A note on the security of CSIDH

J.-F. Biasse [1], A. Iezzi [1], M. J. Jacobson Jr.[2]

[1]University of South Florida
[2]University of Calgary

December 11, 2018

Shor's algorithm

P. Shor

In 1994, Shor described a quantum polynomial time algorithm to factor RSA integers.

Shor's algorithm

P. Shor

In 1994, Shor described a quantum polynomial time algorithm to factor RSA integers.

This algorithm extends to the resolution of the Discrete Logarithm Problem in all finite groups.

Shor's algorithm

P. Shor

In 1994, Shor described a quantum polynomial time algorithm to factor RSA integers.

This algorithm extends to the resolution of the Discrete Logarithm Problem in all finite groups.

All the currently deployed public key infrastructure will need to be replaced.

How serious is the threat?

Common question: is this a serious threat or some theoretical nonsense?

How serious is the threat?

Common question: is this a serious threat or some theoretical nonsense?

"at present,... I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031." (Mosca, 2015)

How serious is the threat?

Common question: is this a serious threat or some theoretical nonsense?

"at present,... I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031." (Mosca, 2015)

In August 2015, NSA announced a future update of its Suite B of cryptographic protocols to account for the quantum threat.

How serious is the threat?

Common question: is this a serious threat or some theoretical nonsense?

"at present,... I estimate a 1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031." (Mosca, 2015)

In August 2015, NSA announced a future update of its Suite B of cryptographic protocols to account for the quantum threat.

In November 30th 2017, NIST initiated a standardization process for PQ primitives.

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

What are the quantum-safe alternatives?

- (Ideal) lattice-based crypto
- Code-based crypto
- Multivariate crypto
- Group-based crypto
- Isogeny-based crypto

What are the quantum-safe alternatives?

- (Ideal) lattice-based crypto
- Code-based crypto
- Multivariate crypto
- Group-based crypto
- Isogeny-based crypto

Little game

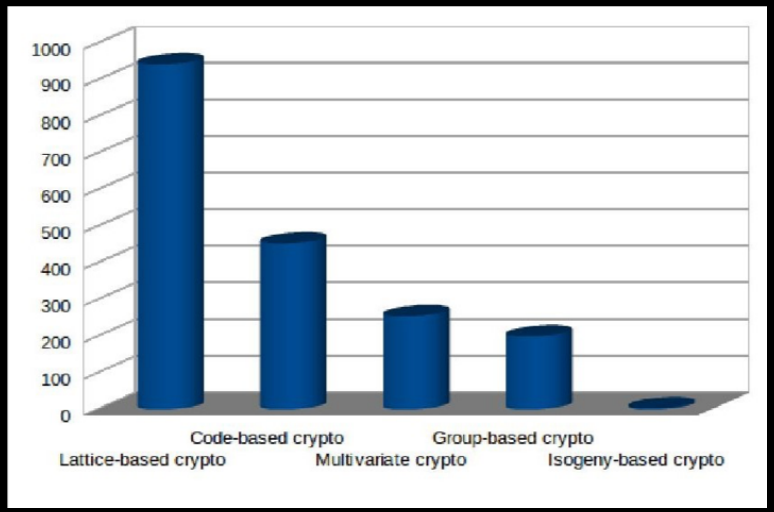
Google scholar "x-based cryptography" for $x =$ lattice, ..., isogeny.

What are the quantum-safe alternatives?

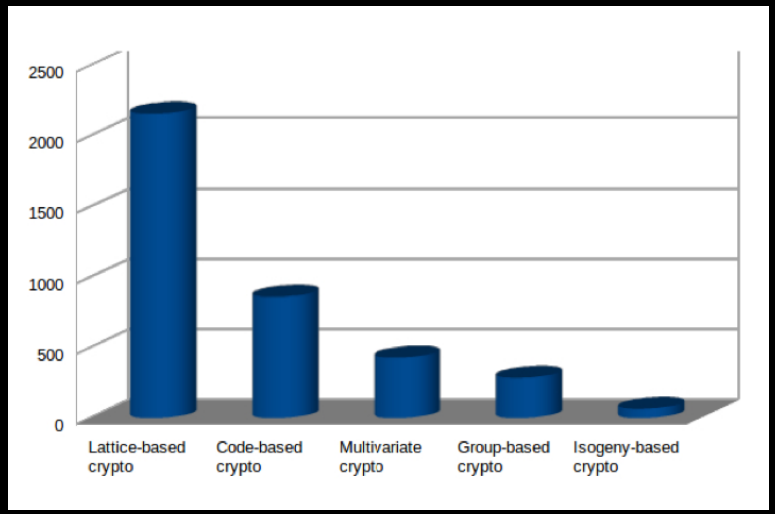
- (Ideal) lattice-based crypto
- Code-based crypto
- Multivariate crypto
- Group-based crypto
- Isogeny-based crypto

Little game

Google scholar "x-based cryptography" for $x =$ lattice, . . . , isogeny.



YYZ: 02/01/2015



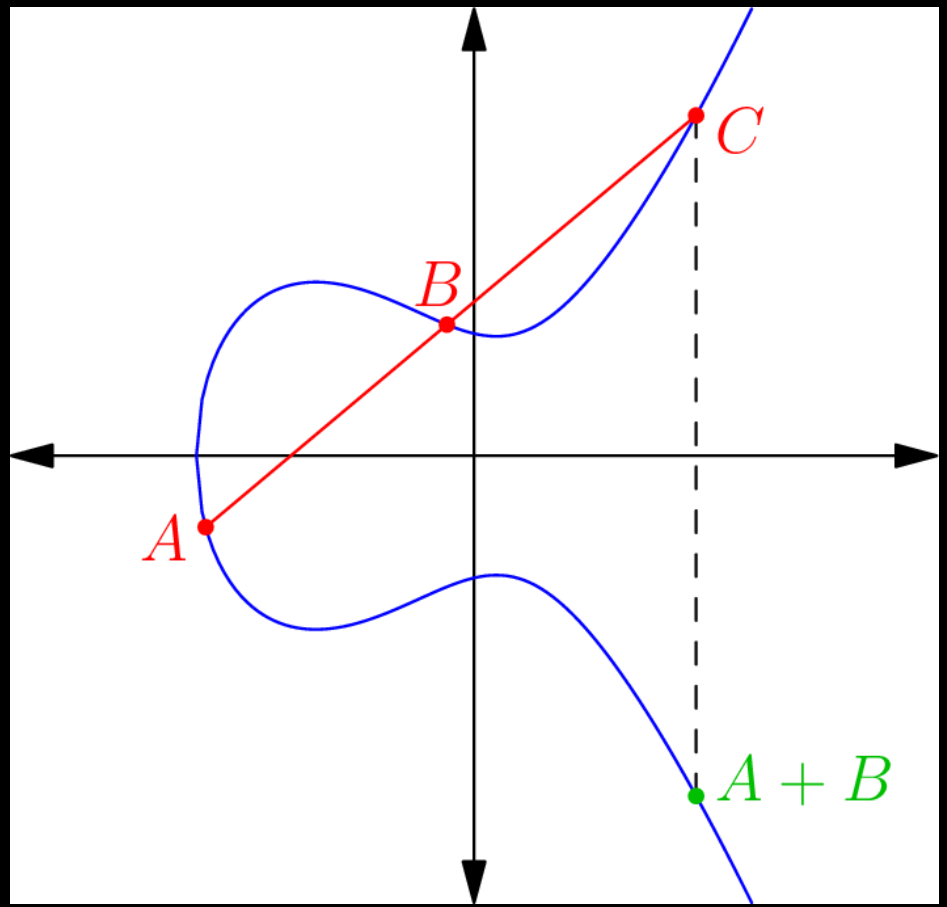
YYZ: 01/20/2018

Elliptic curves and isogenies

$$E : y^2 = x^3 + ax + b, \quad 4a^3 + 27b^2 \neq 0$$

Elliptic curves and isogenies

$$E : y^2 = x^3 + ax + b, \quad 4a^3 + 27b^2 \neq 0$$



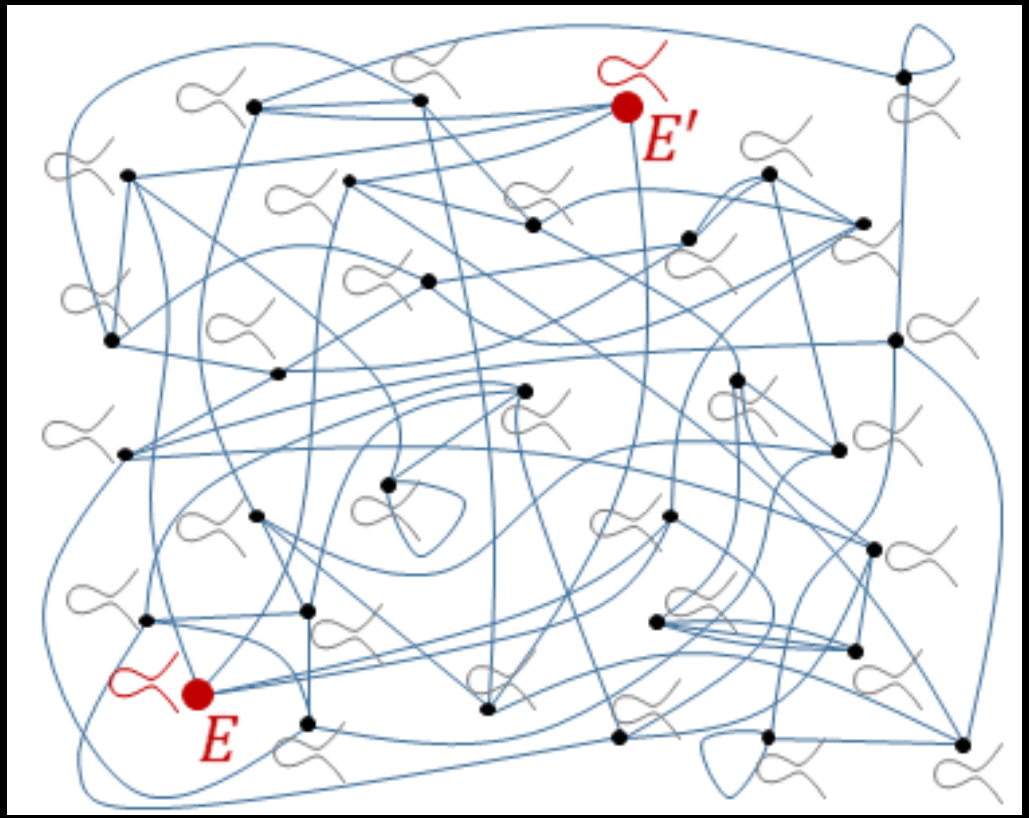
Group law

Elliptic curves and isogenies

$$\begin{array}{rccc} \varphi : & E & \longrightarrow & E' \\ & (x,y) & \mapsto & \left(\frac{f_1(x,y)}{g_1(x,y)}, \frac{f_2(x,y)}{g_2(x,y)}\right) \end{array}$$

Elliptic curves and isogenies

$$\begin{array}{rccc} \varphi : & E & \longrightarrow & E' \\ & (x,y) & \mapsto & \left(\frac{f_1(x,y)}{g_1(x,y)}, \frac{f_2(x,y)}{g_2(x,y)}\right) \end{array}$$



Isogeny graph (source: W. Castryck)

A commutative group action

A commutative group action

For E ordinary or supersingular over $\mathbb{F}_p$.

Let $\mathcal{O} = \mathrm{End}(E)$ be the ring of endomorphisms of E.

A commutative group action

For E ordinary or supersingular over $\mathbb{F}_p$.

Let $\mathcal{O} = \mathrm{End}(E)$ be the ring of endomorphisms of E.

- **Set**: isomorphism classes of elliptic curves having the same endomorphism ring $\mathcal{O}$.

A commutative group action

For E ordinary or supersingular over $\mathbb{F}_p$.

Let $\mathcal{O} = \mathrm{End}(E)$ be the ring of endomorphisms of E.

- **Set**: isomorphism classes of elliptic curves having the same endomorphism ring $\mathcal{O}$.
- **Group**: $G = \mathrm{Cl}(\mathcal{O})$,

$$G = \{[\mathfrak{a}] : \mathfrak{a} \text{ is an ideal of } \mathcal{O}\},$$

where $[\mathfrak{a}] = [\mathfrak{b}] \Leftrightarrow \exists\, 0 \neq \alpha \in \mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q}$ such that $\mathfrak{a} = (\alpha)\mathfrak{b}$.

G is a finite abelian group, $N := \#G$.

A commutative group action

For E ordinary or supersingular over $\mathbb{F}_p$.

Let $\mathcal{O} = \mathrm{End}(E)$ be the ring of endomorphisms of E.

- **Set**: isomorphism classes of elliptic curves having the same endomorphism ring $\mathcal{O}$.
- **Group**: $G = \mathrm{Cl}(\mathcal{O})$,

$$G = \{[\mathfrak{a}] : \mathfrak{a} \text{ is an ideal of } \mathcal{O}\},$$

where $[\mathfrak{a}] = [\mathfrak{b}] \Leftrightarrow \exists\, 0 \neq \alpha \in \mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q}$ such that $\mathfrak{a} = (\alpha)\mathfrak{b}$.

G is a finite abelian group, $N := \#G$.

Group action:

$$[\mathfrak{a}] * \overline{E} = \overline{E}'$$

A commutative group action

For E ordinary or supersingular over $\mathbb{F}_p$.

Let $\mathcal{O} = \mathrm{End}(E)$ be the ring of endomorphisms of E.

- **Set**: isomorphism classes of elliptic curves having the same endomorphism ring $\mathcal{O}$.
- **Group**: $G = \mathrm{Cl}(\mathcal{O})$,

$$G = \{[\mathfrak{a}] : \mathfrak{a} \text{ is an ideal of } \mathcal{O}\},$$

where $[\mathfrak{a}] = [\mathfrak{b}] \Leftrightarrow \exists\, 0 \neq \alpha \in \mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q}$ such that $\mathfrak{a} = (\alpha)\mathfrak{b}$.

G is a finite abelian group, $N := \#G$.

Group action:

$$[\mathfrak{a}] * \overline{E} = \overline{E}'$$

$$\varphi_{\mathfrak{a}} : E \to E'$$

with $\deg(\varphi_{\mathfrak{a}}) = \mathcal{N}(\mathfrak{a})$

CSIDH key exchange [CLMPR18] ([DKS18],[S09],[C97])

CSIDH key exchange [CLMPR18] ([DKS18],[S09],[C97])

Public value: E

CSIDH key exchange [CLMPR18] ([DKS18],[S09],[C97])

Public value: E

- Chooses secret $\mathfrak{a}$
- Broadcasts $[\mathfrak{a}] * \overline{E}$
- Computes:

$$[\mathfrak{a}] * \left([\mathfrak{b}] * \overline{E}\right) = [\mathfrak{ab}] * \overline{E}$$

- Chooses secret $\mathfrak{b}$
- Broadcasts $[\mathfrak{b}] * \overline{E}$
- Computes:

$$[\mathfrak{b}] * \left([\mathfrak{a}] * \overline{E}\right) = [\mathfrak{ab}] * \overline{E}$$

CSIDH key exchange [CLMPR18] ([DKS18],[S09],[C97])

Public value: E

- Chooses secret $\mathfrak{a}$
- Broadcasts $[\mathfrak{a}] * \overline{E}$
- Computes:

$$[\mathfrak{a}] * \left([\mathfrak{b}] * \overline{E}\right) = [\mathfrak{ab}] * \overline{E}$$

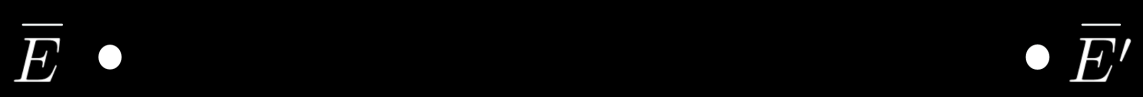
- Chooses secret $\mathfrak{b}$
- Broadcasts $[\mathfrak{b}] * \overline{E}$
- Computes:

$$[\mathfrak{b}] * \left([\mathfrak{a}] * \overline{E}\right) = [\mathfrak{ab}] * \overline{E}$$

Problem: Given $[\mathfrak{a}] * \overline{E}$ find $[\mathfrak{a}]$.

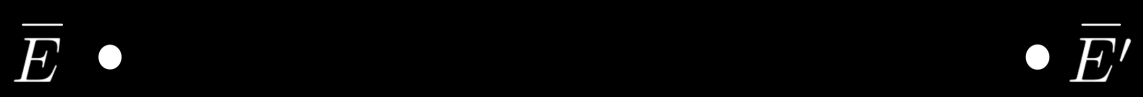
Traveling the isogeny graph

Suppose that $G = \langle [\mathfrak{p}_1], [\mathfrak{p}_2], [\mathfrak{p}_3] \rangle$



Traveling the isogeny graph

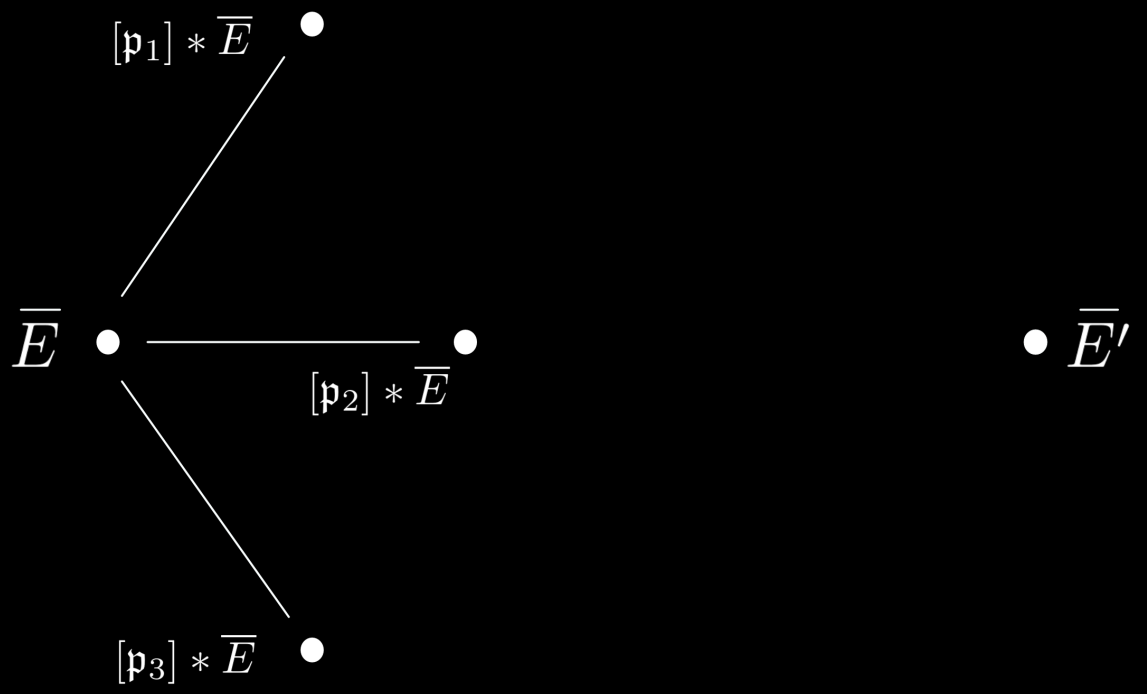
Suppose that $G = \langle [\mathfrak{p}_1], [\mathfrak{p}_2], [\mathfrak{p}_3] \rangle$



We want to find (short) (e_1, e_2, e_3) such that $[\mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\mathfrak{p}_3^{e_3}] * \overline{E} = \overline{E'}$.

Traveling the isogeny graph

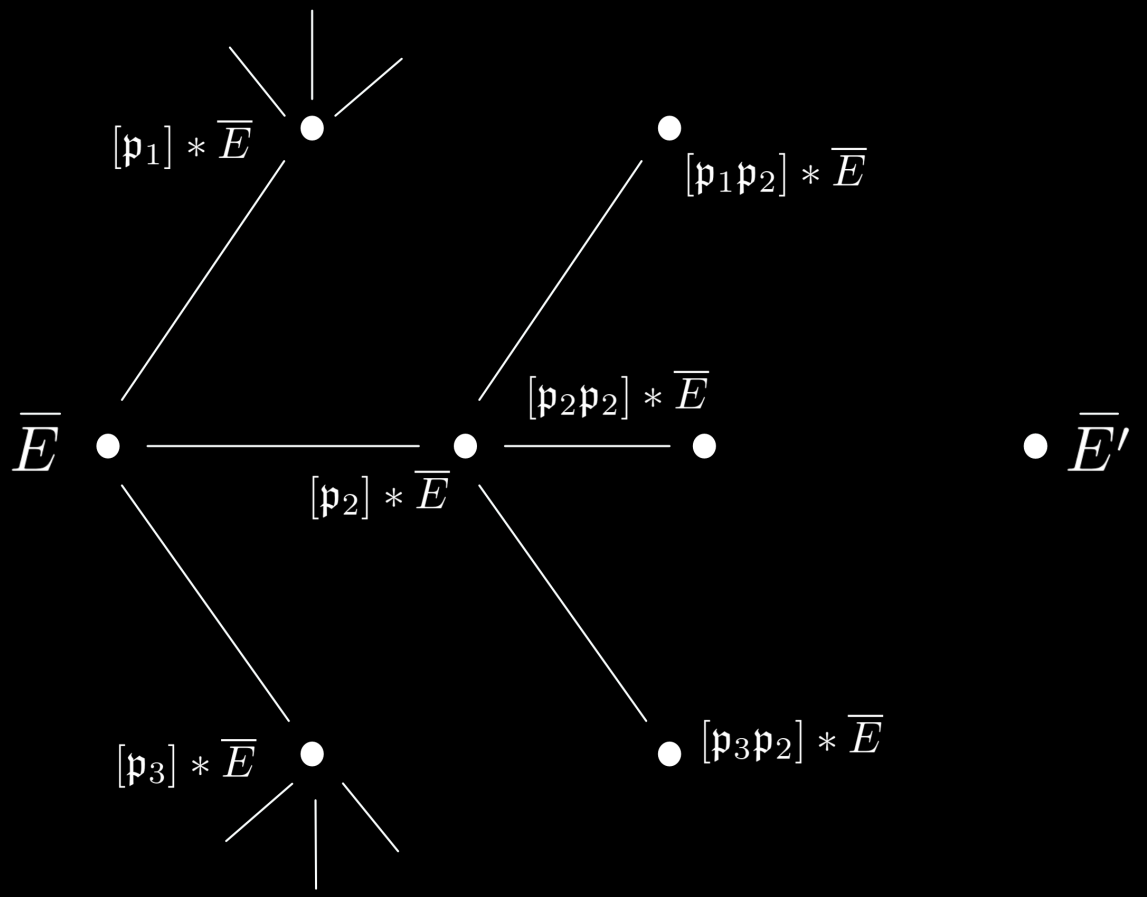
Suppose that $G = \langle [\mathfrak{p}_1], [\mathfrak{p}_2], [\mathfrak{p}_3] \rangle$



We want to find (short) (e_1, e_2, e_3) such that $[\mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\mathfrak{p}_3^{e_3}] * \overline{E} = \overline{E'}$.

Traveling the isogeny graph

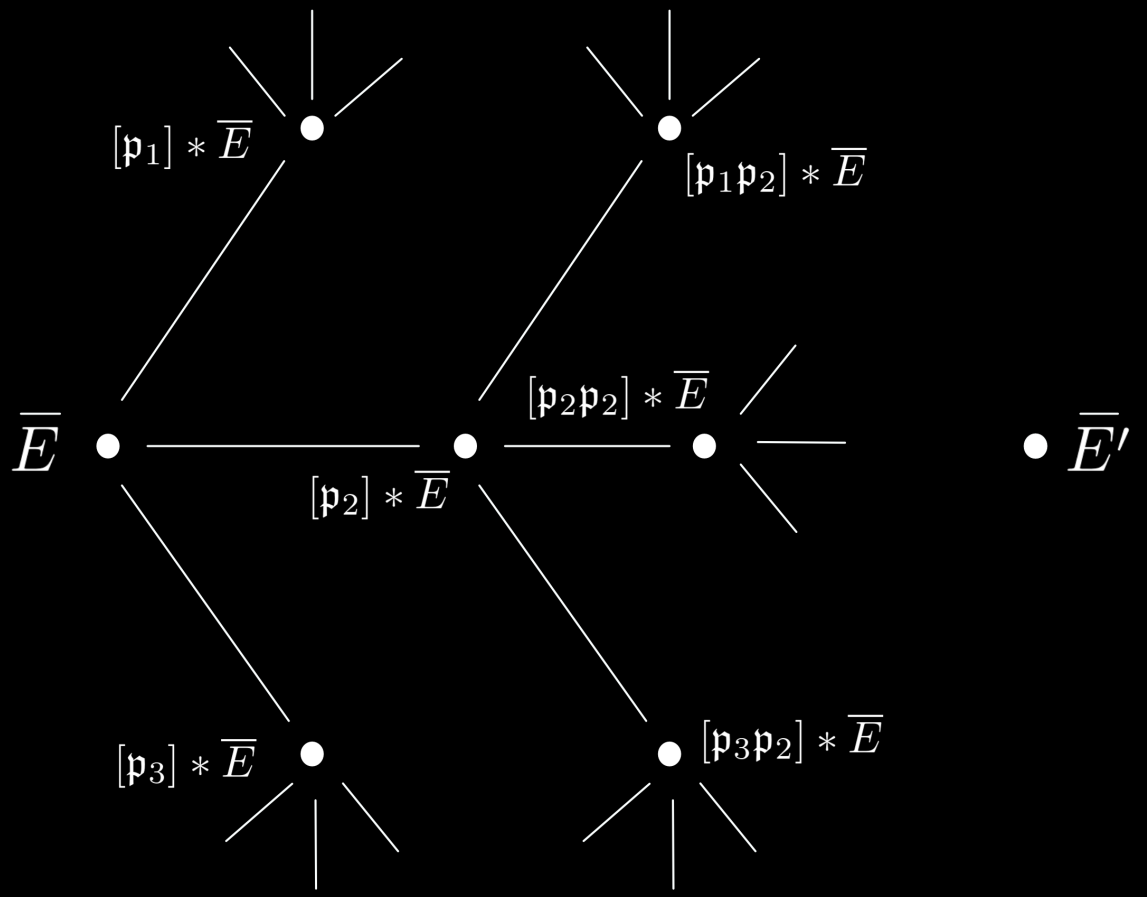
Suppose that $G = \langle [\mathfrak{p}_1], [\mathfrak{p}_2], [\mathfrak{p}_3] \rangle$



We want to find (short) (e_1, e_2, e_3) such that $[\mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\mathfrak{p}_3^{e_3}] * \overline{E} = \overline{E'}$.

Traveling the isogeny graph

Suppose that $G = \langle [\mathfrak{p}_1], [\mathfrak{p}_2], [\mathfrak{p}_3] \rangle$



We want to find (short) (e_1, e_2, e_3) such that $[\mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\mathfrak{p}_3^{e_3}] * \overline{E} = \overline{E'}$.

Hidden Shift Problem

Given curves E, E' we want to find $\mathfrak{s}$ such that $[\mathfrak{s}] * \overline{E} = \overline{E}'$.

This can be rephrased as follows.

Let

$$\begin{bmatrix} f_1 : [\mathfrak{a}] \in G \mapsto [\mathfrak{a}] * \overline{E} \\ f_2 : [\mathfrak{a}] \in G \mapsto [\mathfrak{a}] * \overline{E}'. \end{bmatrix}$$

Hidden Shift Problem

Given curves E, E' we want to find $\mathfrak{s}$ such that $[\mathfrak{s}] * \overline{E} = \overline{E}'$.

This can be rephrased as follows.

Let

$$\begin{cases} f_1 : [\mathfrak{a}] \in G \mapsto [\mathfrak{a}] * \overline{E} \\ f_2 : [\mathfrak{a}] \in G \mapsto [\mathfrak{a}] * \overline{E}'. \end{cases}$$

If we find $\mathfrak{s}$ such that

$$\text{for all } [\mathfrak{a}] \in G,\ f_2([\mathfrak{a}]) = f_1([\mathfrak{a}\mathfrak{s}]),$$

Hidden Shift Problem

Given curves E, E' we want to find $\mathfrak{s}$ such that $[\mathfrak{s}] * \overline{E} = \overline{E}'$.

This can be rephrased as follows.

Let

$$\left[\begin{array}{l} f_1 : [\mathfrak{a}] \in G \mapsto [\mathfrak{a}] * \overline{E} \\ f_2 : [\mathfrak{a}] \in G \mapsto [\mathfrak{a}] * \overline{E}'. \end{array}\right.$$

If we find $\mathfrak{s}$ such that

$$\text{for all } [\mathfrak{a}] \in G,\ f_2([\mathfrak{a}]) = f_1([\mathfrak{a}\mathfrak{s}]),$$

then

$$[\mathfrak{s}] * \overline{E} = \overline{E}'.$$

Kuperberg's sieve

Finding $[\mathfrak{s}]$ boils down to finding the periods H of an oracle f defined over $\frac{\mathbb{Z}}{2\mathbb{Z}} \ltimes G$ by:

$$f(x,[\mathfrak{a}]) := \begin{cases} |[\mathfrak{a}] * \overline{E}\rangle = |f_1([\mathfrak{a}])\rangle, & x = 0 \\ |[\mathfrak{a}]^{-1} * \overline{E}'\rangle = |f_2([\mathfrak{a}]^{-1})\rangle, & x = 1 \end{cases}$$

Kuperberg's sieve

Finding $[\mathfrak{s}]$ boils down to finding the periods H of an oracle f defined over $\frac{\mathbb{Z}}{2\mathbb{Z}} \ltimes G$ by:

$$f(x,[\mathfrak{a}]) := \begin{cases} |[\mathfrak{a}] * \overline{E}\rangle = |f_1([\mathfrak{a}])\rangle, & x = 0 \\ |[\mathfrak{a}]^{-1} * \overline{E}'\rangle = |f_2([\mathfrak{a}]^{-1})\rangle, & x = 1 \end{cases}$$

Kuperberg's sieve finds H by using:

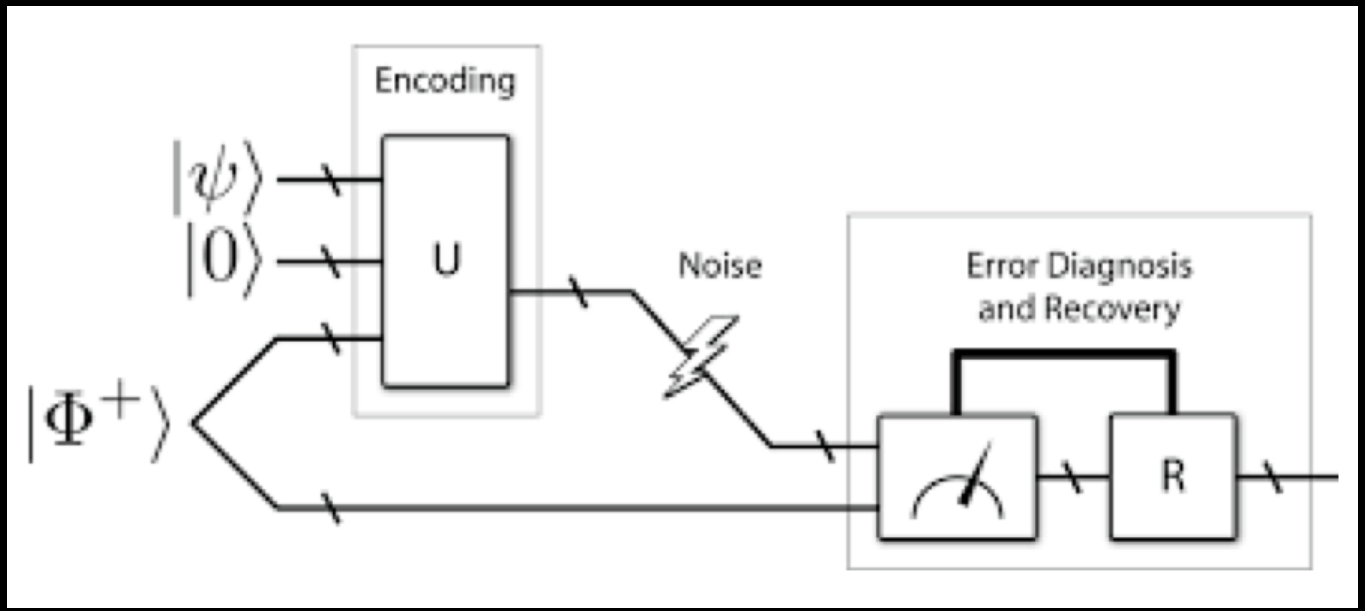
- $2^{O\left(\sqrt{\log(N)}\right)}$ calls to f and $2^{O\left(\sqrt{\log(N)}\right)}$ quantum memory.
- $2^{O\left(\sqrt{\log(N)\log\log(N)}\right)}$ calls to f and polynomial quantum memory.

On the importance of quantum memory

Everyone agrees:

Quantum memory won't be cheap.

It will take a long time to build a quantum computer handling a large amount of memory.



Lattice techniques to find a short product

Goal: given $\mathfrak{a}$ and E compute $[\mathfrak{a}] * \overline{E}$

Lattice techniques to find a short product

Goal: given $\mathfrak{a}$ and E compute $[\mathfrak{a}] * \overline{E}$

Let $[\mathfrak{p}_1], [\mathfrak{p}_2], \ldots, [\mathfrak{p}_k]$ be generators of G.

Lattice techniques to find a short product

Goal: given $\mathfrak{a}$ and E compute $[\mathfrak{a}] * \overline{E}$

Let $[\mathfrak{p}_1], [\mathfrak{p}_2], \ldots, [\mathfrak{p}_k]$ be generators of G.

Let $\vec{x} = (x_1, \ldots, x_k) \in \mathbb{Z}^k$ such that $[\mathfrak{a}] = [\mathfrak{p}_1]^{x_1} [\mathfrak{p}_2]^{x_2} \cdots [\mathfrak{p}_k]^{x_k}$.

Lattice techniques to find a short product

Goal: given $\mathfrak{a}$ and E compute $[\mathfrak{a}] * \overline{E}$

Let $[\mathfrak{p}_1], [\mathfrak{p}_2], \ldots, [\mathfrak{p}_k]$ be generators of G.

Let $\vec{x} = (x_1, \ldots, x_k) \in \mathbb{Z}^k$ such that $[\mathfrak{a}] = [\mathfrak{p}_1]^{x_1}[\mathfrak{p}_2]^{x_2} \cdots [\mathfrak{p}_k]^{x_k}$.

Let $\mathcal{L} \subseteq \mathbb{Z}^k = \{\vec{e} = (e_1, \ldots, e_k) : [\mathfrak{p}_1]^{e_1}[\mathfrak{p}_2]^{e_2} \cdots [\mathfrak{p}_k]^{e_k} = 1_G\}$.

Lattice techniques to find a short product

Goal: given $\mathfrak{a}$ and E compute $[\mathfrak{a}] * \overline{E}$

Let $[\mathfrak{p}_1], [\mathfrak{p}_2], \ldots, [\mathfrak{p}_k]$ be generators of G.

Let $\vec{x} = (x_1, \ldots, x_k) \in \mathbb{Z}^k$ such that $[\mathfrak{a}] = [\mathfrak{p}_1]^{x_1}[\mathfrak{p}_2]^{x_2} \cdots [\mathfrak{p}_k]^{x_k}$.

Let $\mathcal{L} \subseteq \mathbb{Z}^k = \{\vec{e} = (e_1, \ldots, e_k) : [\mathfrak{p}_1]^{e_1}[\mathfrak{p}_2]^{e_2} \cdots [\mathfrak{p}_k]^{e_k} = 1_G\}$.

Let $\vec{y} = \vec{x} - \vec{e}$ for $\vec{e} \in \mathcal{L}$. Then $[\mathfrak{a}] = [\mathfrak{p}_1]^{y_1}[\mathfrak{p}_2]^{y_2} \cdots [\mathfrak{p}_k]^{y_k}$

Lattice techniques to find a short product

Goal: given $\mathfrak{a}$ and E compute $[\mathfrak{a}] * \overline{E}$

Let $[\mathfrak{p}_1], [\mathfrak{p}_2], \ldots, [\mathfrak{p}_k]$ be generators of G.

Let $\vec{x} = (x_1, \ldots, x_k) \in \mathbb{Z}^k$ such that $[\mathfrak{a}] = [\mathfrak{p}_1]^{x_1}[\mathfrak{p}_2]^{x_2} \cdots [\mathfrak{p}_k]^{x_k}$.

Let $\mathcal{L} \subseteq \mathbb{Z}^k = \{\vec{e} = (e_1, \ldots, e_k) : [\mathfrak{p}_1]^{e_1}[\mathfrak{p}_2]^{e_2} \cdots [\mathfrak{p}_k]^{e_k} = 1_G\}$.

Let $\vec{y} = \vec{x} - \vec{e}$ for $\vec{e} \in \mathcal{L}$. Then $[\mathfrak{a}] = [\mathfrak{p}_1]^{y_1}[\mathfrak{p}_2]^{y_2} \cdots [\mathfrak{p}_k]^{y_k}$

BDD: Find $\vec{e} \in \mathcal{L}$ such that $\|\vec{y}\| = \|\vec{x} - \vec{e}\|$ is small.

Performance of the oracle

We decompose the input $[\mathfrak{a}] = [\mathfrak{p}_1]^{y_1}[\mathfrak{p}_2]^{y_2}\cdots[\mathfrak{p}_k]^{y_k}$

- Cost of the calculation of $\mathcal{L}$: Polynomial [BS16]
- Cost of the decomposition of $[\mathfrak{a}]$: Polynomial [BS16]
- Cost of the BDD instance: Polynomial [Ba86]

Performance of the oracle

We decompose the input $[\mathfrak{a}] = [\mathfrak{p}_1]^{y_1}[\mathfrak{p}_2]^{y_2}\cdots[\mathfrak{p}_k]^{y_k}$

- Cost of the calculation of $\mathcal{L}$: Polynomial [BS16]
- Cost of the decomposition of $[\mathfrak{a}]$: Polynomial [BS16]
- Cost of the BDD instance: Polynomial [Ba86]

We iterate the calculation of the action $\overline{E} \leftarrow [\mathfrak{p}_i] * \overline{E}$.

- Cost of the evaluation of the $[\mathfrak{p}_i]$: $2^{\tilde{O}\left(\sqrt[3]{\log(N)}\right)}$

Performance of the oracle

We decompose the input $[\mathfrak{a}] = [\mathfrak{p}_1]^{y_1}[\mathfrak{p}_2]^{y_2}\cdots[\mathfrak{p}_k]^{y_k}$

- Cost of the calculation of $\mathcal{L}$: Polynomial [BS16]
- Cost of the decomposition of $[\mathfrak{a}]$: Polynomial [BS16]
- Cost of the BDD instance: Polynomial [Ba86]

We iterate the calculation of the action $\overline{E} \leftarrow [\mathfrak{p}_i] * \overline{E}$.

- Cost of the evaluation of the $[\mathfrak{p}_i]$: $2^{\tilde{O}\left(\sqrt[3]{\log(N)}\right)}$

All steps have polynomial classical and quantum memory requirement.

Conclusion and open problems

We have algorithms to compute $\mathfrak{s}$ such that $\overline{E}' = [\mathfrak{s}] * \overline{E}$.

Conclusion and open problems

We have algorithms to compute $\mathfrak{s}$ such that $\overline{E}' = [\mathfrak{s}] * \overline{E}$.

Time: $2^{O\left(\sqrt{\log(N)}\right)}$
Quantum memory: $2^{O\left(\sqrt{\log(N)}\right)}$

Time: $2^{O\left(\sqrt{\log(N)\log\log(N)}\right)}$
Quantum memory: Polynomial

Conclusion and open problems

We have algorithms to compute $\mathfrak{s}$ such that $\overline{E}' = [\mathfrak{s}] * \overline{E}$.

Time: $2^{O\left(\sqrt{\log(N)}\right)}$
Quantum memory: $2^{O\left(\sqrt{\log(N)}\right)}$

Time: $2^{O\left(\sqrt{\log(N)\log\log(N)}\right)}$
Quantum memory: Polynomial

Open problems

- Remove heuristics
- Concrete security estimates

ध्यान देने के लिए आपका धन्यवाद

Acknowledgment: research funded by