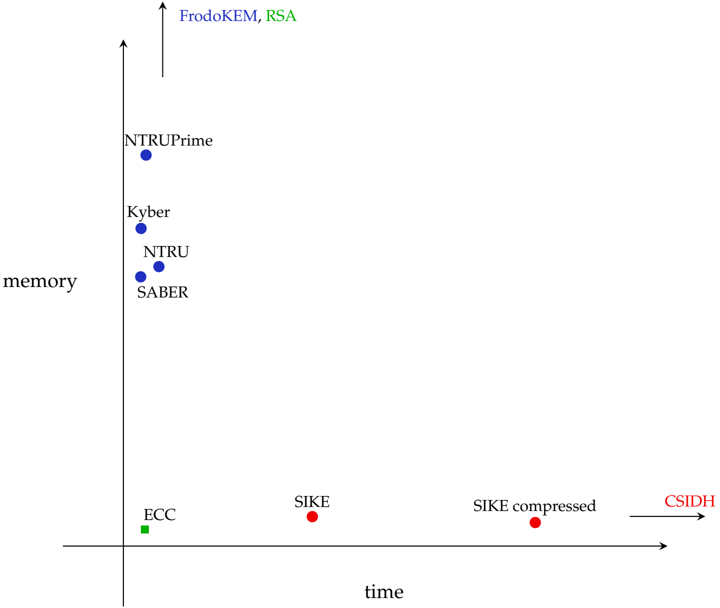
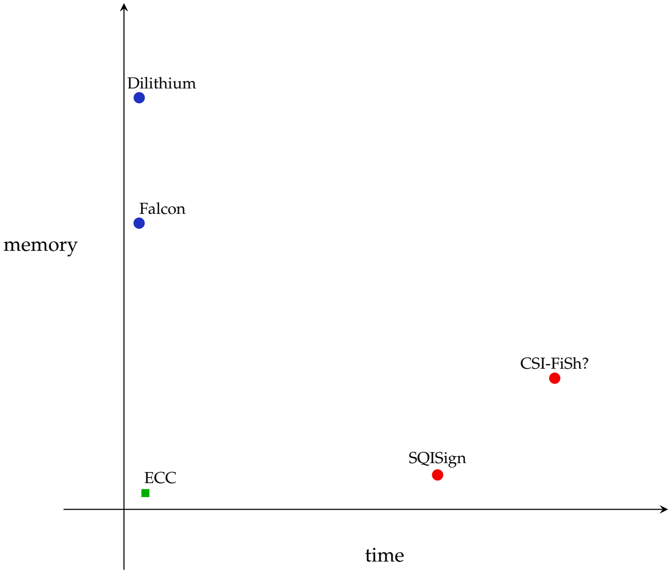
Isogeny-based cryptography: why, how, and what next?

29th July 2022

Zoo of lattice- and isogeny-based KEMs



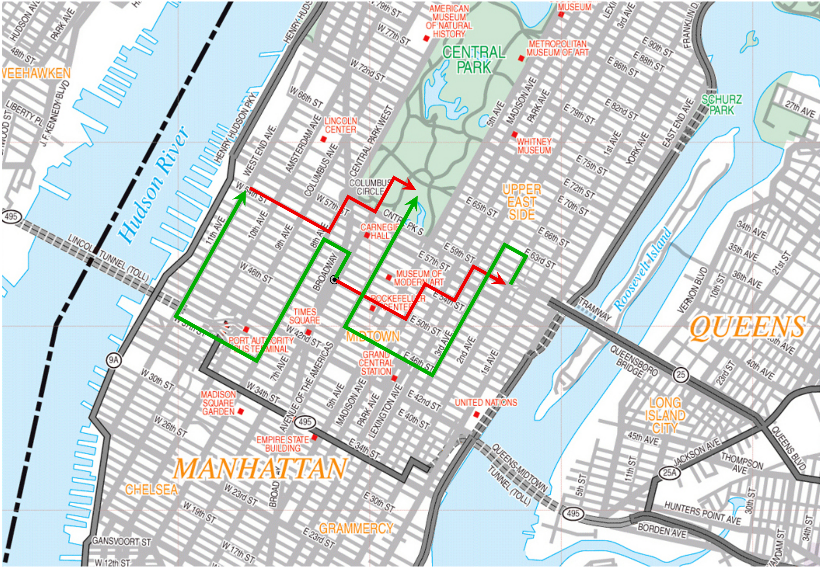
Zoo of lattice- and isogeny-based signatures



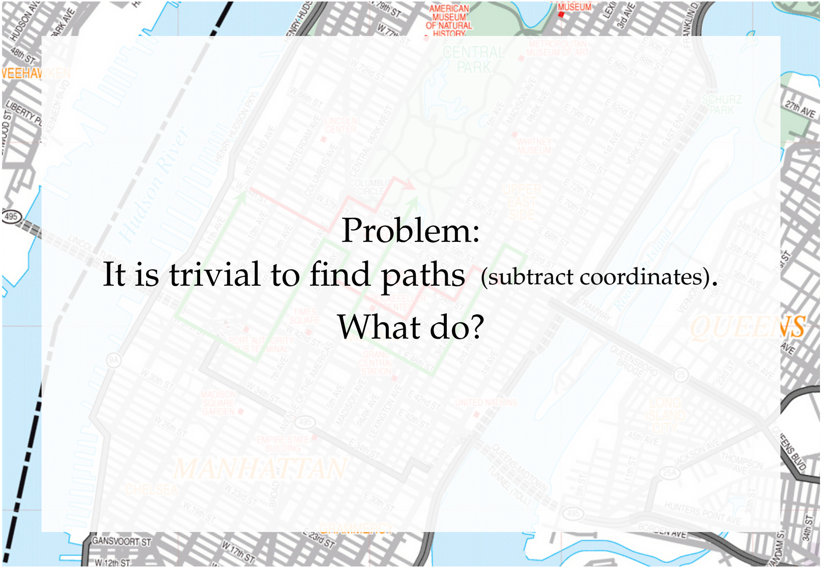
Applications (non-exhaustive list)

	Lattices	Isogenies
KEM	✓	✓
Signatures	✓	✓
NIKE	(×)	✓
FHE	✓	×
IBE	✓	×
Threshold	✓	✓
OPRF	✓	✓
VDF	(×)	(✓)
VRF	(✓)	(✓)

Graph walking Diffie–Hellman?



Graph walking Diffie–Hellman?



Problem:

It is trivial to find paths (subtract coordinates).

What do?

Big picture

- Isogenies are a source of exponentially-sized graphs.

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.
- No known efficient algorithms to recover paths from endpoints.

Big picture 🔍

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.
- No known efficient algorithms to recover paths from endpoints.
- Enough structure to navigate the graph meaningfully.
  That is: some *well-behaved* ‘directions’ to describe paths. More later.

Big picture 🔍

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.
- No known efficient algorithms to recover paths from endpoints.
- Enough structure to navigate the graph meaningfully.
  That is: some *well-behaved* 'directions' to describe paths. More later.

It is easy to construct graphs that satisfy *almost* all of these — not enough for crypto!

Stand back!

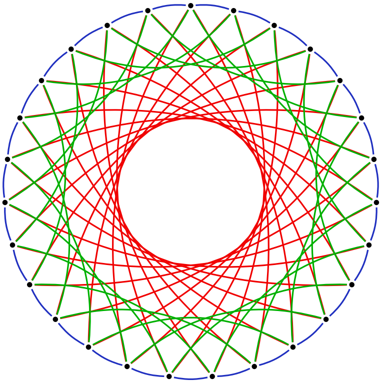
We're going to do maths.

The beauty and the beast

Components of the isogeny graphs look like this:

The beauty and the beast

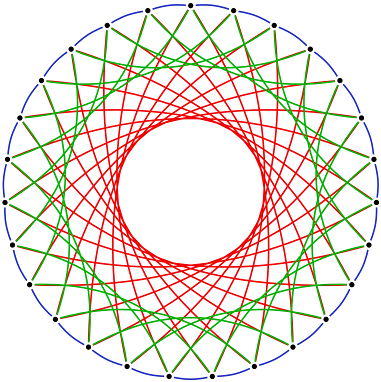
Components of the isogeny graphs look like this:



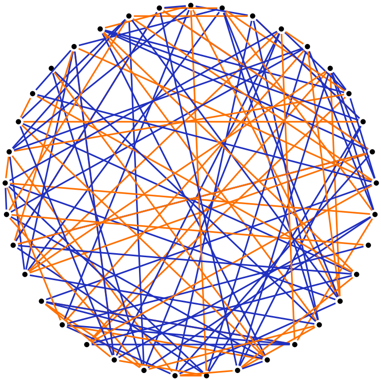
$S = \{3, 5, 7\}$, $q = 419$

The beauty and the beast

Components of the isogeny graphs look like this:



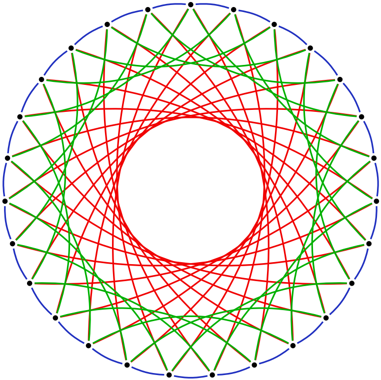
$S = \{3, 5, 7\}$, $q = 419$



$S = \{2, 3\}$, $q = 431^2$

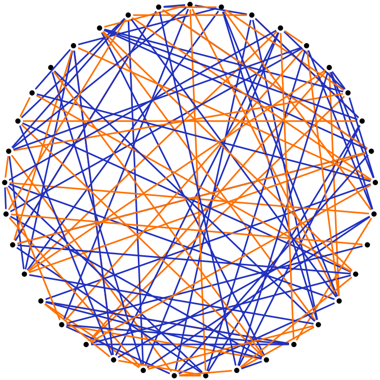
The beauty and the beast

For key exchange/KEM, there are <u>two families</u> of systems:



$q = p$

CSIDH [ˈsiːˌsaɪd]
https://csidh.isogeny.org

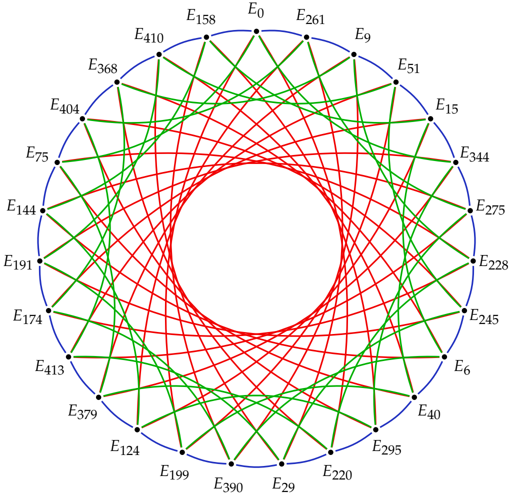


$q = p^2$

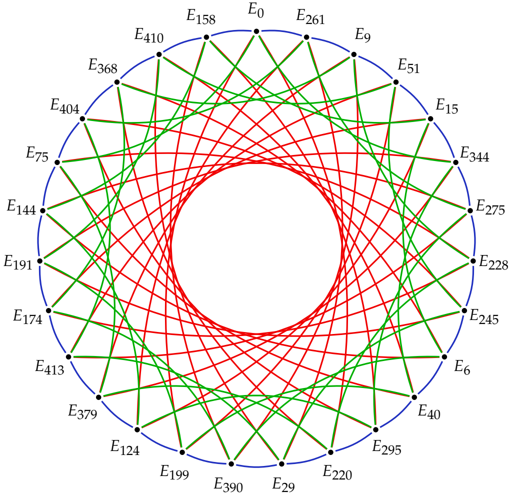
SIDH
https://sike.org


[ˈsiːˌsaɪd]

Isogeny graphs at the CSIDH

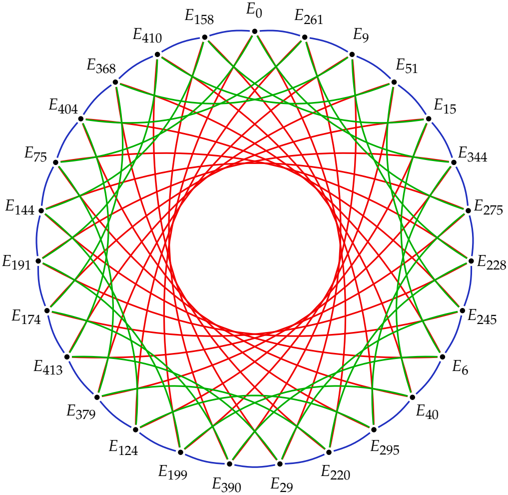


Isogeny graphs at the CSIDH



Nodes: Supersingular curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.

Isogeny graphs at the CSIDH



Nodes: Supersingular curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
Edges: 3-, 5-, and 7-isogenies.

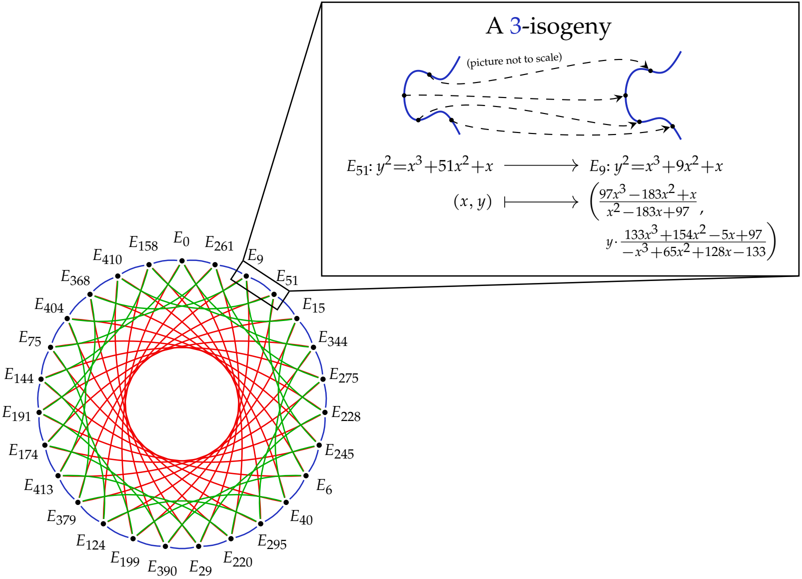
Quantumifying Exponentiation

- Idea to replace DLP: replace exponentiation

$$\begin{array}{rcl} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- Replace $\mathbb{Z}$ by a commutative group H that acts via isogenies.
- The action of $h \in H$ on S moves the elliptic curves one step around one of the cycles.

Graphs of elliptic curves



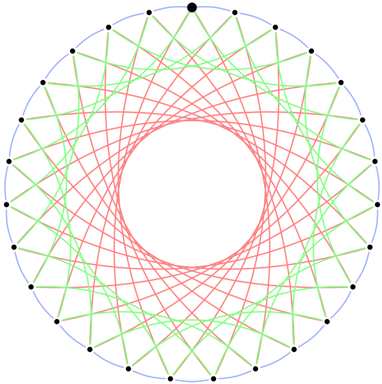
$$E_{51}\colon y^2 = x^3 + 51x^2 + x \longrightarrow E_9\colon y^2 = x^3 + 9x^2 + x$$

$$(x, y) \longmapsto \left(\frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97},\ y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right)$$

Diffie and Hellman go to the CSIDH

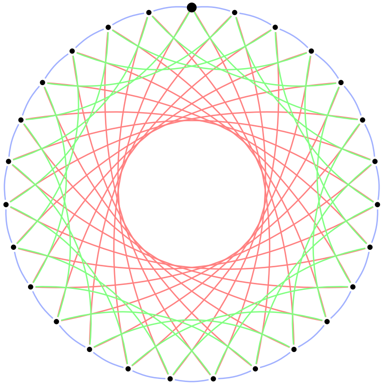
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$

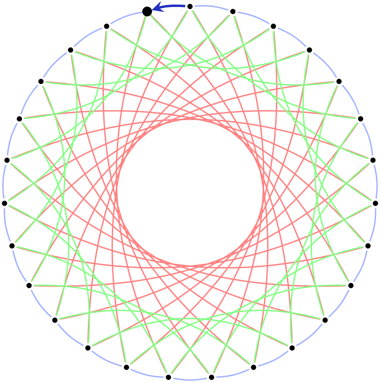


Diffie and Hellman go to the CSIDH

Alice

$[+, -, +, -]$

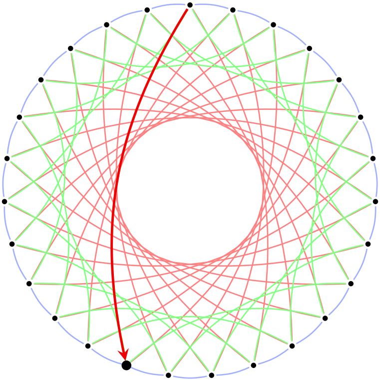
$\uparrow$



Bob

$[+, +, -, +]$

$\uparrow$

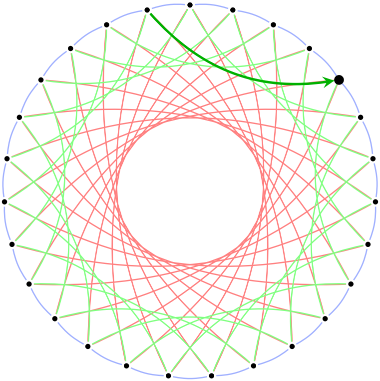


Diffie and Hellman go to the CSIDH

Alice

$[+, -, +, -]$

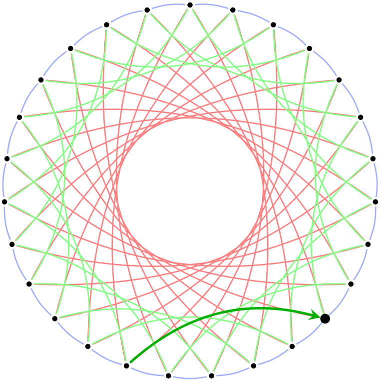
$\uparrow$



Bob

$[+, +, -, +]$

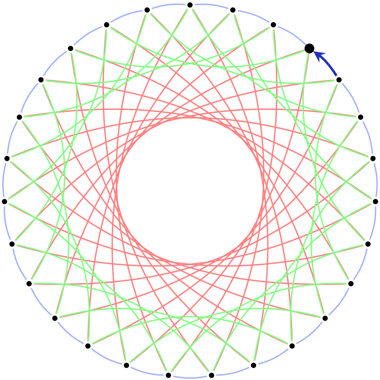
$\uparrow$



Diffie and Hellman go to the CSIDH

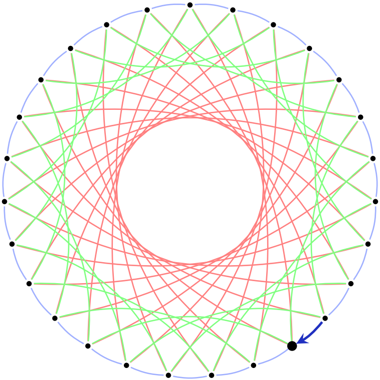
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$



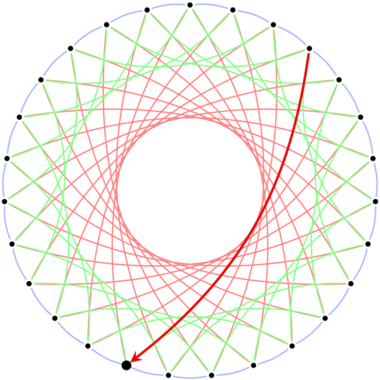
Diffie and Hellman go to the CSIDH

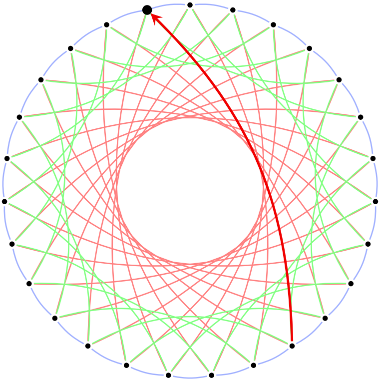
Alice

$[+, -, +, -]$
$\uparrow$

Bob

$[+, +, -, +]$
$\uparrow$

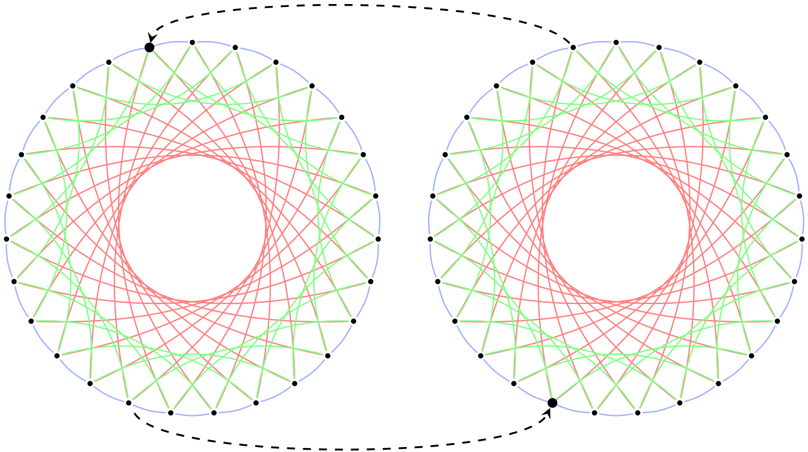




Diffie and Hellman go to the CSIDH

Alice
$[+, -, +, -]$

Bob
$[+, +, -, +]$

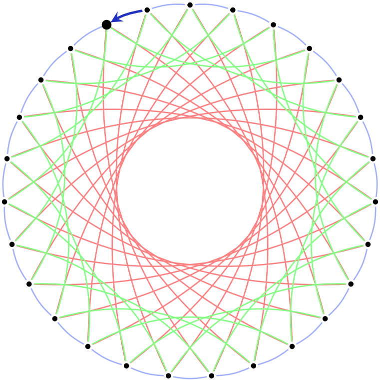


Diffie and Hellman go to the CSIDH

Alice

[+, −, +, −]

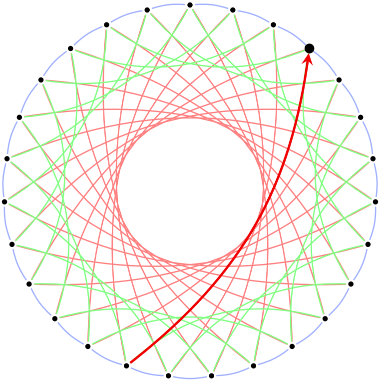
↑



Bob

[+, +, −, +]

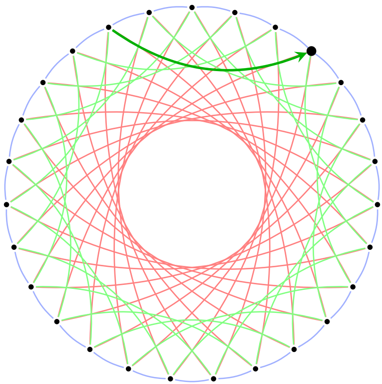
↑



Diffie and Hellman go to the CSIDH

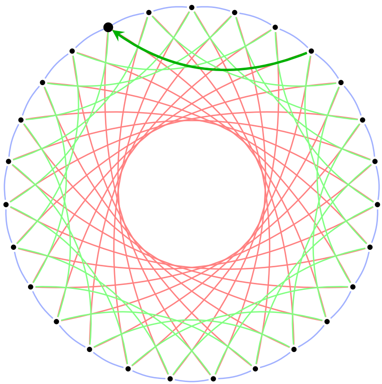
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$



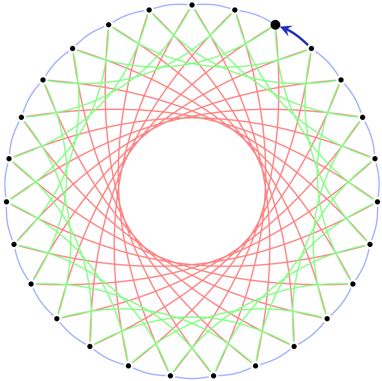
Diffie and Hellman go to the CSIDH

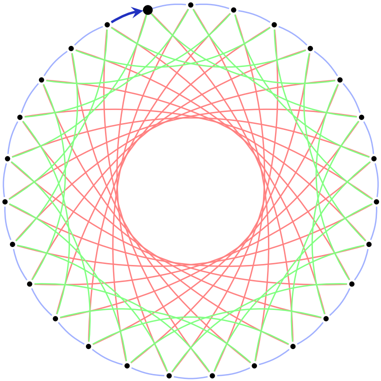
Alice

$[+, -, +, -]$
$\uparrow$

Bob

$[+, +, -, +]$
$\uparrow$



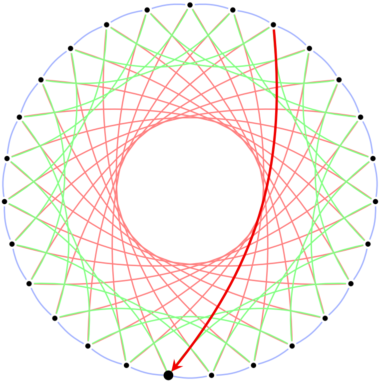


Diffie and Hellman go to the CSIDH

Alice

$[+, -, +, -]$

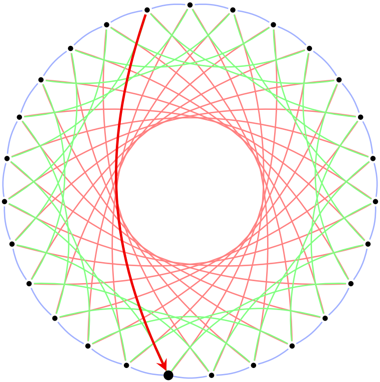
$\uparrow$



Bob

$[+, +, -, +]$

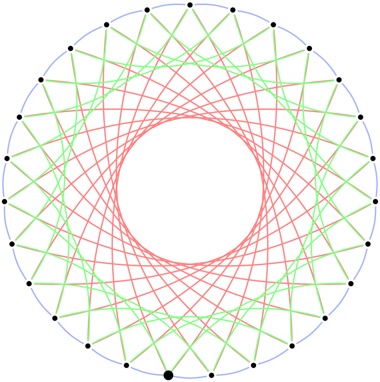
$\uparrow$



Diffie and Hellman go to the CSIDH

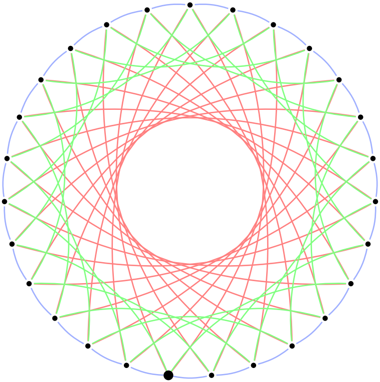
Alice

[+, −, +, −]



Bob

[+, +, −, +]



Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.

- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
  - Suppose we have found $P = E(\mathbb{F}_p)$ of order $p + 1$ or $(p + 1)/2$.

- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
  - Suppose we have found $P = E(\mathbb{F}_p)$ of order $p+1$ or $(p+1)/2$.
  - For every odd prime $\ell | (p+1)$, the point $\frac{p+1}{\ell} P$ is a point of order ℓ.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
  - Suppose we have found $P = E(\mathbb{F}_p)$ of order $p+1$ or $(p+1)/2$.
  - For every odd prime $\ell | (p+1)$, the point $\frac{p+1}{\ell} P$ is a point of order ℓ.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).
  - Given a $\mathbb{F}_p$-rational point of order ℓ, the isogeny computations can be done over $\mathbb{F}_p$.

Representing nodes of the graph

- Every node of G_{ℓ_i} is

$$E_A : y^2 = x^3 + Ax^2 + x.$$

Representing nodes of the graph

- Every node of G_{ℓ_i} is

$$E_A : y^2 = x^3 + Ax^2 + x.$$

$\Rightarrow$ Can compress every node to a single value $A \in \mathbb{F}_p$.

Representing nodes of the graph

- Every node of G_{ℓ_i} is

$$E_A\colon y^2 = x^3 + Ax^2 + x.$$

$\Rightarrow$ Can compress every node to a single value $A \in \mathbb{F}_p$.

$\Rightarrow$ Tiny keys!

Does any A work?

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p + 1$ points.

Does any A work?

No.

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p + 1$ points.

Does any A work?

No.

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p+1$ points.

Does any A work?

No.

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.
- Public-key validation: Check that E_A has $p + 1$ points.

  Easy Monte-Carlo algorithm: Pick random P on E_A and check $[p+1]P = \infty$.[1]

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p + 1$ points.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.
- Peikert's sieve technique [P19] on fastest variant of Kuperberg requires 2^{16} queries using 2^{40} bits of quantum accessible classical memory.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.
- Peikert's sieve technique [P19] on fastest variant of Kuperberg requires 2^{16} queries using 2^{40} bits of quantum accessible classical memory.
- For fastest variant of Kuperberg, total cost of CSIDH-512 attack is at least 2^{56} qubit operations.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.
- Peikert's sieve technique [P19] on fastest variant of Kuperberg requires 2^{16} queries using 2^{40} bits of quantum accessible classical memory.
- For fastest variant of Kuperberg, total cost of CSIDH-512 attack is at least 2^{56} qubit operations.
- Overheads from error correction, high quantum memory etc., not yet understood.

Venturing beyond the CSIDH

A selection of advances since original publication (2018):

- CSURF [CD19]: exploiting 2-isogenies.
- sqrtVelu [BDLS20]: square-root speed-up on computation of large-degree isogenies.
- Radical isogenies [CDV20]: significant speed-up on isogenies of small-ish degree.
- Some work on different curve forms (e.g. Edwards, Huff).
- Knowledge of $\mathrm{End}(E_0)$ and $\mathrm{End}(E_A)$ breaks CSIDH in classical polynomial time [Wes21].
- The SQALE of CSIDH [CCJR22]: carefully constructed CSIDH parameters less susceptible to Kuperberg's algorithm.
- CTIDH [$\mathrm{B}^2\mathrm{C}^2\mathrm{LMS}^2$]: Efficient constant-time CSIDH-style construction.

Now:

SIDH

Supersingular Isogeny Diffie–Hellman

Diffie-Hellman: High-level view

$$\begin{tikzcd}
g \arrow[r] \arrow[d] & g^{a} \arrow[d] \\
g^{b} \arrow[r] & g^{a*b}
\end{tikzcd}$$

SIDH: High-level view

$$
\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B \rangle
\end{tikzcd}
$$

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A : E \to E/A$; Bob computes $\varphi_B : E \to E/B$.

SIDH: High-level view

$$\begin{array}{ccc} E & \xrightarrow{\varphi_A} & E/A \\ {\scriptstyle\varphi_B}\downarrow & & \downarrow{\scriptstyle\varphi_{B'}} \\ E/B & \xrightarrow[\varphi_{A'}]{} & E/\langle A, B\rangle \end{array}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A : E \to E/A$; Bob computes $\varphi_B : E \to E/B$.
- Alice and Bob transmit the values E/A and E/B.

SIDH: High-level view

$$\begin{tikzcd} E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\ E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle \end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A : E \to E/A$; Bob computes $\varphi_B : E \to E/B$.
- Alice and Bob transmit the values E/A and E/B.
- Alice somehow obtains $A' := \varphi_B(A)$. (Similar for Bob.)

SIDH: High-level view

$$
\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}
$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A : E \to E/A$; Bob computes $\varphi_B : E \to E/B$.
- Alice and Bob transmit the values E/A and E/B.
- Alice somehow obtains $A' := \varphi_B(A)$. (Similar for Bob.)
- They both compute the shared secret

$$(E/B)/A' \cong E/\langle A, B\rangle \cong (E/A)/B'.$$

SIDH's auxiliary points

Previous slide: "Alice somehow obtains $A' := \varphi_B(A)$."

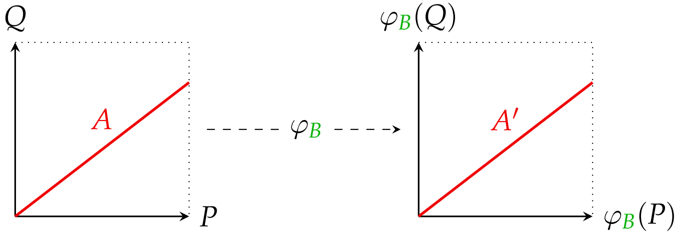
Alice knows only A, Bob knows only φ_B. Hm.

SIDH's auxiliary points

Previous slide: "Alice somehow obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B. Hm.

Solution: φ_B is a group homomorphism!

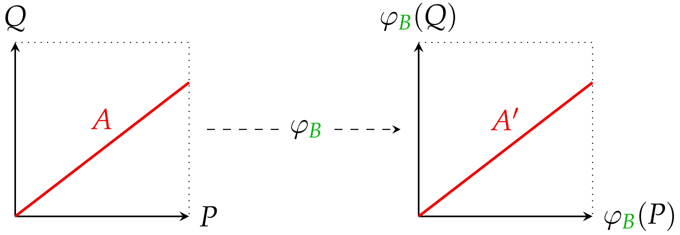


SIDH's auxiliary points

Previous slide: "Alice <u>somehow</u> obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B. Hm.

<u>Solution</u>: φ_B is a group homomorphism!



- 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 $\langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$!

SIDH in one slide

Public parameters:

- a large prime $p = 2^n 3^m - 1$ and a supersingular $E/\mathbb{F}_p$
- bases (P_A, Q_A) and (P_B, Q_B) of $E[2^n]$ and $E[3^m]$

Alice	public	Bob
$a \xleftarrow{\text{random}} \{0 \ldots 2^n - 1\}$		$b \xleftarrow{\text{random}} \{0 \ldots 3^m - 1\}$
$A := \langle P_A + [a] Q_A \rangle$ compute $\varphi_A : E \to E/A$		$B := \langle P_B + [b] Q_B \rangle$ compute $\varphi_B : E \to E/B$
$E/A,\ \varphi_A(P_B),\ \varphi_A(Q_B)$	$\longrightarrow$ $\longleftarrow$	$E/B,\ \varphi_B(P_A),\ \varphi_B(Q_A)$
$A' := \langle \varphi_B(P_A) + [a] \varphi_B(Q_A) \rangle$ $s := j\big((E/B)/A'\big)$		$B' := \langle \varphi_A(P_B) + [b] \varphi_A(Q_B) \rangle$ $s := j\big((E/A)/B'\big)$

Break it by: given public info, find secret key–φ_A or just A.

Security

Hard Problem:
Given

- supersingular public elliptic curves $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ connected by a secret 2^n-degree isogeny $\varphi_A : E_0 \to E_A$, and
- the action of φ_A on the 3^m-torsion of E_0,

find the secret key recover φ_A.

Security

Hard Problem:
Given

- supersingular public elliptic curves $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ connected by a secret 2^n-degree isogeny $\varphi_A : E_0 \to E_A$, and
- the action of φ_A on the 3^m-torsion of E_0,

find the secret key recover φ_A.

- Knowledge of $\mathrm{End}(E_0)$ and $\mathrm{End}(E_A)$ is sufficient to efficiently break it.
- Active attacker can recover secret.
- In SIDH, $\mathrm{End}(E_0)$ is fixed and $3^m \approx 2^n \approx \sqrt{p}$.
- If $3^m > 2^n$ or $3^m, 2^n > \sqrt{p}$, security claims are weakened.

Security of SIKE

- Best known attacks on SIKE, where $E_0/\mathbb{F}_p : y^2 = x^3 + x$ and $2^n \approx 3^m$ are on the Isogeny Problem:

Security of SIKE

- Best known attacks on SIKE, where $E_0/\mathbb{F}_p : y^2 = x^3 + x$ and $2^n \approx 3^m$ are on the Isogeny Problem:
  - The isogeny problem: given two elliptic curves, find an isogeny between them.

Security of SIKE

- Best known attacks on SIKE, where $E_0/\mathbb{F}_p : y^2 = x^3 + x$ and $2^n \approx 3^m$ are on the Isogeny Problem:
  - The isogeny problem: given two elliptic curves, find an isogeny between them.
- Best classical attack: meet-in-the-middle $O(p^{1/4})$.

Security of SIKE

- Best known attacks on SIKE, where $E_0/\mathbb{F}_p : y^2 = x^3 + x$ and $2^n \approx 3^m$ are on the Isogeny Problem:
  - The isogeny problem: given two elliptic curves, find an isogeny between them.
- Best classical attack: meet-in-the-middle $O(p^{1/4})$.
- Best quantum attack: meet-in-the-middle + Grover $O(p^{1/4})$, but slightly better in practise.

Security of SIKE

- Best known attacks on SIKE, where $E_0/\mathbb{F}_p : y^2 = x^3 + x$ and $2^n \approx 3^m$ are on the Isogeny Problem:
  - The isogeny problem: given two elliptic curves, find an isogeny between them.
- Best classical attack: meet-in-the-middle $O(p^{1/4})$.
- Best quantum attack: meet-in-the-middle + Grover $O(p^{1/4})$, but slightly better in practise.
- No commutative group action to exploit here*

What about signatures?

CSI-FiSh (S '06, D-G '18, Beullens-Kleinjung-Vercauteren '19)

Identification scheme from $H \times S \to S$:

Prover	**Public**	**Verifier**
	$E \in S, \mathfrak{l}_i \in H$	
$s_i \leftarrow \$\, \mathbb{Z}$		
$\mathsf{sk} = \prod \mathfrak{l}_i^{s_i}$,		
$\mathsf{pk} = \mathsf{sk} * E \xrightarrow{\mathsf{pk}}$	pk	
	$\xleftarrow{c}$	$c \leftarrow \$\, \{0, 1\}$
$t_i \leftarrow \$\, \mathbb{Z}$		
$\mathsf{esk} = \prod \mathfrak{l}_i^{t_i}$,		
$\mathsf{epk}_1 = \mathsf{esk} * E$,		
$\mathsf{epk}_2 = \mathsf{esk} \cdot \mathsf{sk}^{-c}$	$\xrightarrow{\mathsf{pk}, \mathsf{epk}_1, \mathsf{epk}_2}$	check:
		$\mathsf{epk}_1 = \mathsf{epk}_2 * ([\mathsf{sk}^c] * E)$.

After k challenges c, an imposter succeeds with prob 2^{-k}.

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

Hard Problem in CSIDH, CSI-FiSh, etc:
Given elliptic curves E and $E' \in S$, find $\mathfrak{a} \in H$ such that
$\mathfrak{a} * E = E'$.

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

Hard Problem in CSIDH, CSI-FiSh, etc:
Given elliptic curves E and $E' \in S$, find an isogeny* $E \to E'$

(*rational map + group homomorphism)

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

Hard Problem in CSIDH, CSI-FiSh, etc:
Given elliptic curves E and $E' \in S$, find an isogeny* $E \to E'$
(*rational map + group homomorphism)

SQISign is a newer signature scheme based on this idea:

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

Hard Problem in CSIDH, CSI-FiSh, etc:
Given elliptic curves E and $E' \in S$, find an isogeny* $E \to E'$
(*rational map + group homomorphism)

SQISign is a newer signature scheme based on this idea:

$$\begin{tikzcd} E \arrow[d] \\ E_{\mathrm{pk}} \end{tikzcd}$$

public, secret, ephemeral secret, public challenge, public proof

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

Hard Problem in CSIDH, CSI-FiSh, etc:
Given elliptic curves E and $E' \in S$, find an isogeny* $E \to E'$
(*rational map + group homomorphism)

SQISign is a newer signature scheme based on this idea:

$$\begin{tikzcd} E \arrow[r] \arrow[d] & E_{\mathrm{epk}} \\ E_{\mathrm{pk}} & \end{tikzcd}$$

public, secret, ephemeral secret, public challenge, public proof

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

Hard Problem in CSIDH, CSI-FiSh, etc:
Given elliptic curves E and $E' \in S$, find an isogeny* $E \to E'$

(* rational map + group homomorphism)

SQISign is a newer signature scheme based on this idea:

$$\begin{tikzcd} E \arrow[r] \arrow[d] & E_{\mathrm{epk}} \arrow[d] \\ E_{\mathrm{pk}} & E_{\mathrm{ver}} \end{tikzcd}$$

public, secret, ephemeral secret, public challenge, public proof

SQISign (De Feo-Kohel-Leroux-Petit-Wesolowski '20)

> Hard Problem in CSIDH, CSI-FiSh, etc:
> Given elliptic curves E and $E' \in S$, find an isogeny* $E \to E'$
> (*rational map + group homomorphism)

SQISign is a newer signature scheme based on this idea:

$$\begin{tikzcd} E \arrow[r] \arrow[d] & E_{\text{epk}} \arrow[d] \\ E_{\text{pk}} \arrow[r] & E_{\text{ver}} \end{tikzcd}$$

public, secret, ephemeral secret, public challenge, public proof

Summary and overview

- SIKE '11 KEM. Best-studied, in NIST, fast-ish, small, torsion-point attacks most likely attack avenue.

Summary and overview

- SIKE '11 KEM. Best-studied, in NIST, fast-ish, small, torsion-point attacks most likely attack avenue.
- CSIDH '18 Key exchange. Small, many applications (c.f. group actions), slow, known quantum attack needs further study, other attack avenues non-obvious.

Summary and overview

- SIKE '11 KEM. Best-studied, in NIST, fast-ish, small, torsion-point attacks most likely attack avenue.
- CSIDH '18 Key exchange. Small, many applications (c.f. group actions), slow, known quantum attack needs further study, other attack avenues non-obvious.
- CSI-FiSh '19 Digital signature. Small-ish, flexible, slow, known quantum attack reduces security below NIST Level I, hard to scale up.

Summary and overview

- SIKE ‘11 KEM. Best-studied, in NIST, fast-ish, small, torsion-point attacks most likely attack avenue.
- CSIDH ‘18 Key exchange. Small, many applications (c.f. group actions), slow, known quantum attack needs further study, other attack avenues non-obvious.
- CSI-FiSh ‘19 Digital signature. Small-ish, flexible, slow, known quantum attack reduces security below NIST Level I, hard to scale up.
- SQISign ‘20 Digital signature. Small, slow, clean security assumption, no known attack avenues.

Thank you!

References

[B²C²LMS²]	ctidh.isogeny.org
[BD17]	ia.cr/2017/334
[BDLS20]	velusqrt.isogeny.org
[BEG19]	ia.cr/2019/485
[BLMP19]	quantum.isogeny.org
[CCJR22]	ia.cr/2020/1520
[CD19]	ia.cr/2019/1404
[CDV20]	ia.cr/2020/1108
[FM19]	ia.cr/2019/555
[GMT19]	ia.cr/2019/431
[Wes21]	ia.cr/2021/1583