

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

∴ Issue: Need to find suitable ψ. Not always easy/possible!

The HD representation

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

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

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

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

☹ Issue: Need to find suitable ψ. Not always easy/possible!

⇝ For full generality, need to embed in even higher dimension.

The *full* HD representation

Every $E \times E \times E \times E$ has an endomorphism of *any* degree.

(Proof: *Sum-of-four-squares theorem + quaternions!* ☺)

The *full* HD representation

Every $E \times E \times E \times E$ has an endomorphism of *any* degree.

(Proof: *Sum-of-four-squares theorem + quaternions!* ☺)

$\rightsquigarrow$ The endomorphism of $E^4 \times E'^4$ given by

$$\left(\begin{array}{cccc|cccc} t & u & v & w & -\widehat{\varphi} & 0 & 0 & 0 \\ -u & t & -w & v & 0 & -\widehat{\varphi} & 0 & 0 \\ -v & w & t & -u & 0 & 0 & -\widehat{\varphi} & 0 \\ -w & -v & u & t & 0 & 0 & 0 & -\widehat{\varphi} \\ \hline \varphi & 0 & 0 & 0 & t & -u & -v & -w \\ 0 & \varphi & 0 & 0 & u & t & w & -v \\ 0 & 0 & \varphi & 0 & v & -w & t & u \\ 0 & 0 & 0 & \varphi & w & v & -u & t \end{array}\right)$$

is an N-isogeny, where $N = \deg(\varphi) + t^2 + u^2 + v^2 + w^2$.

The *full* HD representation

Every $E \times E \times E \times E$ has an endomorphism of *any* degree.

(Proof: *Sum-of-four-squares theorem + quaternions!* ☺)

$\rightsquigarrow$ The endomorphism of $E^4 \times E'^4$ given by

$$\left(\begin{array}{cccc|cccc} t & u & v & w & -\widehat{\varphi} & 0 & 0 & 0 \\ -u & t & -w & v & 0 & -\widehat{\varphi} & 0 & 0 \\ -v & w & t & -u & 0 & 0 & -\widehat{\varphi} & 0 \\ -w & -v & u & t & 0 & 0 & 0 & -\widehat{\varphi} \\ \hline \varphi & 0 & 0 & 0 & t & -u & -v & -w \\ 0 & \varphi & 0 & 0 & u & t & w & -v \\ 0 & 0 & \varphi & 0 & v & -w & t & u \\ 0 & 0 & 0 & \varphi & w & v & -u & t \end{array}\right)$$

is an N-isogeny, where $N = \deg(\varphi) + t^2 + u^2 + v^2 + w^2$.

☺ It can be explicitly computed from knowledge of $\varphi|_{E[N]}$.

The *full* HD representation

Every $E \times E \times E \times E$ has an endomorphism of *any* degree.

(Proof: *Sum-of-four-squares theorem + quaternions!* ☺)

$\rightsquigarrow$ The endomorphism of $E^4 \times E'^4$ given by

$$\left(\begin{array}{cccc|cccc} t & u & v & w & -\widehat{\varphi} & 0 & 0 & 0 \\ -u & t & -w & v & 0 & -\widehat{\varphi} & 0 & 0 \\ -v & w & t & -u & 0 & 0 & -\widehat{\varphi} & 0 \\ -w & -v & u & t & 0 & 0 & 0 & -\widehat{\varphi} \\ \hline \varphi & 0 & 0 & 0 & t & -u & -v & -w \\ 0 & \varphi & 0 & 0 & u & t & w & -v \\ 0 & 0 & \varphi & 0 & v & -w & t & u \\ 0 & 0 & 0 & \varphi & w & v & -u & t \end{array}\right)$$

is an N-isogeny, where $N = \deg(\varphi) + t^2 + u^2 + v^2 + w^2$.

☺ It can be explicitly computed from knowledge of $\varphi|_{E[N]}$.

⚠ Requires isogeny formulas for principally polarized abelian varieties of dimension > 2. Highly non-trivial matter, but doable and efficient once $\exists$.

Plan for this talk

- The SIKE attacks. ✓
- Transcending to higher dimensions. ✓
- Isogeny group actions (+ HD).
- Signatures from isogenies (+ HD).

Recap: CSIDH

Recall *(⇜ Tuesday)*:
In CSIDH, we can (only) act efficiently by "left" and "right" ℓ-isogeny steps for small ℓ.

Recap: CSIDH

Recall *(⇜ Tuesday)*:
In CSIDH, we can (only) act efficiently by "left" and "right" ℓ-isogeny steps for small ℓ.

This is good enough for DH-style key exchange, but to get an unrestricted effective group action, we need more.

Recap: CSIDH

Recall *(⇜ Tuesday)*:
In CSIDH, we can (only) act efficiently by "left" and "right" ℓ-isogeny steps for small ℓ.

This is good enough for DH-style key exchange, but to get an unrestricted effective group action, we need more.

Recall *(⇜ Tuesday)*:
Isogeny paths leading to the same curve are characterized by the *class group* $\mathrm{cl}(\mathcal{O})$ where $\mathcal{O} = \mathbb{Z}[\pi] \cong \mathbb{Z}[\sqrt{-p}]$.

The basic problem with the basic strategy

Issue:

The basic problem with the basic strategy

Issue:

- Representing $\mathrm{cl}(\mathcal{O})$ by the group $(\mathbb{Z}^n, +)$ of exponents makes the exponents grow larger with each operation.
  - $\rightsquigarrow$ Cost of evaluating after k operations is $O(\exp(k))$.

The basic problem with the basic strategy

Issue:

- Representing $\mathrm{cl}(\mathcal{O})$ by the group $(\mathbb{Z}^n, +)$ of exponents makes the exponents grow larger with each operation.
  - $\rightsquigarrow$ Cost of evaluating after k operations is $O(\exp(k))$.
- Representing $\mathrm{cl}(\mathcal{O})$ as reduced ideals allows computing in $\mathrm{cl}(\mathcal{O})$ efficiently, but evaluation becomes superpolynomial.

The basic problem with the basic strategy

Issue:

- Representing $\mathrm{cl}(\mathcal{O})$ by the group $(\mathbb{Z}^n, +)$ of exponents makes the exponents grow larger with each operation.
  - $\rightsquigarrow$ Cost of evaluating after k operations is $O(\exp(k))$.
- Representing $\mathrm{cl}(\mathcal{O})$ as reduced ideals allows computing in $\mathrm{cl}(\mathcal{O})$ efficiently, but evaluation becomes superpolynomial.

$\rightsquigarrow$ A priori **not** an *effective* group action when done either way!

The CSI-FiSh approach

...combines exponent vectors with reduction by exploiting the relation lattice of the chosen ideal classes. It works as follows:

The strategy to act by a given, arbitrarily long and ugly exponent vector $\underline{v} \in \mathbb{Z}^d$ consists of the following steps:

1. "Computing the class group": Find a basis of the *relation lattice* $\Lambda \subseteq \mathbb{Z}^d$ with respect to $\mathfrak{l}_1, \ldots, \mathfrak{l}_d$.
   [Classically subexponential-time, quantumly polynomial-time. Precomputation.]
2. "Lattice reduction": Prepare a "good" basis of Λ using a lattice-reduction algorithm such as BKZ.
   [Configurable complexity-quality tradeoff by varying the block size. Precomputation.]
3. "Approximate CVP": Obtain a vector $\underline{w} \in \Lambda$ such that $\|\underline{v} - \underline{w}\|_1$ is "small", using the reduced basis.
   [Polynomial-time, but the quality depends on the quality of step 2.]
4. "Isogeny steps": Evaluate the action of the vector $\underline{v} - \underline{w} \in \mathbb{Z}^d$ as a sequence of $\mathfrak{l}_i$-steps.
   [Complexity depends entirely on the output quality of step 3.]

https://yx7.cc/blah/2023-04-14.html

The CSI-FiSh approach

...combines exponent vectors with reduction by exploiting the relation lattice of the chosen ideal classes. It works as follows:

The strategy to act by a given, arbitrarily long and ugly exponent vector $\underline{v} \in \mathbb{Z}^d$ consists of the following steps:

1. "Computing the class group": Find a basis of the *relation lattice* $\Lambda \subseteq \mathbb{Z}^d$ with respect to $\mathfrak{l}_1, \ldots, \mathfrak{l}_d$.
   [Classically subexponential-time, quantumly polynomial-time. Precomputation.]
2. "Lattice reduction": Prepare a "good" basis of Λ using a lattice-reduction algorithm such as BKZ.
   [Configurable complexity-quality tradeoff by varying the block size. Precomputation.]
3. "Approximate CVP": Obtain a vector $\underline{w} \in \Lambda$ such that $\|\underline{v} - \underline{w}\|_1$ is "small", using the reduced basis.
   [Polynomial-time, but the quality depends on the quality of step 2.]
4. "Isogeny steps": Evaluate the action of the vector $\underline{v} - \underline{w} \in \mathbb{Z}^d$ as a sequence of $\mathfrak{l}_i$-steps.
   [Complexity depends entirely on the output quality of step 3.]

https://yx7.cc/blah/2023-04-14.html

The CSI-FiSh paper (2019) does all this in practice for 512-bit p.

The CSI-FiSh approach

...combines exponent vectors with reduction by exploiting the relation lattice of the chosen ideal classes. It works as follows:

The strategy to act by a given, arbitrarily long and ugly exponent vector $\underline{v} \in \mathbb{Z}^d$ consists of the following steps:

1. "Computing the class group": Find a basis of the *relation lattice* $\Lambda \subseteq \mathbb{Z}^d$ with respect to $\mathfrak{l}_1, \ldots, \mathfrak{l}_d$.
   [Classically subexponential-time, quantumly polynomial-time. Precomputation.]
2. "Lattice reduction": Prepare a "good" basis of Λ using a lattice-reduction algorithm such as BKZ.
   [Configurable complexity-quality tradeoff by varying the block size. Precomputation.]
3. "Approximate CVP": Obtain a vector $\underline{w} \in \Lambda$ such that $\|\underline{v} - \underline{w}\|_1$ is "small", using the reduced basis.
   [Polynomial-time, but the quality depends on the quality of step 2.]
4. "Isogeny steps": Evaluate the action of the vector $\underline{v} - \underline{w} \in \mathbb{Z}^d$ as a sequence of $\mathfrak{l}_i$-steps.
   [Complexity depends entirely on the output quality of step 3.]

https://yx7.cc/blah/2023-04-14.html

The CSI-FiSh paper (2019) does all this in practice for 512-bit p.

What about asymptotics?

Tradeoff: Lattice part vs. isogeny part

- By increasing the number n of ideals $\mathfrak{l}_i$, we can trade off some "isogeny effort" for "lattice effort".

$\rightsquigarrow$ Sweet spot: Minimize total cost.

Tradeoff: Lattice part vs. isogeny part

- By increasing the number n of ideals $\mathfrak{l}_i$, we can trade off some "isogeny effort" for "lattice effort".

$\rightsquigarrow$ Sweet spot: Minimize total cost.

CSI-FiSh really isn't polynomial-time

It is fairly well-known that CSIDH[1] in its basic form is merely a *restricted* effective group action $G \times X \to X$: There is a small number of group elements $\mathfrak{l}_1, \ldots, \mathfrak{l}_d \in G$ whose action can be applied to arbitrary elements of X efficiently, but applying other elements (say, large products $\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_d^{e_d}$ of the $\mathfrak{l}_i$) quickly becomes infeasible as the exponents grow.

The only known method to circumvent this issue consists of a folklore strategy first employed in practice by the signature scheme CSI-FiSh. The core of the technique is to rewrite any given group element as a *short* product combination of the $\mathfrak{l}_i$, whose action can then be computed in the usual way much more affordably. (Notice how this is philosophically similar to the role of the square-and-multiply algorithm in discrete-logarithm land!)

The main point of this post is to remark that this approach is **not asymptotically efficient**, even when a quantum computer can be used, contradicting a false belief that appears to be rather common among isogeny aficionados.

$\rightsquigarrow$

- Classically: Evaluation $L_p[1/2]$. Attack $L_p[1]$.
- Quantumly: Evaluation $L_p[1/3]$. Attack $L_p[1/2]$.

https://yx7.cc/blah/2023-04-14.html

Alternative approach: Clapoti (Page–Robert 2023)

Idea:

- Find two ideals $\mathfrak{b}$, $\mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.
  (That is, solve $f(x_1, y_1) + f(x_2, y_2) = N$ over $\mathbb{Z}$ where f is a binary quadratic form.)

Alternative approach: Clapoti (Page–Robert 2023)

Idea:

- Find two ideals $\mathfrak{b}$, $\mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$. Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.

  (That is, solve $f(x_1, y_1) + f(x_2, y_2) = N$ over $\mathbb{Z}$ where f is a binary quadratic form.)

$$\begin{array}{ccc} E & \xrightarrow{\phi_{\mathfrak{b}}} & E_{\mathfrak{a}} \\ {\scriptstyle \phi_{\overline{\mathfrak{c}}}}\downarrow & & \downarrow{\scriptstyle \psi_{\overline{\mathfrak{c}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\psi_{\mathfrak{b}}]{} & E \end{array}$$

Alternative approach: Clapoti (Page–Robert 2023)

Idea:

- Find two ideals $\mathfrak{b}$, $\mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.
  (That is, solve $f(x_1, y_1) + f(x_2, y_2) = N$ over $\mathbb{Z}$ where f is a binary quadratic form.)

$$\begin{array}{ccc} E & \xrightarrow{\phi_{\mathfrak{b}}} & E_{\mathfrak{a}} \\ {\scriptstyle \phi_{\overline{\mathfrak{c}}}}\downarrow & & \downarrow{\scriptstyle \psi_{\overline{\mathfrak{c}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\psi_{\mathfrak{b}}]{} & E \end{array}$$

- Kani: This gives an N-isogeny

$$\begin{aligned} \Phi \colon E \times E &\longrightarrow E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}, \\ (P, Q) &\longmapsto (\phi_{\mathfrak{b}}(P) + \widehat{\psi}_{\overline{\mathfrak{c}}}(Q),\ -\phi_{\overline{\mathfrak{c}}}(P) + \widehat{\psi}_{\mathfrak{b}}(Q)) \,. \end{aligned}$$

Alternative approach: Clapoti (Page–Robert 2023)

Idea:

- Find two ideals $\mathfrak{b}$, $\mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.
  (That is, solve $f(x_1, y_1) + f(x_2, y_2) = N$ over $\mathbb{Z}$ where f is a binary quadratic form.)

$$\begin{array}{ccc} E & \xrightarrow{\phi_{\mathfrak{b}}} & E_{\mathfrak{a}} \\ {\scriptstyle \phi_{\overline{\mathfrak{c}}}}\downarrow & & \downarrow{\scriptstyle \psi_{\overline{\mathfrak{c}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\psi_{\mathfrak{b}}]{} & E \end{array}$$

- Kani: This gives an N-isogeny

$$\begin{aligned} \Phi \colon E \times E &\longrightarrow E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}, \\ (P, Q) &\longmapsto (\phi_{\mathfrak{b}}(P) + \widehat{\psi}_{\overline{\mathfrak{c}}}(Q),\ -\phi_{\overline{\mathfrak{c}}}(P) + \widehat{\psi}_{\mathfrak{b}}(Q)) \,. \end{aligned}$$

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi}_{\mathfrak{b}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.

Alternative approach: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi}_{\mathfrak{b}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.

Alternative approach: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

Alternative approach: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{ \left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R) \right) \mid R \in E[N] \right\}$$

where $\gamma \in \mathrm{End}(E)$ is a generators of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

Alternative approach: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

- The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{ \left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R) \right) \mid R \in E[N] \right\}$$

  where $\gamma \in \mathrm{End}(E)$ is a generators of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

+ Doing all this in dimension 8 instead of 2, as before.

Alternative approach: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

🪄 The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{ \left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R) \right) \mid R \in E[N] \right\}$$

where $\gamma \in \mathrm{End}(E)$ is a generators of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

\+ Doing all this in dimension 8 instead of 2, as before.

$\Longrightarrow$ The isogeny group action can now be computed in polynomial time even for "ugly" input ideals.

Alternative approach: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

🪄 The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{ \left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R) \right) \mid R \in E[N] \right\}$$

where $\gamma \in \mathrm{End}(E)$ is a generators of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

\+ Doing all this in dimension 8 instead of 2, as before.

$\Longrightarrow$ The isogeny group action can now be computed in polynomial time even for "ugly" input ideals.

$\Longrightarrow$ Isogenies yield true effective group actions, at last!

Efficient in theory *and* practice: PEGASIS

PEGASIS: Practical Effective Class Group Action using 4-Dimensional Isogenies

Pierrick Dartois[1,2], Jonathan Komada Eriksen[5], Tako Boris Fouotsa[3], Arthur Herlédan Le Merdy[4], Riccardo Invernizzi[5], Damien Robert[1,2], Ryan Rueger[6,7], Frederik Vercauteren[5] and Benjamin Wesolowski[4]

Polynomial-time group action: PEGASIS

PEGASIS applies Clapoti in dimension 4 to essentially the CSIDH construction (but with $p = f \cdot 2^e - 1$ where f is small).

Polynomial-time group action: PEGASIS

PEGASIS applies Clapoti in dimension 4 to essentially the CSIDH construction (but with $p = f \cdot 2^e - 1$ where f is small).

The results of our SageMath 10.5 implementation can be found in Table 2; timings for each steps are in seconds, and are obtained by averaging 100 runs on an Intel Core i5-1235U clocked at 4.0 GHz.

Parameter set	Step 1	Step 2	Step 3	Tot. Time
500	0.097 s	0.48 s	0.96 s	1.53 s
1000	0.21 s	1.16 s	2.84 s	4.21 s
1500	1.19 s	2.85 s	6.49 s	10.5 s
2000	1.68 s	8.34 s	11.3 s	21.3 s
4000	15.6 s	52.8 s	53.5 s	122 s

Table 2. SageMath 10.5 timings on Intel Core i5-1235U at 4.0 GHz, where s denotes the number of seconds in wall-clock time. Step 1 is the time used to solve the norm equation, Step 2 is the time used to derive the kernel of the dimension 4 isogeny, and Step 3 is the time used to compute the dimension 4 isogeny.

Plan for this talk

- The SIKE attacks. ✓
- Transcending to higher dimensions. ✓
- Isogeny group actions (+ HD). ✓
- Signatures from isogenies (+ HD).

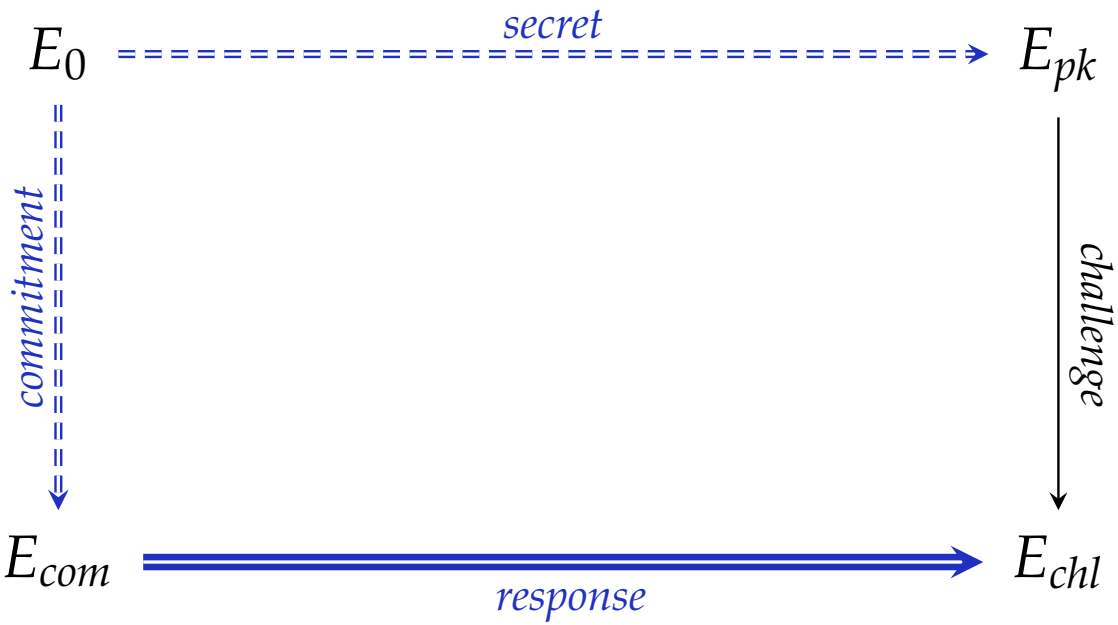
SQIsign (current version): Dramatically improved!

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

SQIsign (current version): Dramatically improved!

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

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



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

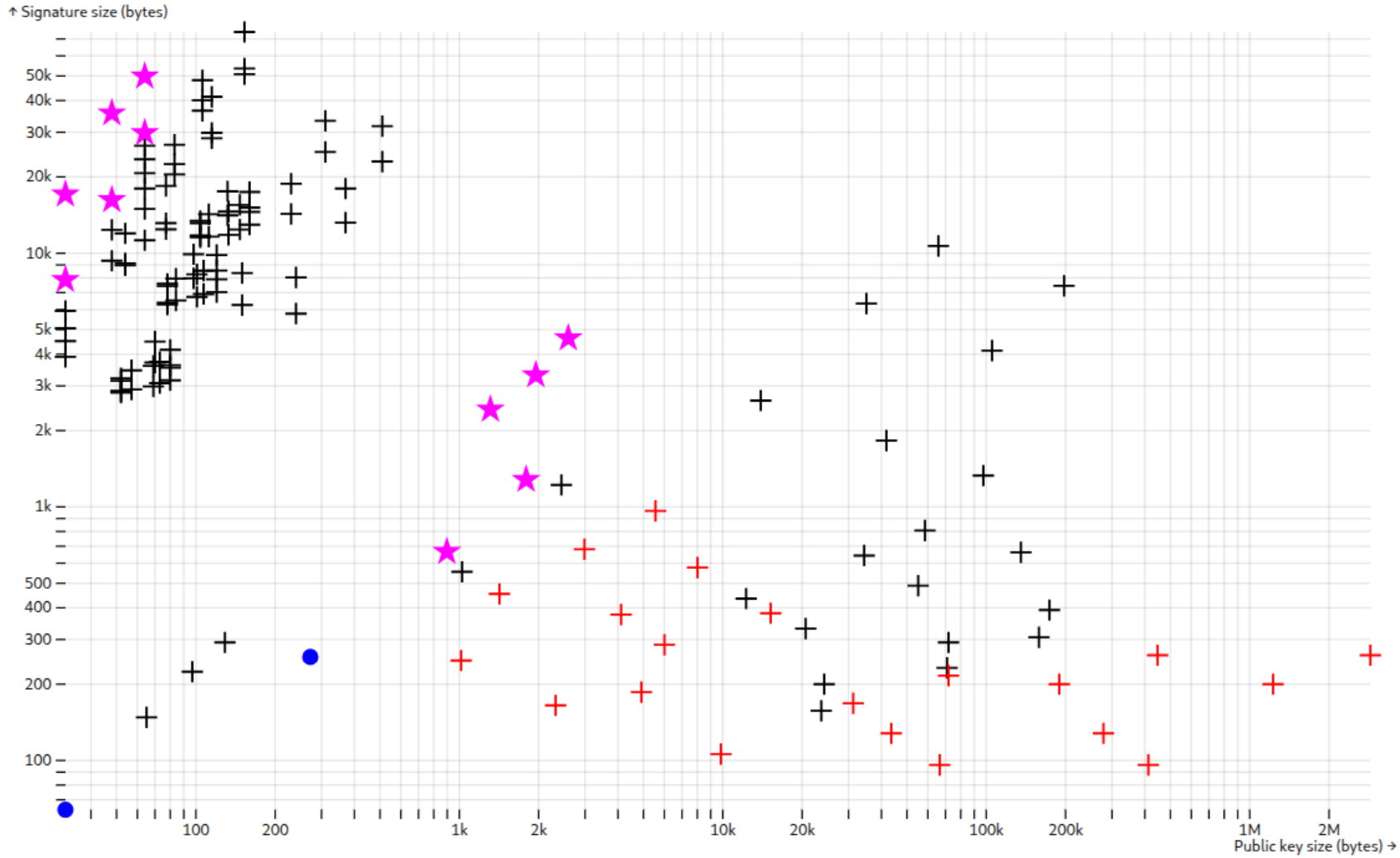
-- performance --

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`

SQIsign (current version): Security

Just as before, we require two main properties:

SQIsign (current version): Security

Just as before, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

SQIsign (current version): Security

Just as before, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In (new) SQIsign: $\approx$ Same reasoning as before.

SQIsign (current version): Security

Just as before, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In (new) SQIsign: $\approx$ Same reasoning as before.

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

SQIsign (current version): Security

Just as before, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In (new) SQIsign: $\approx$ Same reasoning as before.

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

  In (new) SQIsign: No more KLPT-dependent heuristics.

SQIsign (current version): Security

Just as before, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In (new) SQIsign: $\approx$ Same reasoning as before.

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

  In (new) SQIsign: No more KLPT-dependent heuristics.

  Only remaining issue: Simulator needs to produce HD representations of (certain) random isogenies.

SQIsign (current version): Security

Just as before, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In (new) SQIsign: $\approx$ Same reasoning as before.

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

  In (new) SQIsign: No more KLPT-dependent heuristics.

  Only remaining issue: Simulator needs to produce HD representations of (certain) random isogenies.

  This seems difficult... ☹

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

- The SIKE attacks. ✓
- Transcending to higher dimensions. ✓
- Isogeny group actions (+ HD). ✓
- Signatures from isogenies (+ HD). ✓

Ad break

`https://cryptohack.org`

(There is an isogeny category!!)

Questions?

lorenz@yx7.cc