Quantum attacks on isogeny-based cryptography

Chloe Martindale

University of Bristol

Based on joint work with
Daniel J. Bernstein, Tanja Lange, and Lorenz Panny

quantum.isogeny.org


[ˈsiːˌsaɪd]

Why CSIDH?

- Drop-in post-quantum replacement for (EC)DH
- Non-interactive key exchange (full public-key validation); previously an open problem post-quantumly
- Smallest keys of all post-quantum key exchange candidates
- Competitive speed: 50-60ms for a full key exchange

Post-quantum Diffie-Hellman?

Traditionally, Diffie-Hellman works in a group G via the map

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

Post-quantum Diffie-Hellman?

Traditionally, Diffie-Hellman works in a group G via the map

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

Idea:

Replace exponentiation on the group G by a group action of a group H on a set S:

$$H \times S \to S.$$

Post-quantum Diffie-Hellman!

Traditionally, Diffie-Hellman works in a group G via the map

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

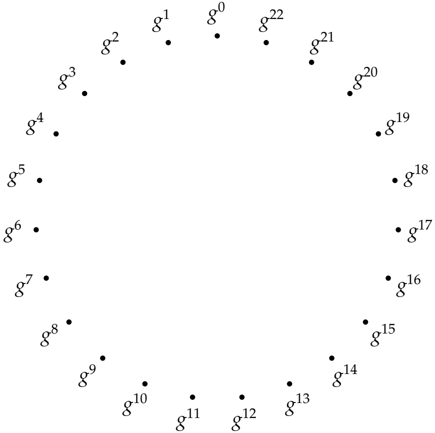
Idea:

Replace exponentiation on the group G by a group action of a group H on a set S:

$$H \times S \to S.$$

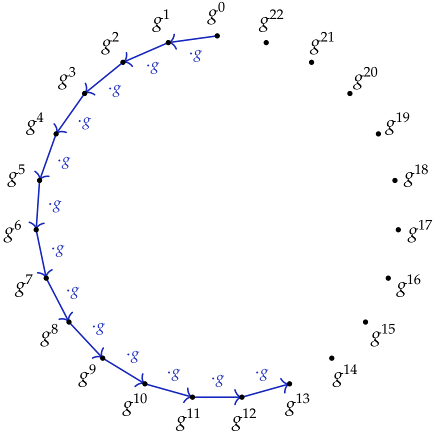
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes g^{13}.



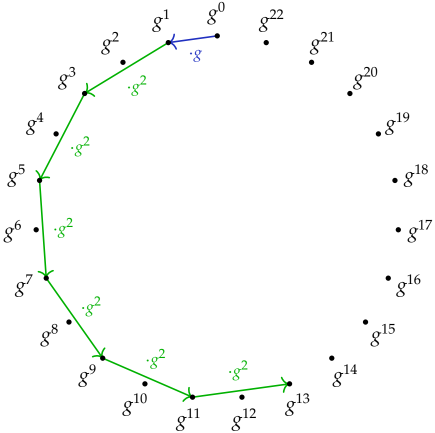
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes g^{13}.



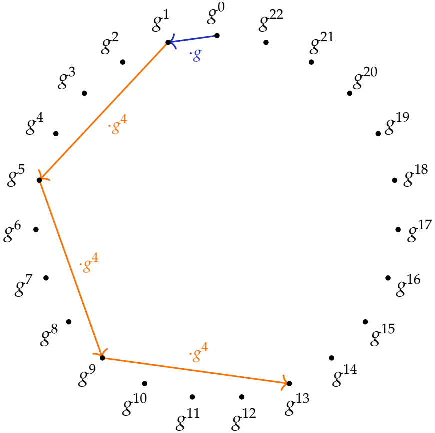
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes g^{13}.



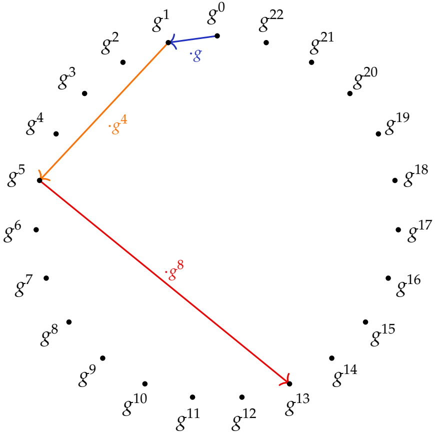
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes g^{13}.

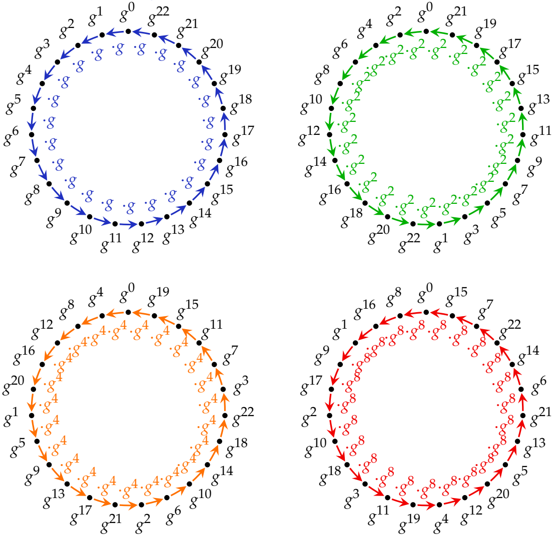


Square-and-multiply

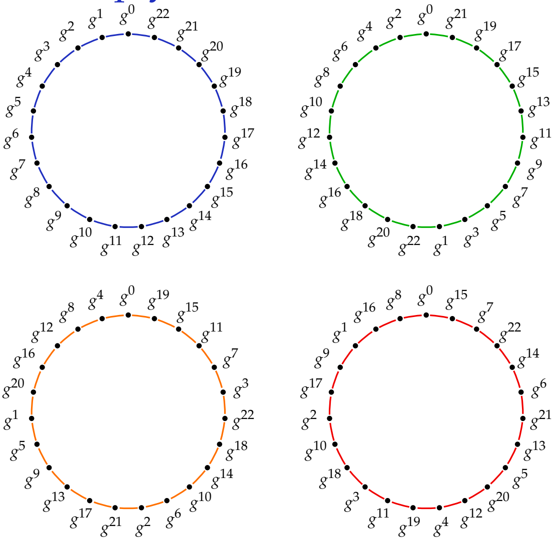
Suppose $G \cong \mathbb{Z}/23$ and that Alice computes g^{13}.



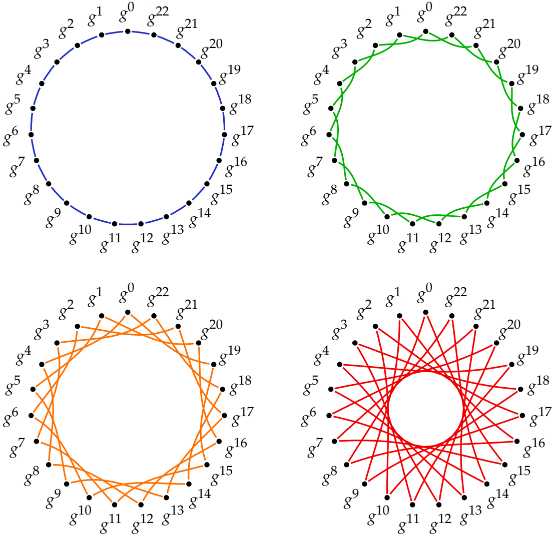
Square-and-multiply



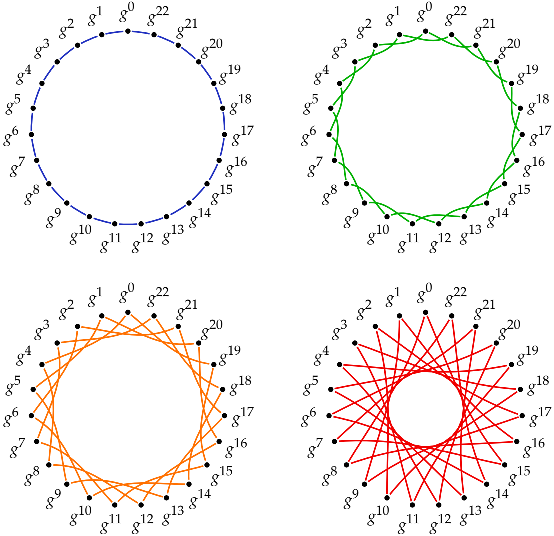
Square-and-multiply



Square-and-multiply

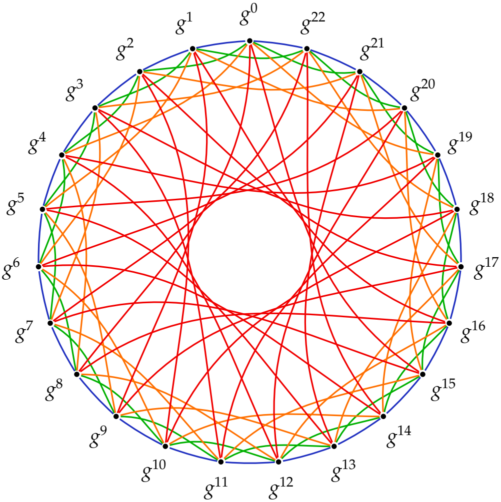


Square-and-multiply

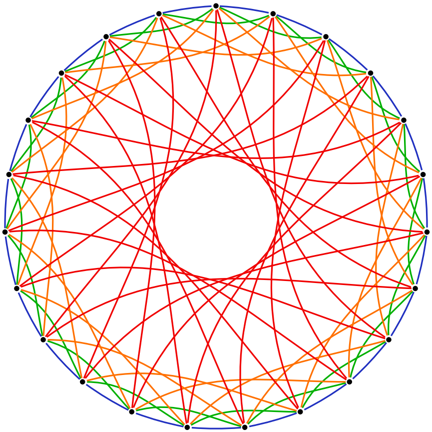


Cycles are compatible: [right, then left] = [left, then right], etc.

Union of cycles: rapid mixing

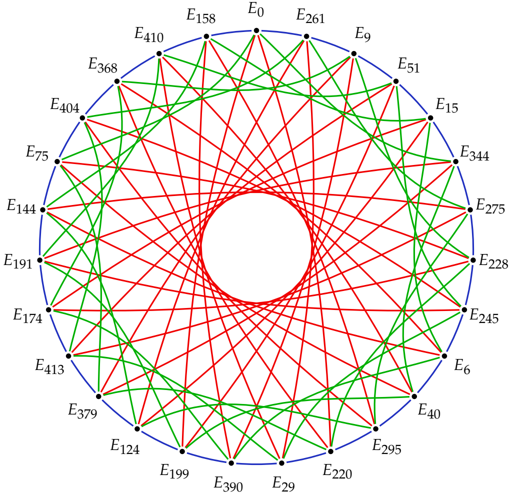


Union of cycles: rapid mixing

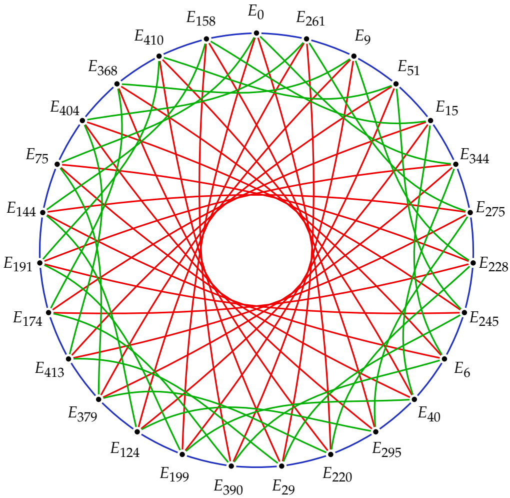


CSIDH: Nodes are now elliptic curves and edges are isogenies.

Graphs of elliptic curves

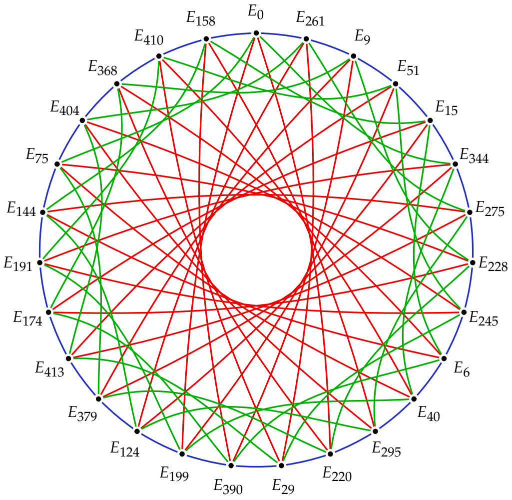


Graphs of elliptic curves



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

Graphs of elliptic curves



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

Graphs of elliptic curves

- $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with $A \in \mathbb{F}_p - \{\pm 2\}$ are examples of elliptic curves.

Graphs of elliptic curves

- $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with $A \in \mathbb{F}_p - \{\pm 2\}$ are examples of elliptic curves.
  - The set of $\mathbb{F}_p$-rational points of E_A form a group $E_A(\mathbb{F}_p)$.

Graphs of elliptic curves

- $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with $A \in \mathbb{F}_p - \{\pm 2\}$ are examples of elliptic curves.
  - The set of $\mathbb{F}_p$-rational points of E_A form a group $E_A(\mathbb{F}_p)$.
  - If $\#E_A(\mathbb{F}_p) = p + 1$ then E_A is supersingular.

Graphs of elliptic curves

- $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with $A \in \mathbb{F}_p - \{\pm 2\}$ are examples of elliptic curves.
    - The set of $\mathbb{F}_p$-rational points of E_A form a group $E_A(\mathbb{F}_p)$.
    - If $\#E_A(\mathbb{F}_p) = p + 1$ then E_A is supersingular.

- A rational map $E_A \to E_B$ is an isogeny if it preserves the group structure and is surjective.

Graphs of elliptic curves

- $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with $A \in \mathbb{F}_p - \{\pm 2\}$ are examples of elliptic curves.
  - The set of $\mathbb{F}_p$-rational points of E_A form a group $E_A(\mathbb{F}_p)$.
  - If $\#E_A(\mathbb{F}_p) = p + 1$ then E_A is supersingular.
- A rational map $E_A \to E_B$ is an isogeny if it preserves the group structure and is surjective.
  - Isogenies have finite kernel.

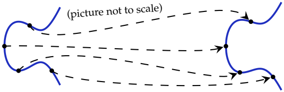
Graphs of elliptic curves

- $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with $A \in \mathbb{F}_p - \{\pm 2\}$ are examples of elliptic curves.
  - The set of $\mathbb{F}_p$-rational points of E_A form a group $E_A(\mathbb{F}_p)$.
  - If $\#E_A(\mathbb{F}_p) = p + 1$ then E_A is supersingular.
- A rational map $E_A \to E_B$ is an isogeny if it preserves the group structure and is surjective.
  - Isogenies have finite kernel.
  - Vélu's formulas:

$$\text{generators of kernel} \rightsquigarrow \text{rational maps}$$

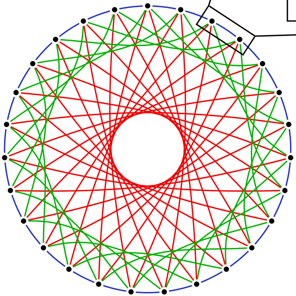
Graphs of elliptic curves

A 3-isogeny



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

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



Quantumifying Exponentiation

- We want to replace the exponentiation map

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

by a group action on a set.

Quantumifying Exponentiation

- We want to replace the exponentiation map

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

  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 $\mathbb{F}_{419}$.

Quantumifying Exponentiation

- We want to replace the exponentiation map

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

  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 by isogenies.*

Quantumifying Exponentiation

- We want to replace the exponentiation map

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

  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 by isogenies.*
- The action of a well-chosen $h \in H$ on S moves the elliptic curves one step around one of the cycles.

Quantumifying Exponentiation

- We want to replace the exponentiation map

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

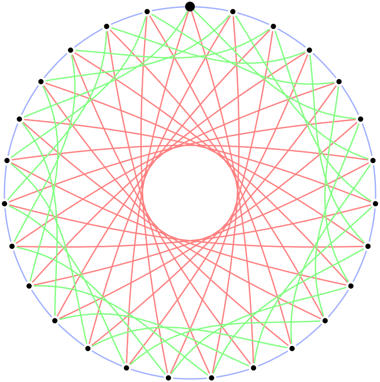
  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- Replace $\mathbb{Z}$ by a commutative group H that acts by isogenies.*
- The action of a well-chosen $h \in H$ on S moves the elliptic curves one step around one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

*Die-hards: $H = \mathrm{cl}(\mathrm{End}_{\mathbb{F}_p}(E)) = \mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$; an ideal class $[I] \in H$ defines the kernel.

Diffie-Hellman on ‘nice’ graphs

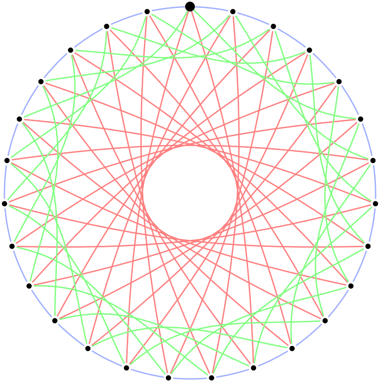
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$



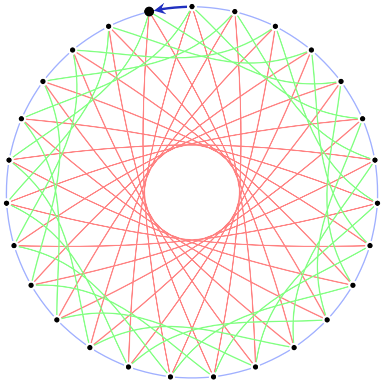
Diffie-Hellman on ‘nice’ graphs

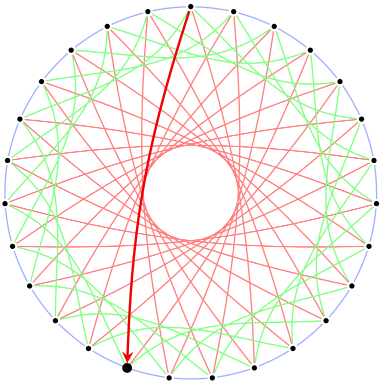
Alice

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

Bob

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



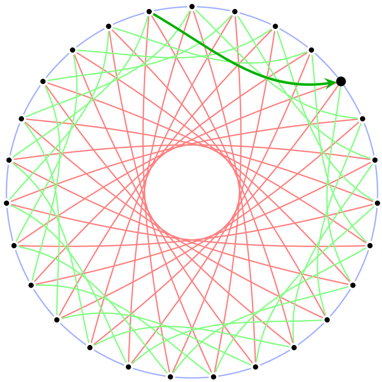


Diffie-Hellman on ‘nice’ graphs

Alice

$[+, -, +, -]$

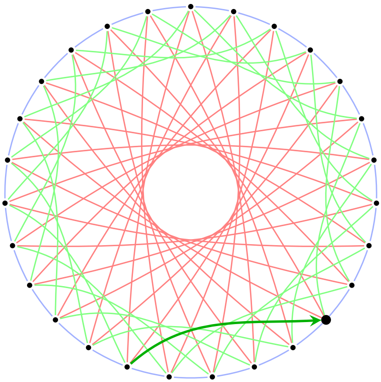
$\uparrow$



Bob

$[+, +, -, +]$

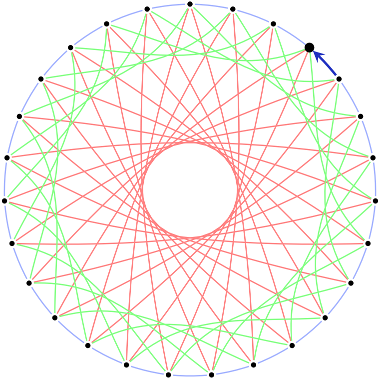
$\uparrow$



Diffie-Hellman on 'nice' graphs

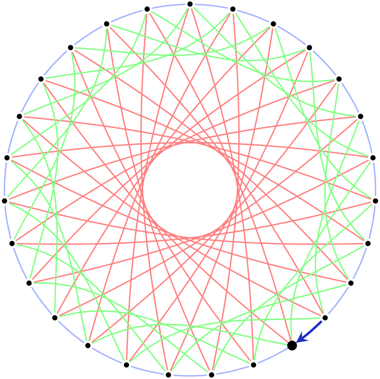
Alice

$[+, -, +, -]$



Bob

$[+, +, -, +]$



Diffie-Hellman on ‘nice’ graphs

Alice

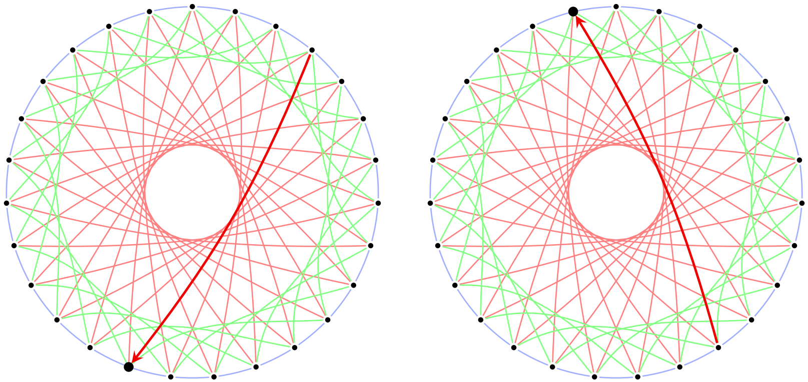
$[+, -, +, -]$

$\uparrow$

Bob

$[+, +, -, +]$

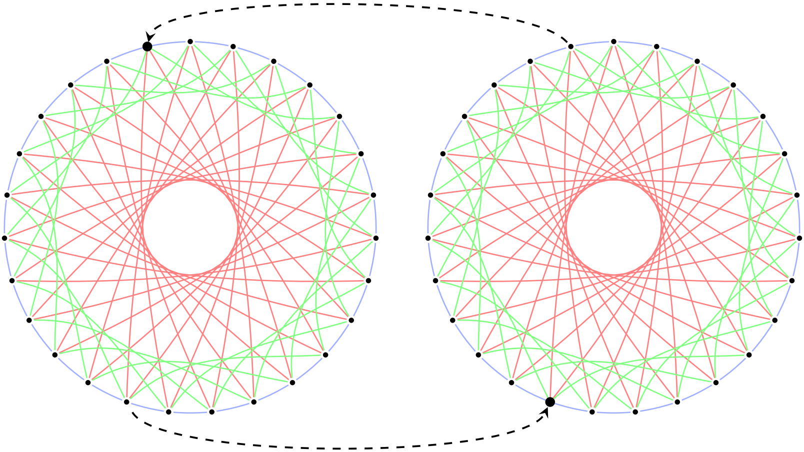
$\uparrow$



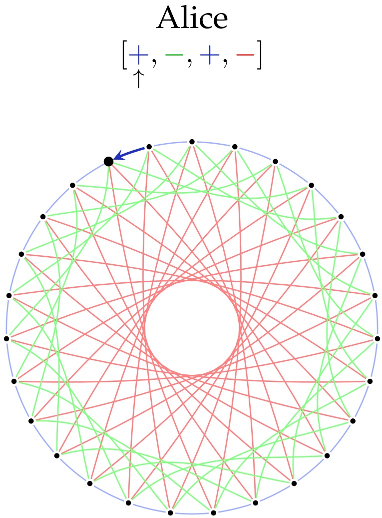
Diffie-Hellman on 'nice' graphs

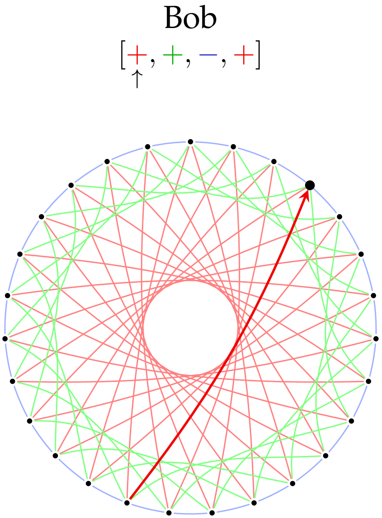
Alice $[+, -, +, -]$

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



Diffie-Hellman on 'nice' graphs

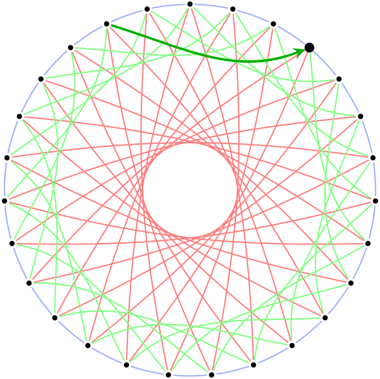




Diffie-Hellman on 'nice' graphs

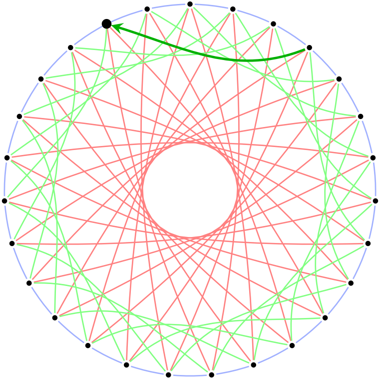
Alice

$[{\color{blue}+}, {\color{green}-}, {\color{blue}+}, {\color{red}-}]$
$\uparrow$



Bob

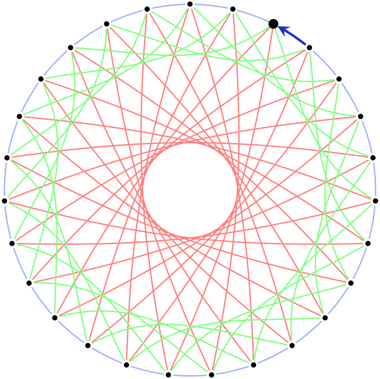
$[{\color{red}+}, {\color{green}+}, {\color{blue}-}, {\color{red}+}]$
$\uparrow$



Diffie-Hellman on ‘nice’ graphs

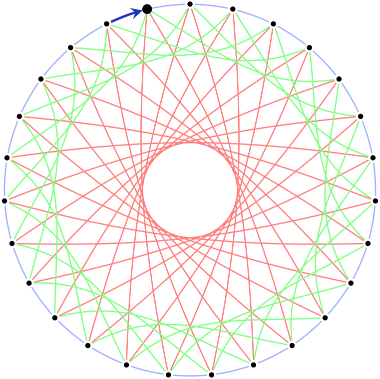
Alice

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



Bob

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



Diffie-Hellman on 'nice' graphs

Alice

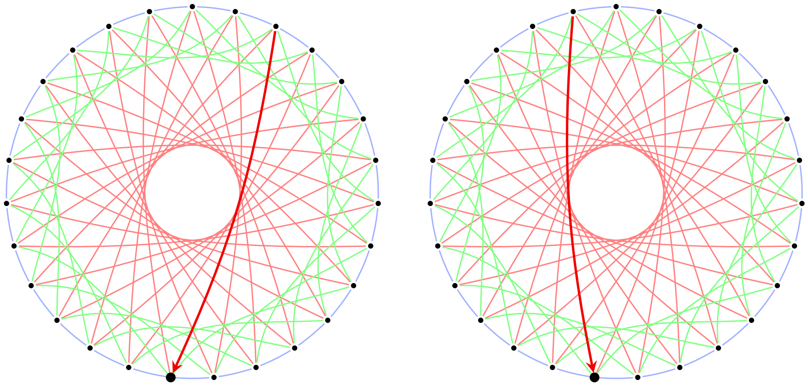
$[+, -, +, -]$

$\uparrow$

Bob

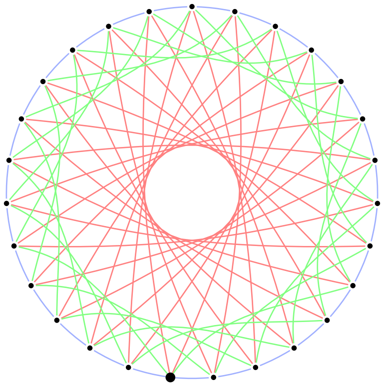
$[+, +, -, +]$

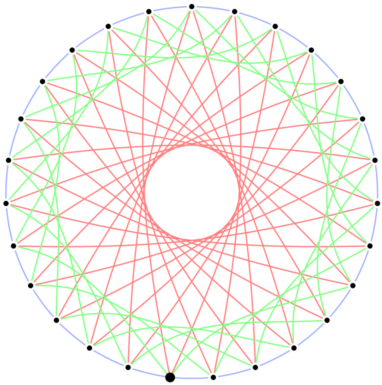
$\uparrow$



Diffie-Hellman on 'nice' graphs

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





Quantum security: from CSIDH to DHSP

- Hard problem in CSIDH: given group action

  $$H \times S \to S$$

  and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.

Quantum security: from CSIDH to DHSP

- Hard problem in CSIDH: given group action

  $$H \times S \to S$$

  and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- [CJS]: this is a dihedral hidden subgroup problem (DHSP).

Quantum security: from CSIDH to DHSP

- Hard problem in CSIDH: given group action

$$H \times S \to S$$

  and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- [CJS]: this is a dihedral hidden subgroup problem (DHSP).

Recall the dihedral group: $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$

Quantum security: from CSIDH to DHSP

- Hard problem in CSIDH: given group action
$$H \times S \to S$$
and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- [CJS]: this is a dihedral hidden subgroup problem (DHSP).

Recall the dihedral group: $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ where

- $\varphi : \mathbb{Z}/2\mathbb{Z} \to \mathrm{Aut}(\mathbb{Z}/n\mathbb{Z})$ a homomorphism, and
- $$\begin{array}{rcl} (\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}) & \to & \mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z} \\ (a_1, b_1), (a_2, b_2) & \mapsto & (a_1\varphi(b_1)(a_2), b_1 b_2). \end{array}$$

Quantum security: from CSIDH to DHSP

- CSIDH: given cyclic H and group action
$$H \times S \to S$$
and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.

Quantum security: from CSIDH to DHSP

- CSIDH: given cyclic H and group action
$$H \times S \to S$$
and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- Define
$$\begin{array}{rccc} \varphi : & \mathbb{Z}/2\mathbb{Z} & \to & \mathrm{Aut}(H) \\ & a & \mapsto & (h \mapsto h^{(-1)^a}). \end{array}$$

Quantum security: from CSIDH to DHSP

- CSIDH: given cyclic H and group action
$$H \times S \to S$$
and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- Define
$$\begin{array}{rccc} \varphi : & \mathbb{Z}/2\mathbb{Z} & \to & \mathrm{Aut}(H) \\ & a & \mapsto & (h \mapsto h^{(-1)^a}). \end{array}$$
- Define
$$\begin{array}{rccc} f : & H \rtimes_\varphi \mathbb{Z}/2\mathbb{Z} & \to & S \\ & (h, a) & \mapsto & h \cdot s_a. \end{array}$$

Quantum security: from CSIDH to DHSP

- CSIDH: given cyclic H and group action
$$H \times S \to S$$
and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- Define
$$\begin{array}{rccc} \varphi : & \mathbb{Z}/2\mathbb{Z} & \to & \mathrm{Aut}(H) \\ & a & \mapsto & (h \mapsto h^{(-1)^a}). \end{array}$$
- Define
$$\begin{array}{rccc} f : & H \rtimes_\varphi \mathbb{Z}/2\mathbb{Z} & \to & S \\ & (h, a) & \mapsto & h \cdot s_a. \end{array}$$
- Now
$$\begin{aligned} f(h,a) = f(h',a') \Leftrightarrow & \, a = 0,\, a' = 1,\, h' = h\chi, \text{ or} \\ & \, a = 1,\, a' = 0,\, h = h'\chi, \text{ or} \\ & \, a = a' = 1,\, h = h'. \end{aligned}$$

$\rightsquigarrow$ f hides the subgroup $\{(1,0),(\chi,1)\} \subset H \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$.

Quantum security: from CSIDH to DHSP

- CSIDH: given cyclic H and group action
$$H \times S \to S$$
and $s_0, s_1 \in S$, find $\chi \in H$ such that $\chi \cdot s_0 = s_1$.
- Define
$$\begin{array}{rccc} \varphi: & \mathbb{Z}/2\mathbb{Z} & \to & \mathrm{Aut}(H) \\ & a & \mapsto & (h \mapsto h^{(-1)^a}). \end{array}$$
- Define
$$\begin{array}{rccc} f: & H \rtimes_\varphi \mathbb{Z}/2\mathbb{Z} & \to & S \\ & (h,a) & \mapsto & h \cdot s_a. \end{array}$$
- Now
$$\begin{aligned} f(h,a) = f(h',a') \Leftrightarrow & \, a = 0,\ a' = 1,\ h' = h\chi, \text{ or} \\ & \, a = 1,\ a' = 0,\ h = h'\chi, \text{ or} \\ & \, a = a' = 1,\ h = h'. \end{aligned}$$
$\rightsquigarrow f$ hides the subgroup $\{(1,0),(\chi,1)\} \subset H \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$.
- Finding subgroup hidden by f gives secret χ.

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_{\varphi} \mathbb{Z}/2\mathbb{Z}$ using

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.
- 2004: Regev gives variant with polynomial number of qubits and exponential time.

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.
- 2004: Regev gives variant with polynomial number of qubits and exponential time.
- 2011: Kuperberg gives more trade-offs and improvements. Best time using (only) subexponential number of qubits: $2^{(\sqrt{2}+o(1))\sqrt{\log_2 n}}$.

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.
- 2004: Regev gives variant with polynomial number of qubits and exponential time.
- 2011: Kuperberg gives more trade-offs and improvements. Best time using (only) subexponential number of qubits: $2^{(\sqrt{2}+o(1))\sqrt{\log_2 n}}$.

Main open questions on asymptotic complexity:

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.
- 2004: Regev gives variant with polynomial number of qubits and exponential time.
- 2011: Kuperberg gives more trade-offs and improvements. Best time using (only) subexponential number of qubits: $2^{(\sqrt{2}+o(1))\sqrt{\log_2 n}}$.

Main open questions on asymptotic complexity:

- Can the power of $\log_2 n$ be reduced?

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.
- 2004: Regev gives variant with polynomial number of qubits and exponential time.
- 2011: Kuperberg gives more trade-offs and improvements. Best time using (only) subexponential number of qubits: $2^{(\sqrt{2}+o(1))\sqrt{\log_2 n}}$.

Main open questions on asymptotic complexity:

- Can the power of $\log_2 n$ be reduced?
- If not, can the constant $\sqrt{2}$ be improved?

Quantum complexity analysis

- 2003: Kuperberg gives quantum algorithm for DHSP in $\mathbb{Z}/n\mathbb{Z} \rtimes_\varphi \mathbb{Z}/2\mathbb{Z}$ using
  - $\exp((\log_2 n)^{1/2+o(1)})$ queries
  - $\exp((\log_2 n)^{1/2+o(1)})$ ops on $\exp((\log_2 n)^{1/2+o(1)})$ qubits.
- 2004: Regev gives variant with polynomial number of qubits and exponential time.
- 2011: Kuperberg gives more trade-offs and improvements. Best time using (only) subexponential number of qubits: $2^{(\sqrt{2}+o(1))\sqrt{\log_2 n}}$.

Main open questions on asymptotic complexity:

- Can the power of $\log_2 n$ be reduced?
- If not, can the constant $\sqrt{2}$ be improved?
- If not, what's the smallest o(1)?

Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Subquestions:

- Exactly how many queries needed?

Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Subquestions:

- Exactly how many queries needed?
- How expensive is each CSIDH query?

Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Subquestions:

- Exactly how many queries needed?
- How expensive is each CSIDH query?
- How is attack affected by occasional errors and non-uniform distributions over the group?

Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Subquestions:

- Exactly how many queries needed?
- How expensive is each CSIDH query?
- How is attack affected by occasional errors and non-uniform distributions over the group?
- What about memory, using parallel *AT* metric? Are trade-offs worth it: (theoretically) fastest variant uses billions of qubits.

Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Subquestions:

- Exactly how many queries needed?
- How expensive is each CSIDH query?
- How is attack affected by occasional errors and non-uniform distributions over the group?
- What about memory, using parallel AT metric? Are trade-offs worth it: (theoretically) fastest variant uses billions of qubits.

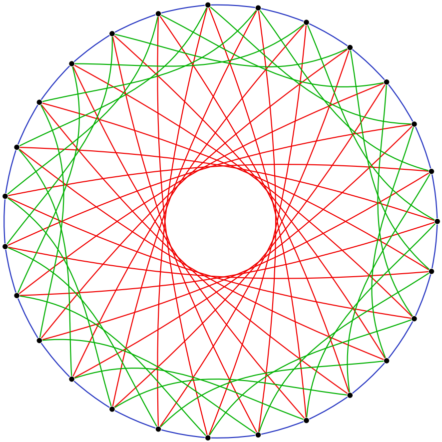
Concrete quantum complexity analysis

What CSIDH key sizes are needed for concrete post-quantum security levels 2^{64}? 2^{96}? 2^{128}?

Subquestions:

- Exactly how many queries needed?
- How expensive is each CSIDH query?
- How is attack affected by occasional errors and non-uniform distributions over the group?
- What about memory, using parallel *AT* metric? Are trade-offs worth it: (theoretically) fastest variant uses billions of qubits.

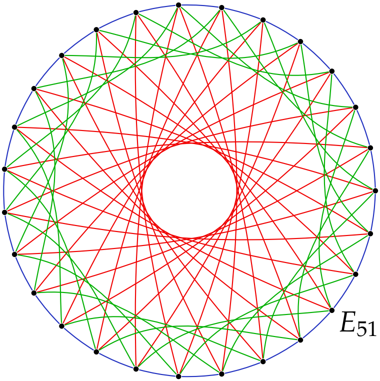
How expensive is each CSIDH query?



Secret key: path on the graph
Public key: end points of path
One query: computes many paths in superposition

Computing isogenies

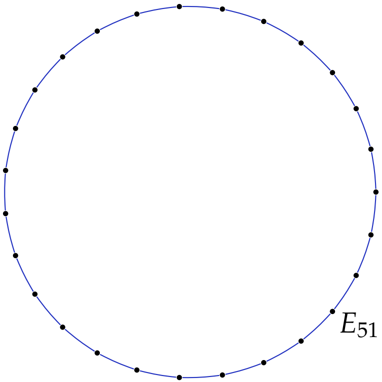
Aim: given curve E_A, find a neighbour in the isogeny graph



Edges: 3-, 5-, and 7-isogenies.

Computing isogenies

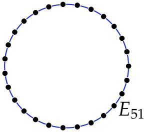
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



Edges: 3-isogenies.

Computing isogenies

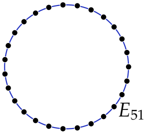
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.

Computing isogenies

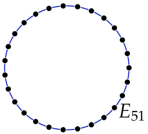
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}

Computing isogenies

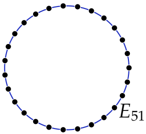
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.

Computing isogenies

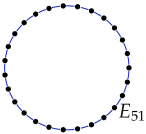
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{2}{3}$, $140 \cdot P$ has order 3

Computing isogenies

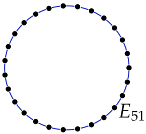
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{2}{3}$, $140 \cdot P$ has order 3
- Using Vélu's formulas, find map with kernel = $\langle 140 \cdot P \rangle$

Computing isogenies

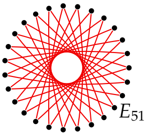
Aim: given curve E_A, find a neighbour in the 3-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{2}{3}$, $140 \cdot P$ has order 3
- Using Vélu's formulas, find map with kernel = $\langle 140 \cdot P \rangle$
- Image of map is a neighbour

Computing isogenies

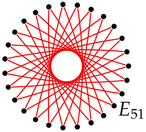
Aim: given curve E_A, find a neighbour in the 5-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{2}{3}$, $140 \cdot P$ has order 3
- Using Vélu's formulas, find map with kernel = $\langle 140 \cdot P \rangle$
- Image of map is a neighbour

Computing isogenies

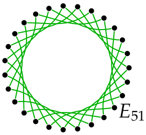
Aim: given curve E_A, find a neighbour in the 5-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{4}{5}$, $84 \cdot P$ has order 5
- Using Vélu's formulas, find map with kernel = $\langle 84 \cdot P \rangle$
- Image of map is a neighbour

Computing isogenies

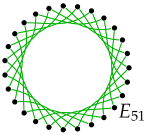
Aim: given curve E_A, find a neighbour in the 7-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{4}{5}$, $84 \cdot P$ has order 5
- Using Vélu's formulas, find map with kernel = $\langle 84 \cdot P \rangle$
- Image of map is a neighbour

Computing isogenies

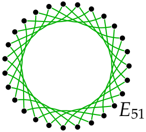
Aim: given curve E_A, find a neighbour in the 7-isogeny graph



- Recall: $E_{51}/\mathbb{F}_{419} : y^2 = x^3 + 51x^2 + x$.
- Choose a random $\mathbb{F}_{419}$-point $P = (x, y)$ on E_{51}
- P has order dividing 420.
- With probability $\frac{6}{7}$, $60 \cdot P$ has order 7
- Using Vélu's formulas, find map with kernel = $\langle 60 \cdot P \rangle$
- Image of map is a neighbour

Computing isogenies

Aim: given curve E_A, find a neighbour in the ℓ-isogeny graph



- Recall: $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$
- Choose a random $\mathbb{F}_p$-point $P = (x, y)$ on E_A
- P has order dividing $p + 1$.
- With probability $\frac{\ell-1}{\ell}$, $\frac{p+1}{\ell} \cdot P$ has order ℓ.*
- Using Vélu's formulas, find map with kernel = $\langle \frac{p+1}{\ell} \cdot P \rangle$
- Image of map is a neighbour

* assuming $\ell | (p + 1)$.

Computing a query

- A query computes paths in superposition.

Computing a query

- A query computes paths in superposition.
- A path is a sequence of isogenies (of varying degrees).

Computing a query

- A query computes paths in superposition.
- A path is a sequence of isogenies (of varying degrees).
- Larger degree isogenies are more expensive.

Computing a query

- A query computes paths in superposition.
- A path is a sequence of isogenies (of varying degrees).
- Larger degree isogenies are more expensive.
  Different degrees computed in superposition
  $\rightsquigarrow$ bored qubits.

Computing a query

- A query computes paths in superposition.
- A path is a sequence of isogenies (of varying degrees).
- Larger degree isogenies are more expensive.
  Different degrees computed in superposition
  $\rightsquigarrow$ bored qubits.
- Isogeny computation fails often for small ℓ.

Computing a query

- A query computes paths in superposition.
- A path is a sequence of isogenies (of varying degrees).
- Larger degree isogenies are more expensive.
  Different degrees computed in superposition
  $\rightsquigarrow$ bored qubits.
- Isogeny computation fails often for small ℓ.
  $\rightsquigarrow$ problematic for quantum implementation.

Computing a query

- A query computes paths in superposition.
- A path is a sequence of isogenies (of varying degrees).
- Larger degree isogenies are more expensive.
  Different degrees computed in superposition
  $\rightsquigarrow$ bored qubits.
- Isogeny computation fails often for small ℓ.
  $\rightsquigarrow$ problematic for quantum implementation.

[BLMP] Gives many optimizations / more complex variants–trying to mitigate these problems.

Computing a query

[BLMP] provides software to compute a path using basic bit operations: automatic tallies of nonlinear ops (AND, OR) and linear ops (XOR, NOT).

Computing a query

[BLMP] provides software to compute a path using basic bit operations: automatic tallies of nonlinear ops (AND, OR) and linear ops (XOR, NOT).

We then apply a generic conversion:

Computing a query

[BLMP] provides software to compute a path using basic bit operations: automatic tallies of nonlinear ops (AND, OR) and linear ops (XOR, NOT).

We then apply a generic conversion:

sequence of basic bit ops with $\leq B$ nonlinear ops	$\rightsquigarrow$	sequence of reversible ops with $\leq 2B$ Toffoli ops	$\rightsquigarrow$	sequence of reversible ops with $\leq 14B$ T-gates

Computing a query

[BLMP] provides software to compute a path using basic bit operations: automatic tallies of nonlinear ops (AND, OR) and linear ops (XOR, NOT).

We then apply a generic conversion:

sequence of basic bit ops with $\leq B$ nonlinear ops	$\rightsquigarrow$	sequence of reversible ops with $\leq 2B$ Toffoli ops	$\rightsquigarrow$	sequence of reversible ops with $\leq 14B$ T-gates

Why this generic conversion?

Unknown expense of extra $O(B)$ measurements in context of surface-code error correction

Computing a query

[BLMP] provides software to compute a path using basic bit operations: automatic tallies of nonlinear ops (AND, OR) and linear ops (XOR, NOT).

We then apply a generic conversion:

sequence of basic bit ops with $\leq B$ nonlinear ops	$\rightsquigarrow$	sequence of reversible ops with $\leq 2B$ Toffoli ops	$\rightsquigarrow$	sequence of reversible ops with $\leq 14B$ T-gates

Why this generic conversion?

Unknown expense of extra $O(B)$ measurements in context of surface-code error correction

Open question:
How much faster than the generic conversion is possible?

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with

$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$

where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with
$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$
where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.
- Note that each ℓ_i divides $p + 1$.

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with

$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$

  where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.
- Note that each ℓ_i divides $p + 1$.
- For an error rate of $< 2^{-32}$, our best algorithm requires

$$765325228976 \approx 0.7 \cdot 2^{40}$$

  nonlinear bit operations. Previous record was 2^{51}.

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with

$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$

  where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.
- Note that each ℓ_i divides $p + 1$.
- For an error rate of $< 2^{-32}$, our best algorithm requires

$$765325228976 \approx 0.7 \cdot 2^{40}$$

  nonlinear bit operations. Previous record was 2^{51}.
- Generic conversion gives $\approx 2^{43.3}$ T-gates using 2^{40} qubits.

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with

$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$

where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.
- Note that each ℓ_i divides $p + 1$.
- For an error rate of $< 2^{-32}$, our best algorithm requires

$$765325228976 \approx 0.7 \cdot 2^{40}$$

nonlinear bit operations. Previous record was 2^{51}.
- Generic conversion gives $\approx 2^{43.3}$ T-gates using 2^{40} qubits.
- Can do $\approx 2^{45.3}$ T-gates using $\approx 2^{20}$ qubits.

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with

$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$

  where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.
- Note that each ℓ_i divides $p + 1$.
- For an error rate of $< 2^{-32}$, our best algorithm requires

$$765325228976 \approx 0.7 \cdot 2^{40}$$

  nonlinear bit operations. Previous record was 2^{51}.
- Generic conversion gives $\approx 2^{43.3}$ T-gates using 2^{40} qubits.
- Can do $\approx 2^{45.3}$ T-gates using $\approx 2^{20}$ qubits.
- Total gates for one query (T+Clifford): $\approx 2^{46.9}$.

Case study: CSIDH-512

[CLMPR]: proposes CSIDH-512 for NIST level I

- Here the finite field is $\mathbb{F}_p$ with

$$p = 4 \cdot \ell_1 \cdots \ell_{74} - 1,$$

  where $\ell_1, \ldots, \ell_{74}$ are small distinct primes.
- Note that each ℓ_i divides $p + 1$.
- For an error rate of $< 2^{-32}$, our best algorithm requires

$$765325228976 \approx 0.7 \cdot 2^{40}$$

  nonlinear bit operations. Previous record was 2^{51}.
- Generic conversion gives $\approx 2^{43.3}$ T-gates using 2^{40} qubits.
- Can do $\approx 2^{45.3}$ T-gates using $\approx 2^{20}$ qubits.
- Total gates for one query (T+Clifford): $\approx 2^{46.9}$.
- Number of queries: $\approx 2^{19.3}$ using $\approx 2^{32}$ bits of QRACM [P].

Oracle errors

- [BLMP] gives oracle costs for error rates 2^{-1}, 2^{-32}, 2^{-256}.

Oracle errors

- [BLMP] gives oracle costs for error rates $2^{-1}, 2^{-32}, 2^{-256}$.
- Understanding the error tolerance of Kuperberg's algorithm is essential to obtain accurate concrete numbers.

Oracle errors

- [BLMP] gives oracle costs for error rates $2^{-1}, 2^{-32}, 2^{-256}$.
- Understanding the error tolerance of Kuperberg's algorithm is essential to obtain accurate concrete numbers.
- Advances in quantum error correction would also massively change the complexity.

Open questions: summary

- How do oracle errors interact with Kuperberg's algorithm?
- What kind of overheads come from handling large numbers of qubits?
- Is there a quantum algorithm that does better than L(1/2)?
  - Should be difficult: this would also decrease the security of all lattice proposals.
- Can we decrease the cost of one query?

Open questions: summary

- How do oracle errors interact with Kuperberg's algorithm?
- What kind of overheads come from handling large numbers of qubits?
- Is there a quantum algorithm that does better than $L(1/2)$?
  - Should be difficult: this would also decrease the security of all lattice proposals.
- Can we decrease the cost of one query?

Thank you!

References

[BLMP] Bernstein, Lange, Martindale, and Panny,
Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies,
Eurocrypt 2019, `quantum.isogeny.org`.

[CLMPR] Castryck, Lange, Martindale, Panny, and Renes,
CSIDH: An efficient post-quantum commutative group action,
Asiacrypt 2018, `csidh.isogeny.org`.

[CJS] Childs, Jao, and Soukharev,
Constructing elliptic curve isogenies in quantum subexponential time,
J. Math. Crypto 2014, `arxiv.org/abs/1012.4019`.

[P] Peikert,
He gives C-sieves on the CSIDH,
Eurocrypt 2020, `ia.cr.org/2019/725`.

Credits to my coauthors Daniel J. Bernstein, Tanja Lange, and Lorenz Panny for many of the contents of this presentation.