One-way functions and malleability oracles: Hidden shift attacks on isogeny-based protocols

Péter Kutas[1,2] **Simon-Philipp Merz**[3] Christophe Petit[1,4]
Charlotte Weitkämper[1]

[1]University of Birmingham, UK
[2]Eötvös Loránd University, Budapest, Hungary
[3]Royal Holloway, University of London, UK
[4]Université libre de Bruxelles, Belgium

October 2021

Eurocrypt 2021 - Zagreb, Croatia

Overview

- We find a *quantum* subexponential attack on (overstretched and unbalanced) SIDH via reduction to an injective abelian hidden shift problem.
- For ordinary curves and CSIDH: Childs-Jao-Soukharev give a quantum subexponential attack to compute an isogeny between two curves using this reduction.
- Previously widespread belief for SIDH: *"Since the algorithm of Childs et al. depends crucially on the properties of abelian groups, we believe that no reasonable variant of this strategy would apply to supersingular curves."* [DJP11]
- **Disclaimer:** The attack does not apply to balanced SIDH or SIKE parameters.

Overview

- We find a *quantum* subexponential attack on (overstretched and unbalanced) SIDH via reduction to an injective abelian hidden shift problem.
- For ordinary curves and CSIDH: Childs-Jao-Soukharev give a quantum subexponential attack to compute an isogeny between two curves using this reduction.
- Previously widespread belief for SIDH: *"Since the algorithm of Childs et al. depends crucially on the properties of abelian groups, we believe that no reasonable variant of this strategy would apply to supersingular curves."* [DJP11]
- **Disclaimer:** The attack does not apply to balanced SIDH or SIKE parameters.

Isogenies

Let E_0, E_1 be elliptic curves defined over a field K.

- An *isogeny* is a non-constant rational map $\varphi : E_0 \to E_1$ that is also a group homomorphism.
- (Separable) isogenies correspond to finite subgroups of E_0.
- The kernel of an isogeny determines the image curve up to isomorphism. $(E_0 / \ker(\varphi) := E_1)$
- Two curves E_0, E_1 are isomorphic if and only if they have the same j-invariant.

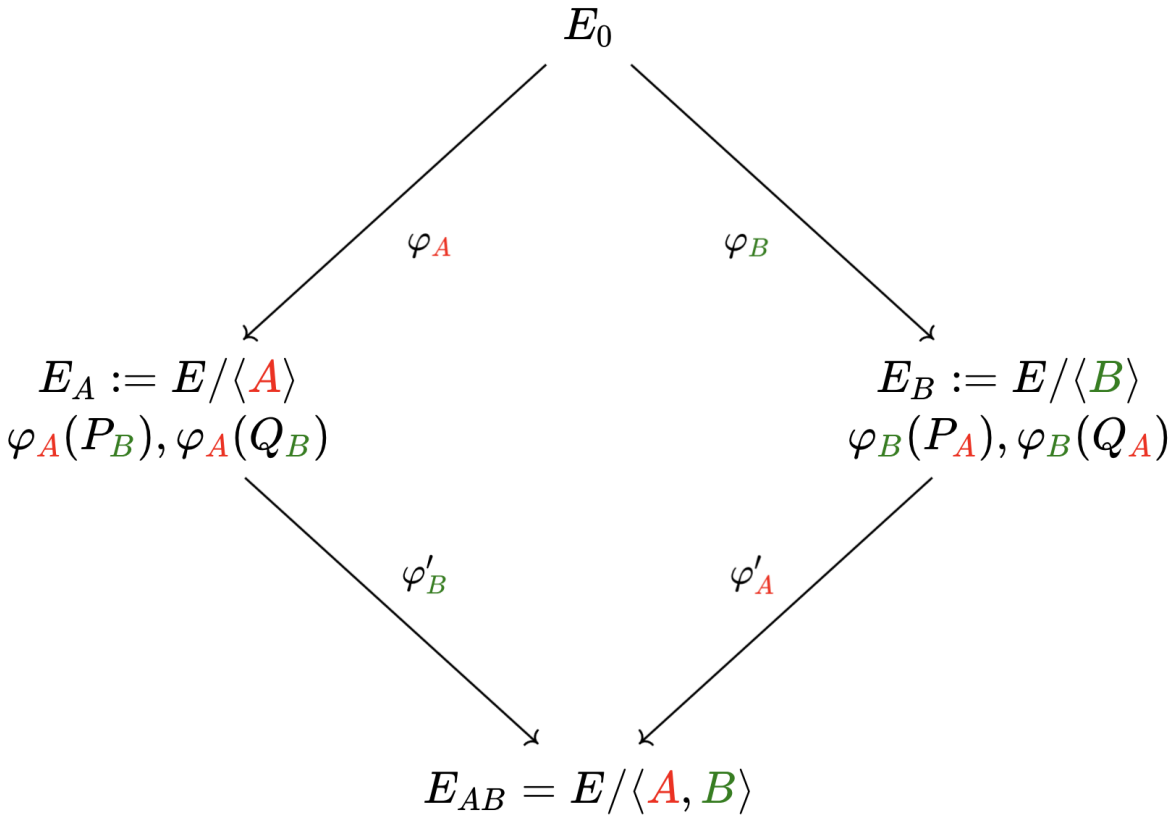
Isogeny-based cryptography

- Candidate for post-quantum cryptography.
- Based on the hardness of finding (large degree) isogenies between supersingular elliptic curves.
- Most isogeny-based cryptosystems (e.g. SIDH) are based on a relaxation of this problem.
- Small key sizes.

SIDH [JD11]

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$.

- Alice's secret is $A := P_A + [\mathsf{sk}_A] Q_A$.
- Bob's secret is $B := P_B + [\mathsf{sk}_B] Q_B$.



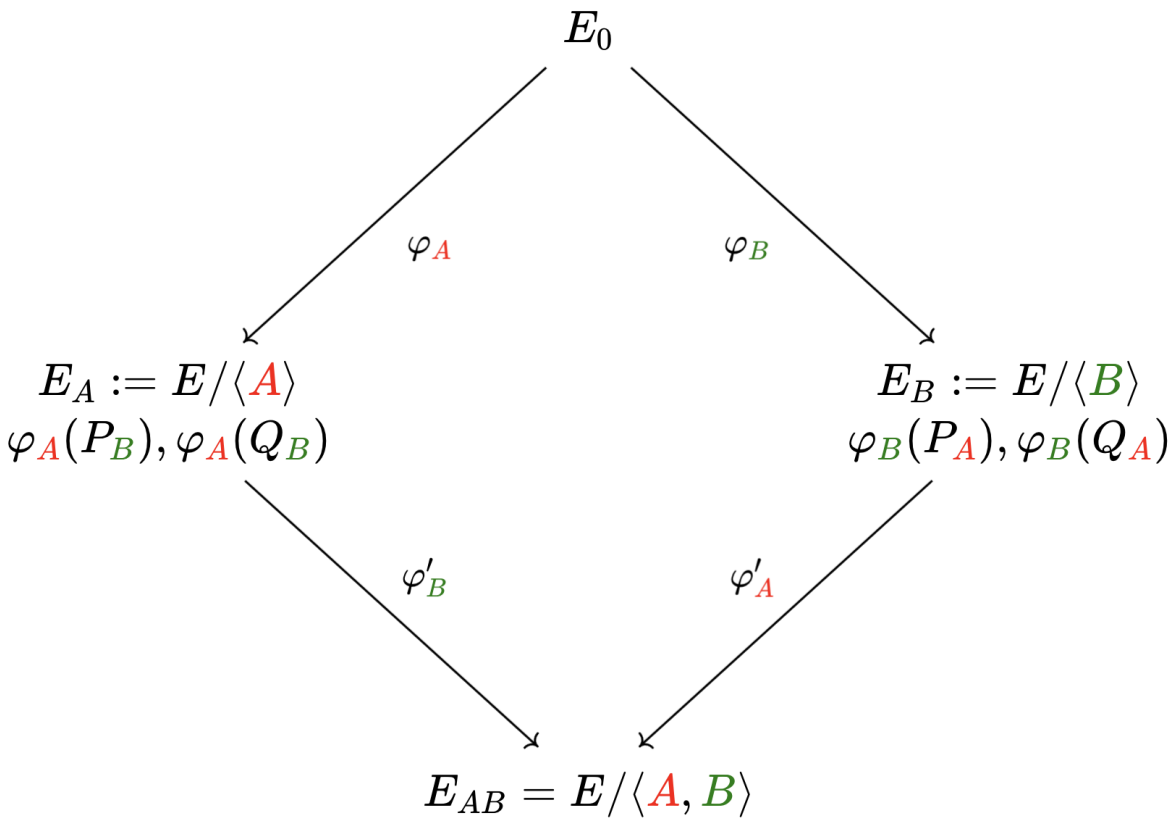
- Alice sends E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$.
- Bob sends E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$.

- The shared secret is the j-invariant of E_{AB}.

SIDH [JD11]

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$.

- Alice's secret is $A := P_A + [\mathsf{sk}_A] Q_A$.
- Bob's secret is $B := P_B + [\mathsf{sk}_B] Q_B$.



- Alice sends E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$.
- Bob sends E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$.

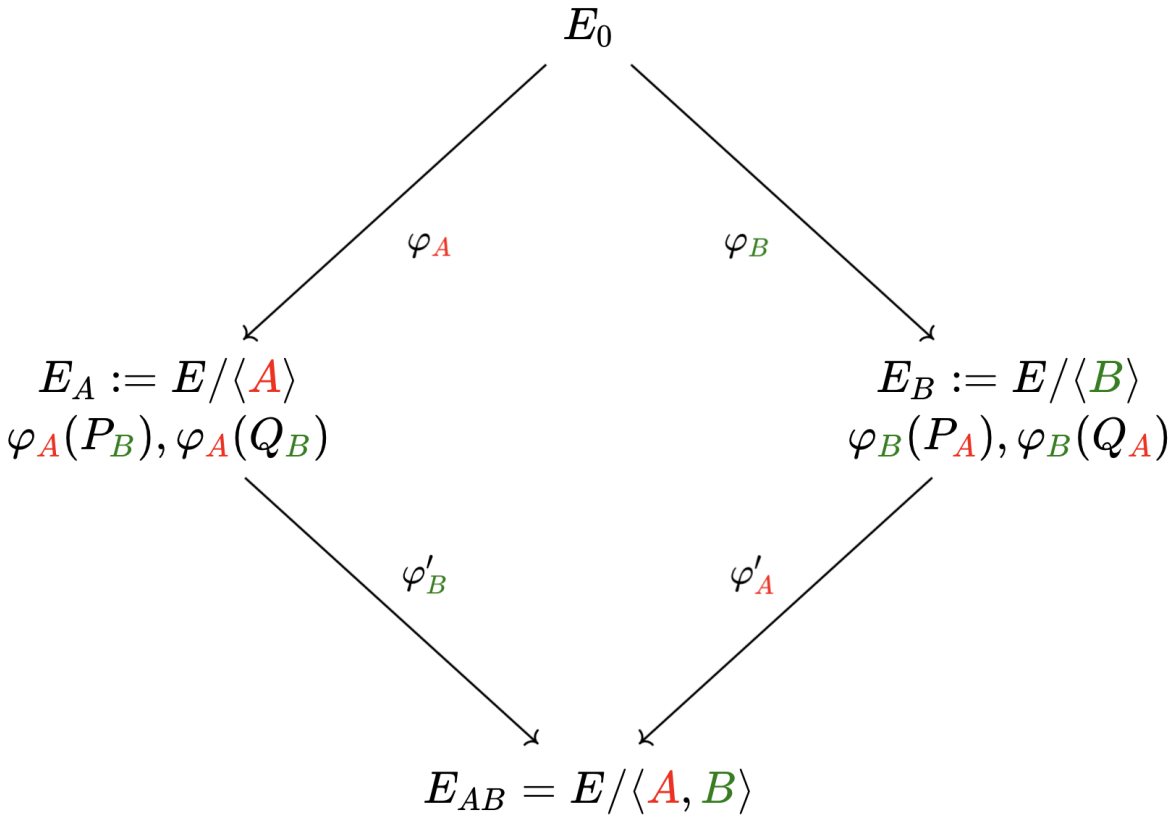
- The shared secret is the j-invariant of E_{AB}.

SIDH [JD11]

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$.

- Alice's secret is $A := P_A + [\mathsf{sk}_A] Q_A$.
- Bob's secret is $B := P_B + [\mathsf{sk}_B] Q_B$.

- Alice sends E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$.
- Bob sends E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$.



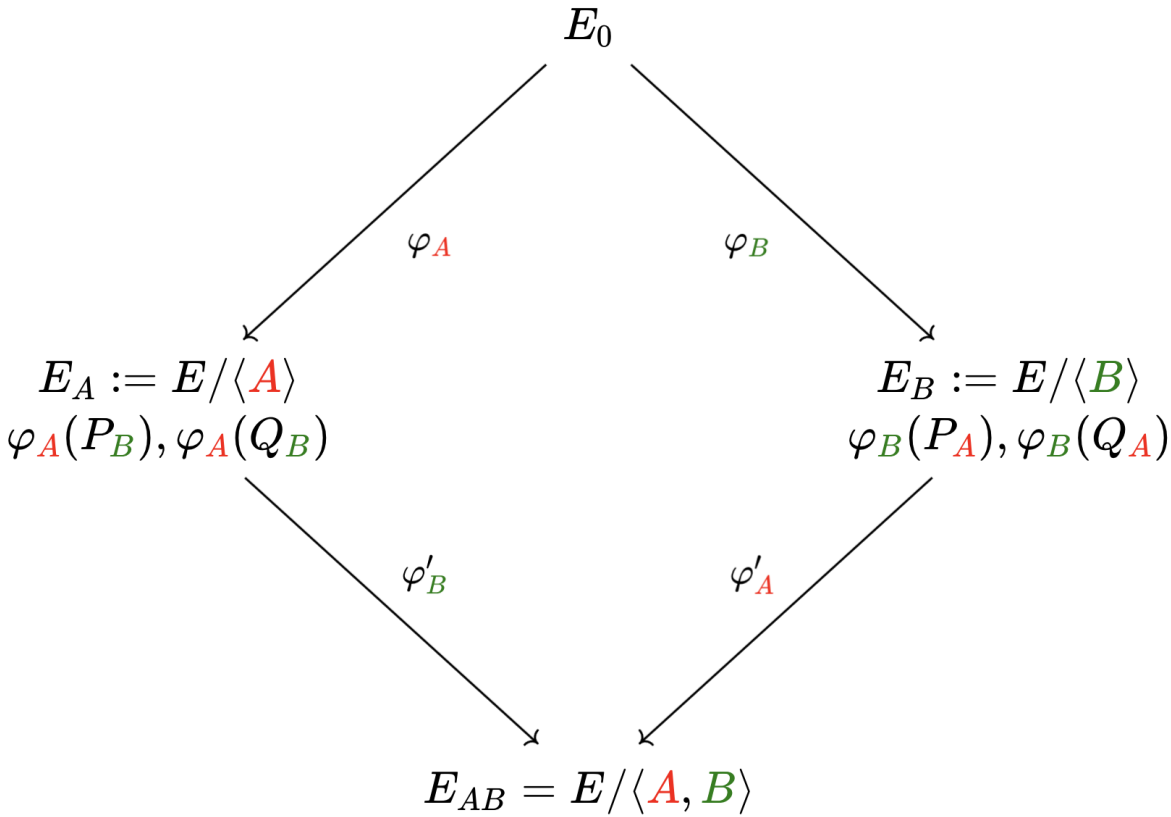
- The shared secret is the j-invariant of E_{AB}.

SIDH [JD11]

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$.

- Alice's secret is $A := P_A + [\mathsf{sk}_A] Q_A$.
- Bob's secret is $B := P_B + [\mathsf{sk}_B] Q_B$.

- Alice sends E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$.
- Bob sends E_B, $\phi_B(P_A)$, $\phi_B(Q_A)$.



- The shared secret is the j-invariant of E_{AB}.

The hidden shift problem

Definition (Hidden shift problem)

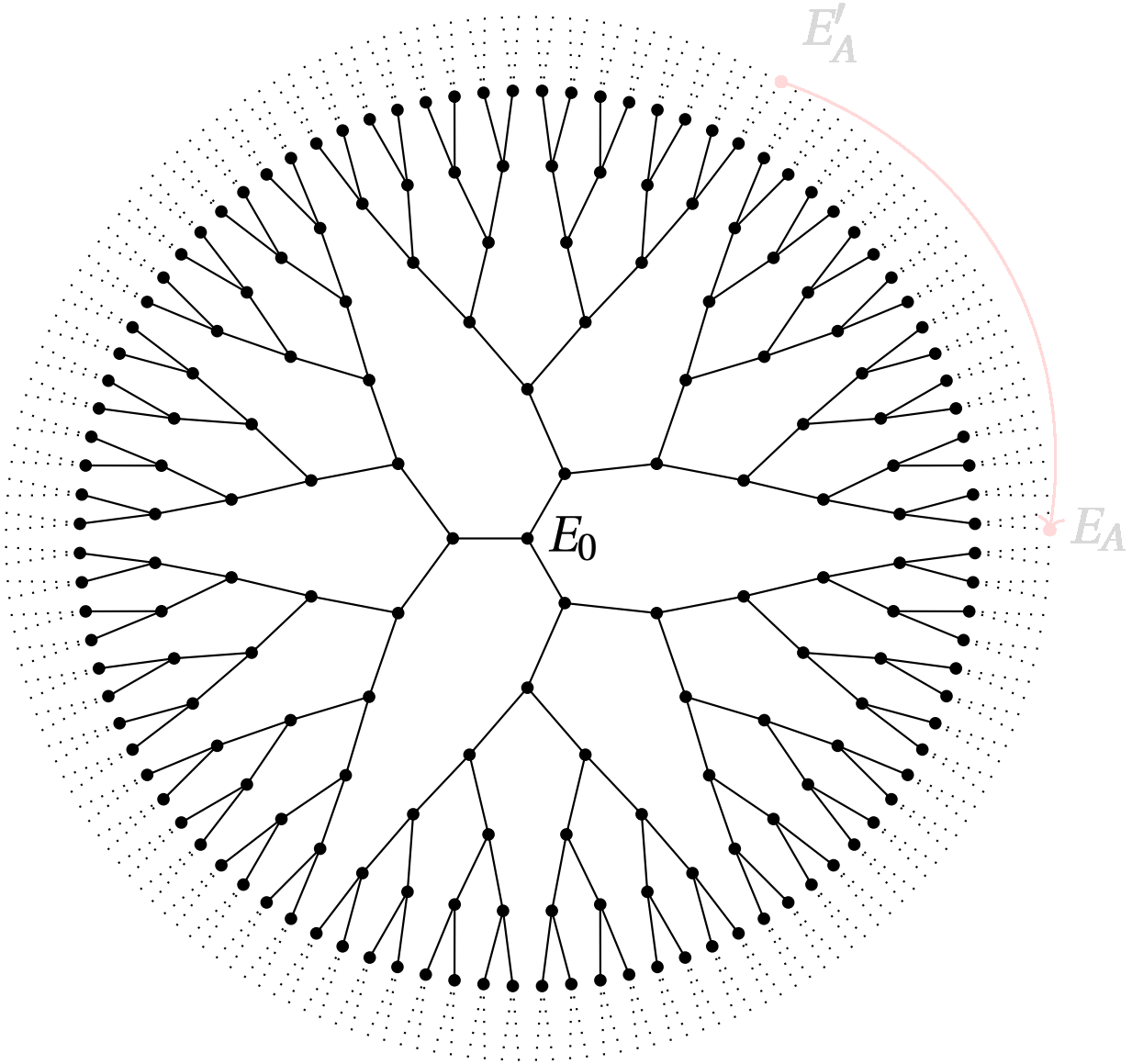
Let F_0, $F_1 : G \to X$ be two functions defined on some group G, such that there exists some $s \in G$ satisfying

$$F_0(g) = F_1(g \cdot s)$$

for all $g \in G$. The hidden shift problem is to find s given oracle access to the functions F_0 and F_1.

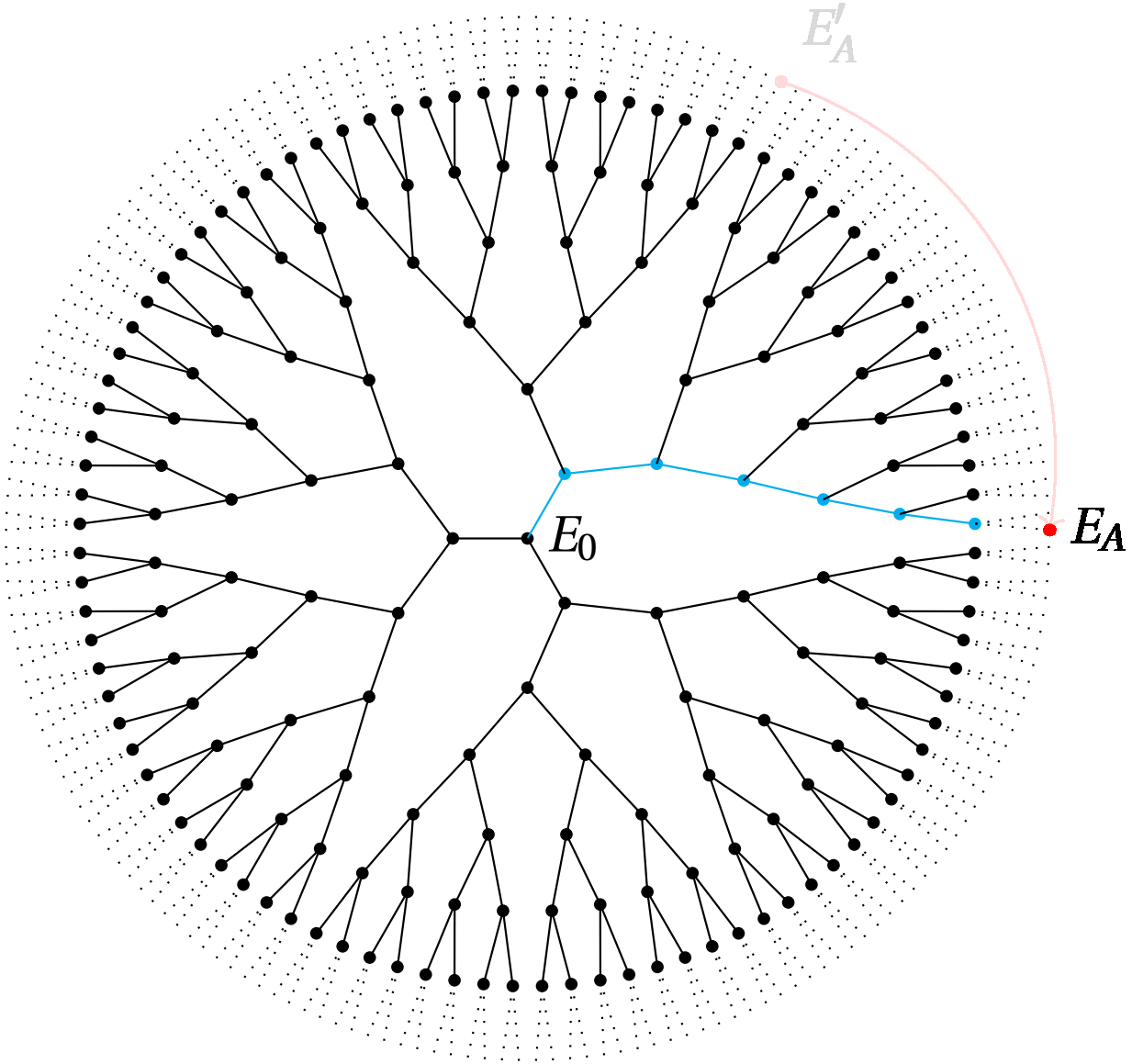
- If G is abelian and F_0, F_1 are injective, this can be solved in quantum subexponential time in $|G|$.

Intuition



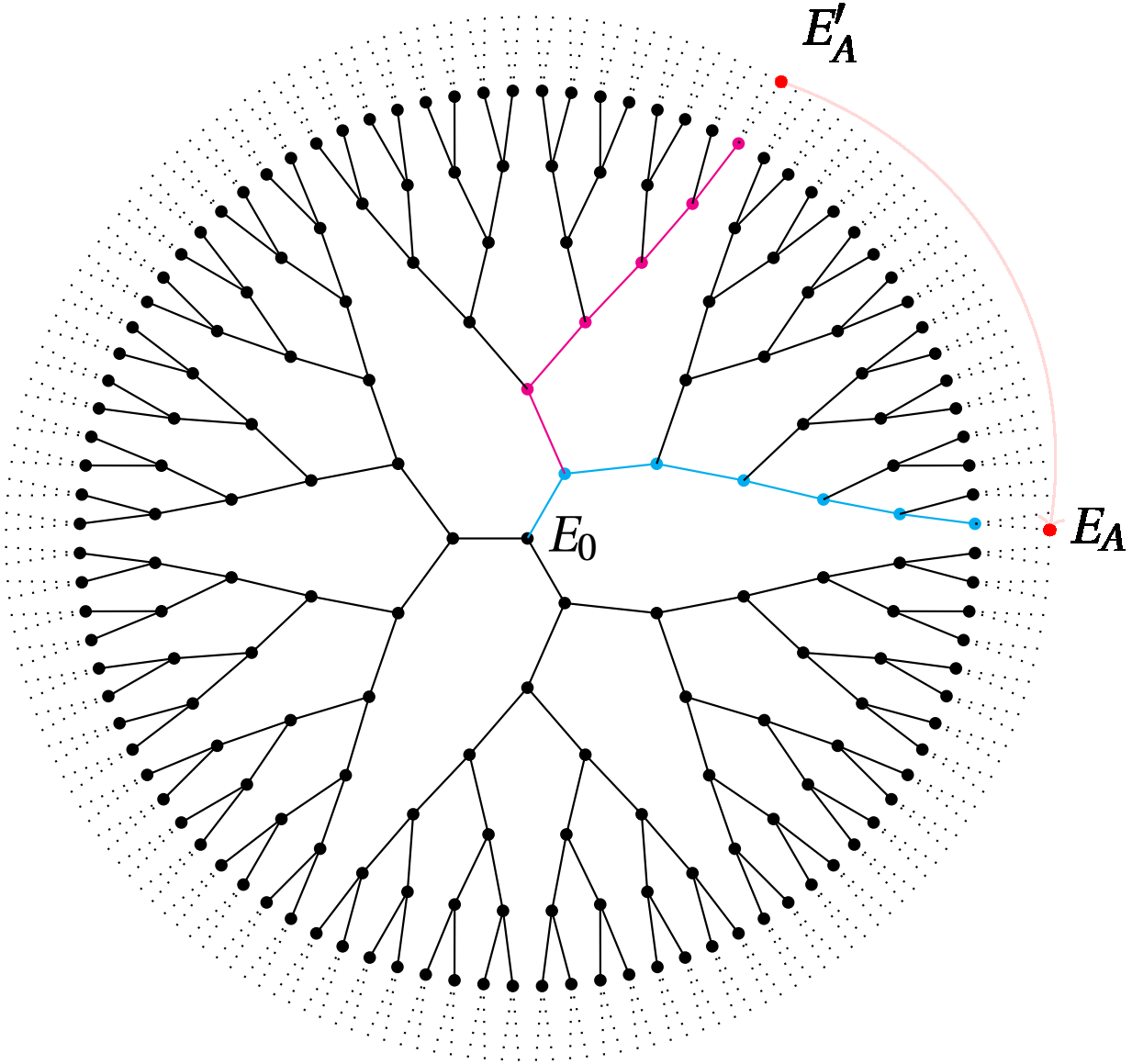
Thanks to Luca de Feo for the template.

Intuition



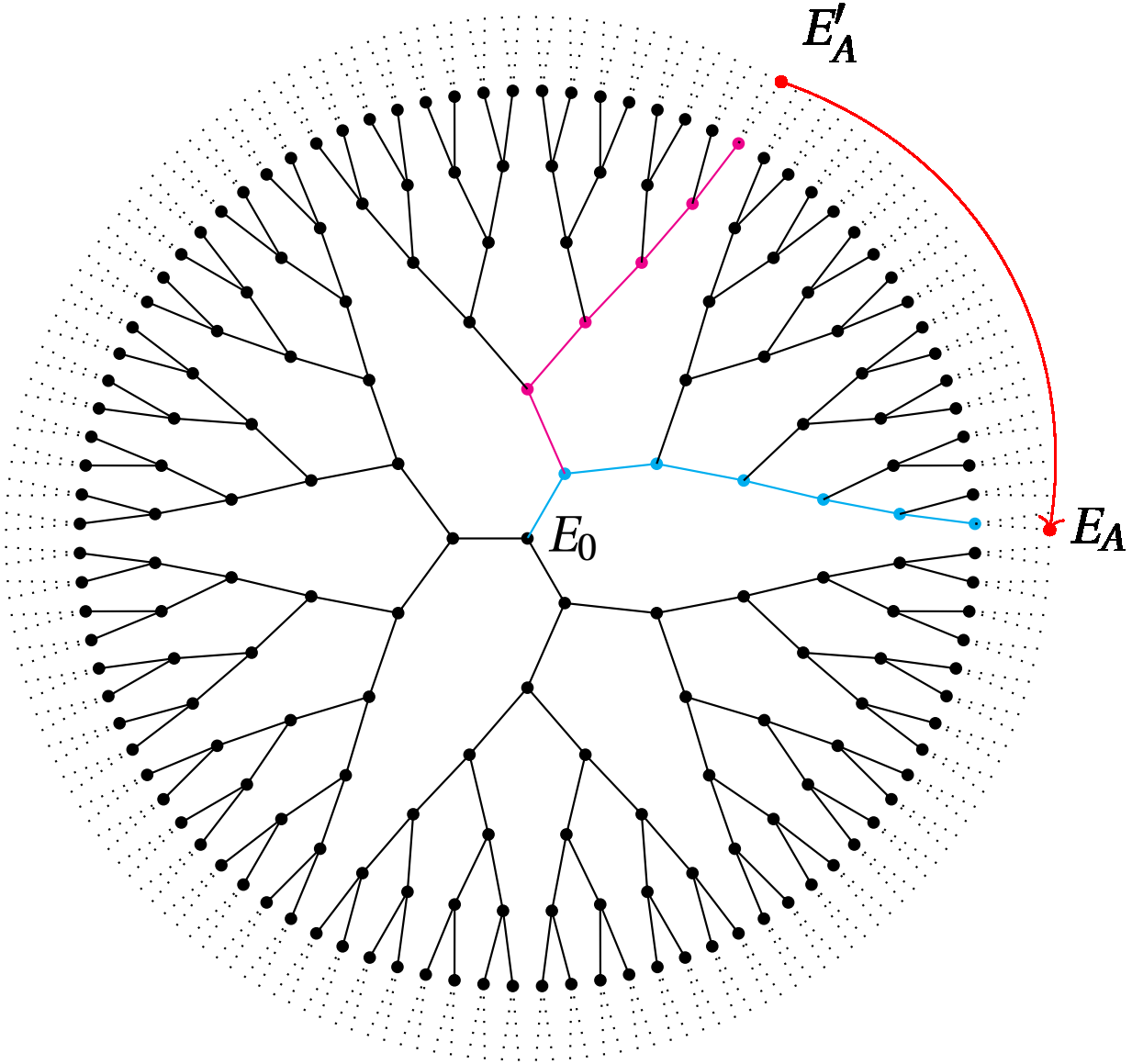
Thanks to Luca de Feo for the template.

Intuition



Thanks to Luca de Feo for the template.

Intuition



Thanks to Luca de Feo for the template.

Malleability Oracles

- Let $f : I \to O$ be an injective one-way function.
- Let G be a group acting on I.

Definition (Malleability oracles)

A *malleability oracle* for G at $o := f(i)$ provides $f(g \cdot i)$ for any $g \in G$, i.e., given any $g \in G$ and $f(i)$ the malleability oracle evaluates

$$g \mapsto f(g \cdot i).$$

Malleability oracles

Idea: Find preimage of $f(i)$ via a hidden shift computation.

Requirements:

1. Let G act transitively on I, and assume we have a malleability oracle for G at $f(i)$.
2. Let f be injective and let G be a finitely generated abelian group acting freely on I.

Strategy to compute i:

1. Pick any $j \in I$. (By transitivity of G, $\exists \sigma$ s.t. $i = \sigma \cdot j$.)
2. Define $F_0(g) := f(g \cdot j)$.
3. Define $F_1(g) := f(g \cdot i)$.
4. Compute σ such that $F_0(\sigma g) = F_1(g)$.
   (Injective abelian hidden shift problem which can be solved in quantum subexponential time.)
5. Compute $i = \sigma \cdot j$.

Malleability oracles

Idea: Find preimage of $f(i)$ via a hidden shift computation.

Requirements:

1. Let G act transitively on I, and assume we have a malleability oracle for G at $f(i)$.
2. Let f be injective and let G be a finitely generated abelian group acting freely on I.

Strategy to compute i:

1. Pick any $j \in I$. (By transitivity of G, $\exists \sigma$ s.t. $i = \sigma \cdot j$.)
2. Define $F_0(g) := f(g \cdot j)$.
3. Define $F_1(g) := f(g \cdot i)$.
4. Compute σ such that $F_0(\sigma g) = F_1(g)$.
   (Injective abelian hidden shift problem which can be solved in quantum subexponential time.)
5. Compute $i = \sigma \cdot j$.

Malleability oracles

Idea: Find preimage of $f(i)$ via a hidden shift computation.

Requirements:

1. Let G act transitively on I, and assume we have a malleability oracle for G at $f(i)$.
2. Let f be injective and let G be a finitely generated abelian group acting freely on I.

Strategy to compute i:

1. Pick any $j \in I$. (By transitivity of G, $\exists \sigma$ s.t. $i = \sigma \cdot j$.)
2. Define $F_0(g) := f(g \cdot j)$.
3. Define $F_1(g) := f(g \cdot i)$.
4. Compute σ such that $F_0(\sigma g) = F_1(g)$.
   (Injective abelian hidden shift problem which can be solved in quantum subexponential time.)
5. Compute $i = \sigma \cdot j$.

Malleability oracles

Idea: Find preimage of $f(i)$ via a hidden shift computation.

Requirements:

1. Let G act transitively on I, and assume we have a malleability oracle for G at $f(i)$.
2. Let f be injective and let G be a finitely generated abelian group acting freely on I.

Strategy to compute i:

1. Pick any $j \in I$. (By transitivity of G, $\exists \sigma$ s.t. $i = \sigma \cdot j$.)
2. Define $F_0(g) := f(g \cdot j)$.
3. Define $F_1(g) := f(g \cdot i)$.
4. Compute σ such that $F_0(\sigma g) = F_1(g)$.
   (Injective abelian hidden shift problem which can be solved in quantum subexponential time.)
5. Compute $i = \sigma \cdot j$.

OWF underlying SIDH

Let E_0 with $j(E_0) = 1728$ be the starting curve in SIDH with known endomorphism ring, and let N_1 and N_2 be the security parameters of Alice and Bob respectively.

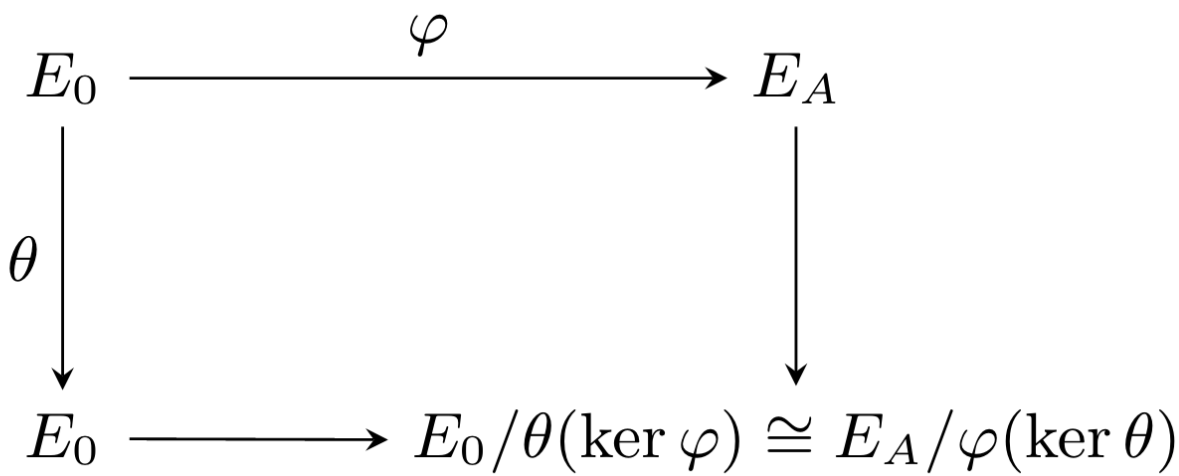
- $I :=$ {subgroups of order N_1 of E_0}
- $O :=$ {elliptic curves at distance N_1 from E_0}

$$f : I \to O$$
$$K \mapsto E_0/K \text{ and torsion point images}$$

Malleability oracle for SIDH

Let G be a multiplicative subgroup of $\left(\mathrm{End}(E_0)/N_1\mathrm{End}(E_0)\right)^*$, where representatives are endomorphisms of degree coprime to $\deg(\varphi)$.

Idea: Use torsion point information to construct malleability oracle at $E_A = f(K)$, i.e. given $\theta \in G$ compute $f(\theta \cdot K) = E_0/(\theta(K))$.



Lifting:

- For $\theta \in G$, find $\theta' \in \mathrm{End}(E_0)$ inducing the same action on I such that $\deg(\theta')|N_2$.
- Compute $\varphi(\ker\theta')$ using torsion point information.

Intermediate tasks

1. Partition I, i.e. cyclic subgroups of order N_1 of E_0, into (large) partitions such that f restricted to each one is injective.
2. Find abelian subgroups of $\left(\mathrm{End}(E_0)/N_1\mathrm{End}(E_0)\right)^*$ acting freely and transitively on each partition.
3. Give an algorithm to lift elements from G to an endomorphism of norm N_2.

Intermediate tasks

1. Partition I, i.e. cyclic subgroups of order N_1 of E_0, into (large) partitions such that f restricted to each one is injective.
2. Find abelian subgroups of $\left(\mathrm{End}(E_0)/N_1\mathrm{End}(E_0)\right)^*$ acting freely and transitively on each partition.
3. Give an algorithm to lift elements from G to an endomorphism of norm N_2.

Intermediate tasks

1. Partition I, i.e. cyclic subgroups of order N_1 of E_0, into (large) partitions such that f restricted to each one is injective.
2. Find abelian subgroups of $\left(\mathrm{End}(E_0)/N_1\mathrm{End}(E_0)\right)^*$ acting freely and transitively on each partition.
3. Give an algorithm to lift elements from G to an endomorphism of norm N_2.

Lifting of endomorphisms from $\pi\mathbb{Z}[\iota]$

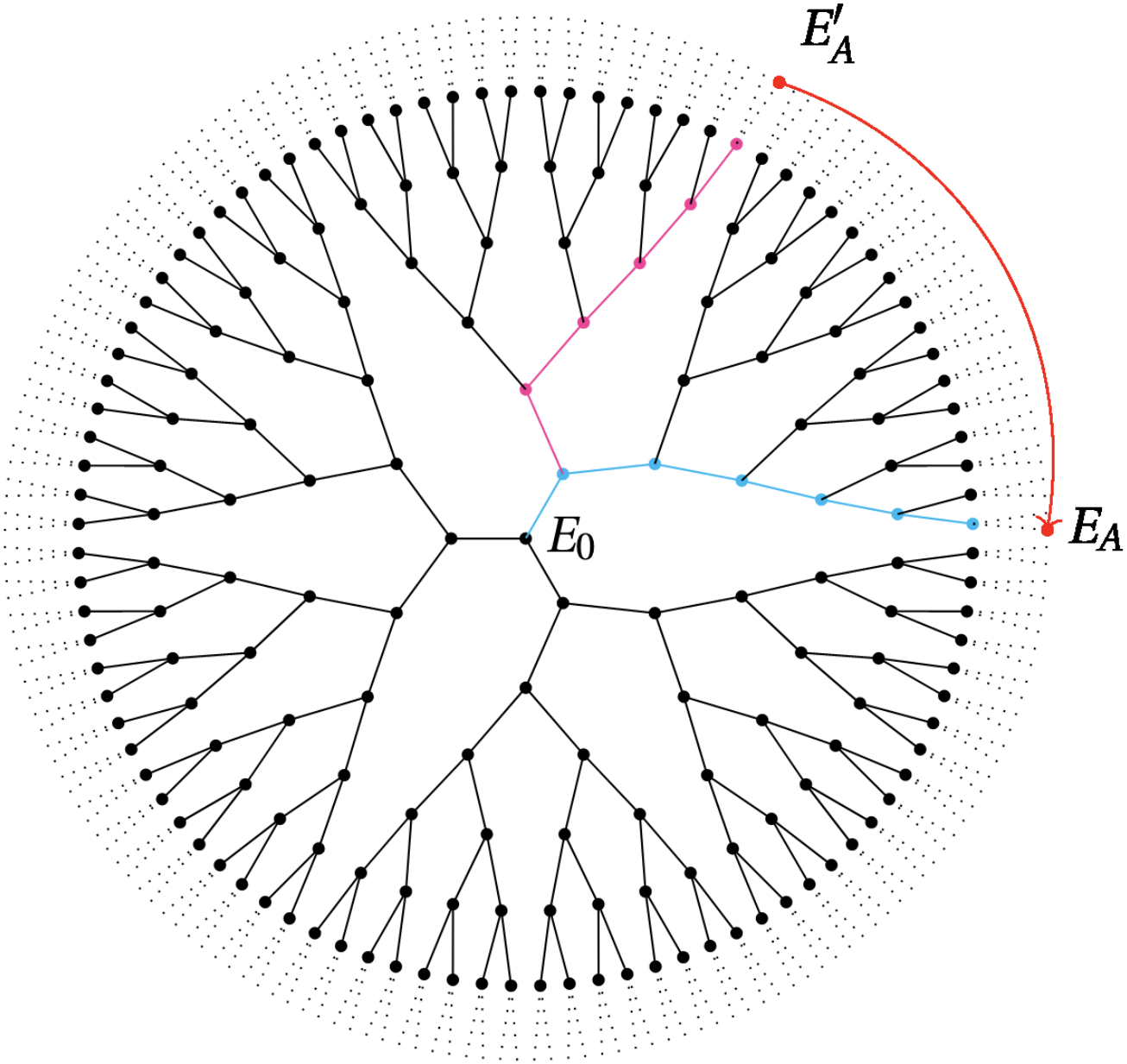
- Lifting algorithm: Given $\theta \in \pi\mathbb{Z}[\iota]$, return θ' of norm N_2 or eN_2 for some small e s.t. θ and θ' induce the same action on I.
- Works by solving a norm equation similar to the one in the KLPT algorithm.
- Our algorithm works for $N_2 > pN_1^4$.

Remark

For an improved lifting algorithm, we only expect solutions if

$$N_2 > pN_1^2.$$

Intuition continued



Summary

- For $N_2 > pN_1^4$, the problem underlying SIDH can be solved in quantum subexponential time via a reduction to the hidden shift problem.
- Despite SIDH's "non-commutative nature", there is an abelian group action on its key space.
- We embed the attack in a more general framework that also captures further attacks.
- SIKE or balanced SIDH parameters are not threatened by this attack, which breaks parameters that were already known to be insecure [DKL+21].

Summary

- For $N_2 > pN_1^4$, the problem underlying SIDH can be solved in quantum subexponential time via a reduction to the hidden shift problem.
- Despite SIDH's "non-commutative nature", there is an abelian group action on its key space.
- We embed the attack in a more general framework that also captures further attacks.
- SIKE or balanced SIDH parameters are not threatened by this attack, which breaks parameters that were already known to be insecure [DKL+21].

Summary

- For $N_2 > pN_1^4$, the problem underlying SIDH can be solved in quantum subexponential time via a reduction to the hidden shift problem.
- Despite SIDH's "non-commutative nature", there is an abelian group action on its key space.
- We embed the attack in a more general framework that also captures further attacks.
- SIKE or balanced SIDH parameters are not threatened by this attack, which breaks parameters that were already known to be insecure [DKL+21].

Summary

- For $N_2 > pN_1^4$, the problem underlying SIDH can be solved in quantum subexponential time via a reduction to the hidden shift problem.
- Despite SIDH's "non-commutative nature", there is an abelian group action on its key space.
- We embed the attack in a more general framework that also captures further attacks.
- SIKE or balanced SIDH parameters are not threatened by this attack, which breaks parameters that were already known to be insecure [DKL+21].