Constructing elliptic curve isogenies in quantum subexponential time

Andrew Childs David Jao Vladimir Soukharev

University of Waterloo

arXiv:1012.4019

Public-key cryptography in the quantum world

Shor 94: Quantum computers can efficiently
- factor integers
- calculate discrete logarithms (in any group)

This breaks two common public-key cryptosystems:
- RSA
- elliptic curve cryptography

How do quantum computers affect the security of PKC in general?

Practical question: we'd like to be able to send confidential information even after quantum computers are built

Theoretical question: crypto is a good setting for exploring the potential strengths/limitations of quantum computers

Isogeny-based elliptic curve cryptography

Not all elliptic curve cryptography is known to be quantumly broken!

Couveignes 97, Rostovstev-Stolbunov 06, Stolbunov 10: Public-key cryptosystems based on the assumption that it is hard to construct an *isogeny* between given elliptic curves

Best known classical algorithm takes time about $q^{1/4}$ [Galbraith, Hess, Smart 02]

Our main result:

Given two (isogenous, ordinary, with same endomorphism ring) elliptic curves over $\mathbb{F}_q$, there is a quantum algorithm that constructs an isogeny between them in time $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ (assuming GRH), where

$$L_q(\alpha, c) := \exp\left[(c + o(1))(\ln q)^\alpha (\ln \ln q)^{1-\alpha}\right]$$

Outline

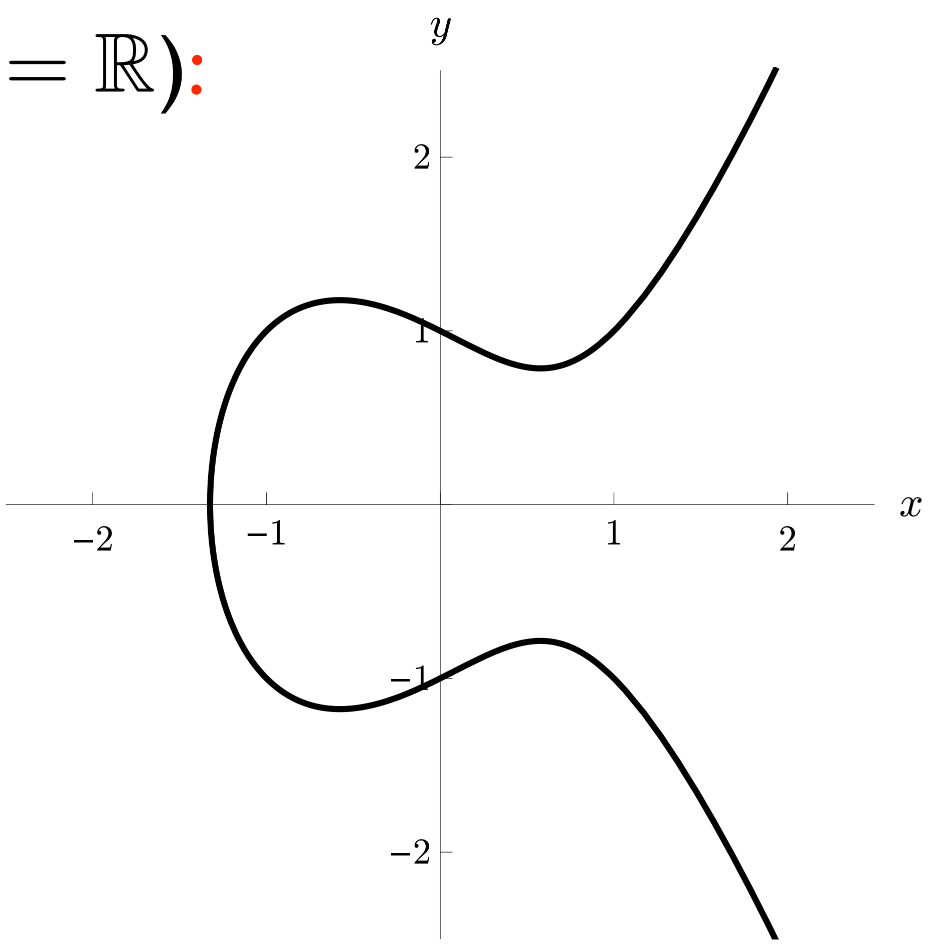
1. Elliptic curves
2. Isogenies
3. The abelian hidden shift problem
4. Computing the action of the ideal class group
5. Removing heuristic assumptions
6. Unknown endomorphism ring
7. Solving the abelian hidden shift problem with polynomial space
8. Open problems

Elliptic curves

Let $\mathbb{F}$ be a field of characteristic different from 2 or 3

An elliptic curve E is the set of points in $\mathbb{PF}^2$ satisfying an equation of the form $y^2 = x^3 + ax + b$

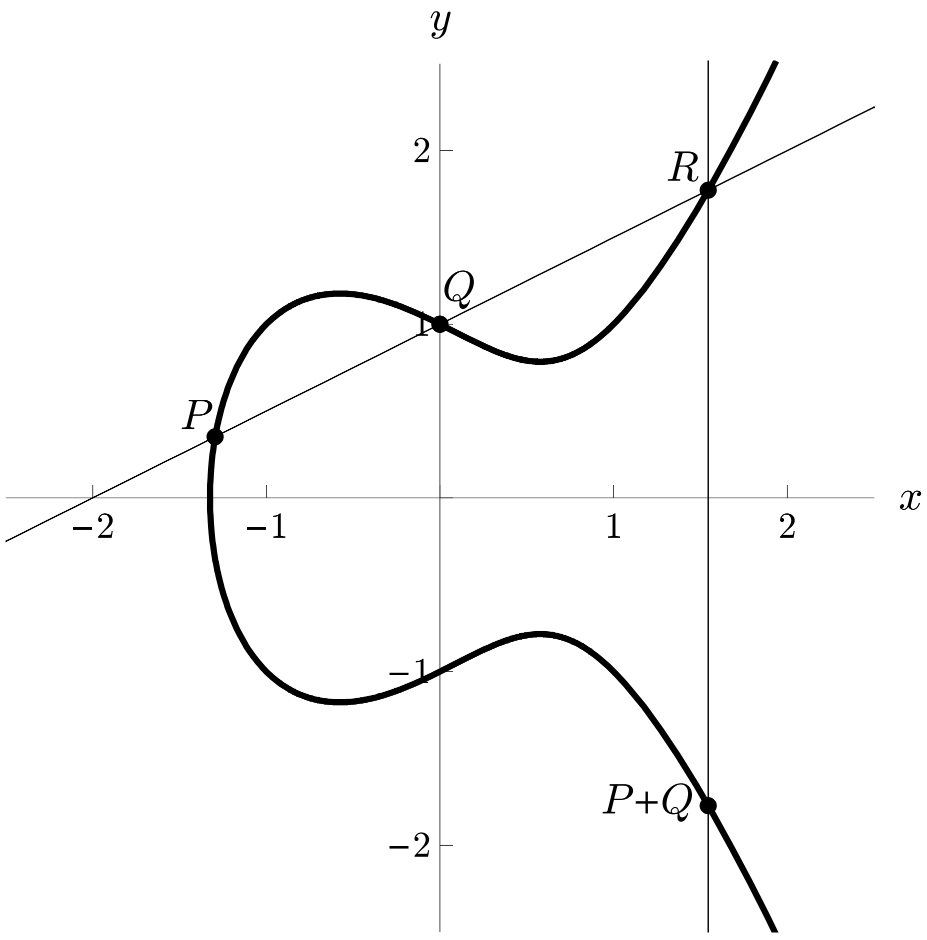
Example ($\mathbb{F} = \mathbb{R}$):



$$y^2 = x^3 - x + 1$$

Elliptic curve group

Geometric definition of a binary operation on points of E:



Algebraic definition:

for $x_P \neq x_Q$,

$$\lambda := \frac{y_Q - y_P}{x_Q - x_P}$$

$$x_{P+Q} = \lambda^2 - x_P - x_Q$$

$$y_{P+Q} = \lambda(x_P - x_{P+Q}) - y_P$$

for $P = Q$, $\lambda := \dfrac{3x_P^2 + a}{2y_P}$

for $(x_P, y_P) = (x_Q, -y_Q)$,

$$P + Q = \infty$$

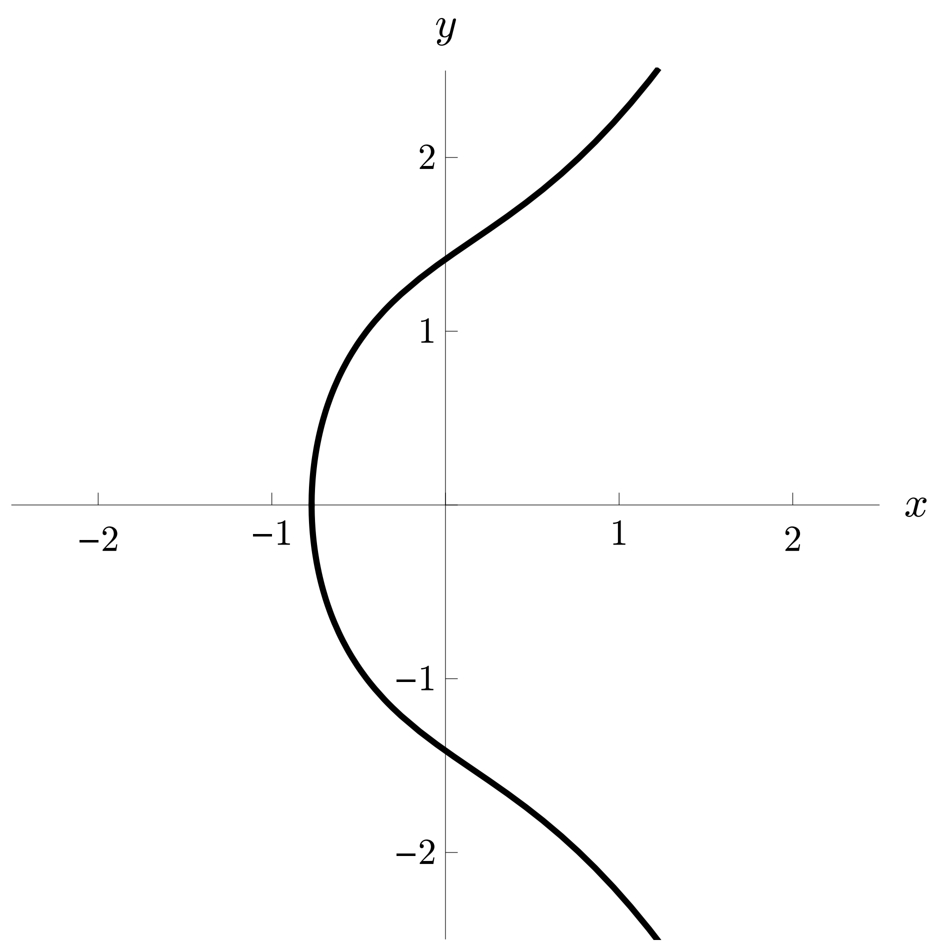
This defines an abelian group with additive identity ∞

Elliptic curves over finite fields

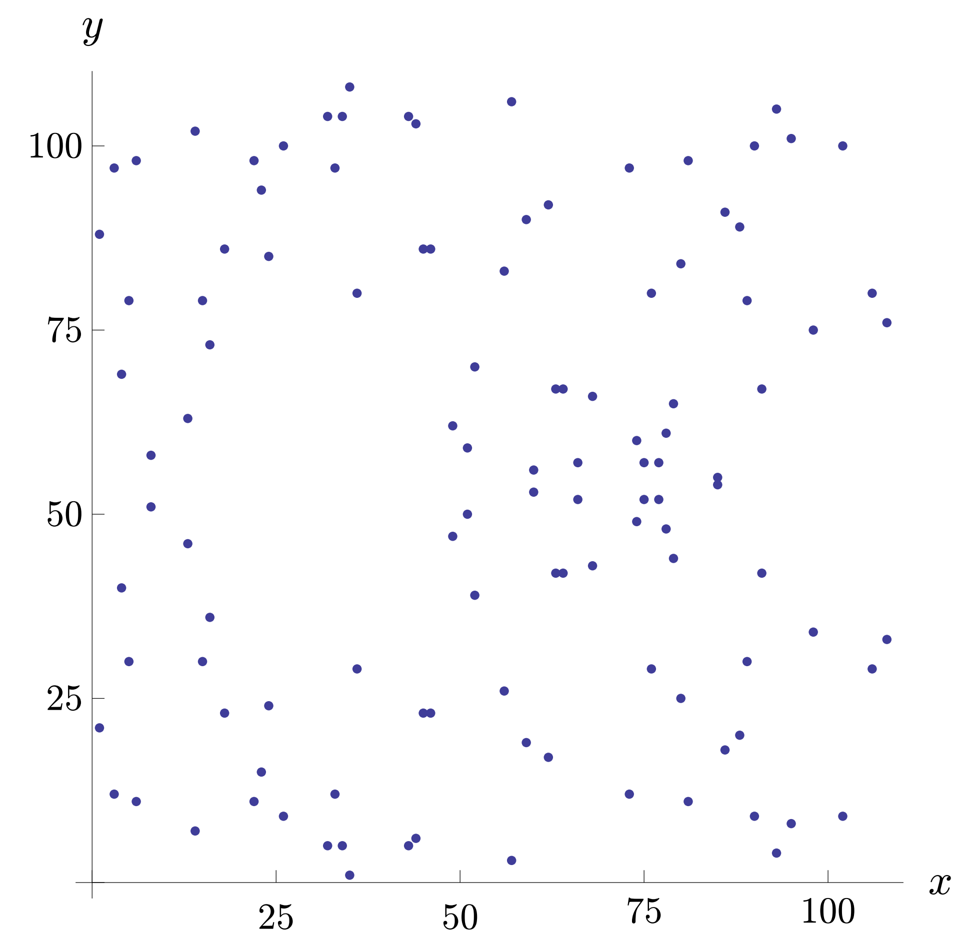
Cryptographic applications use a finite field $\mathbb{F}_q$

Example: $y^2 = x^3 + 2x + 2$

$\mathbb{F} = \mathbb{R}$



$\mathbb{F} = \mathbb{F}_{109}$



Elliptic curve isogenies

Let E_0, E_1 be elliptic curves

An isogeny $\phi : E_0 \to E_1$ is a rational map

$$\phi(x,y) = \left(\frac{f_x(x,y)}{g_x(x,y)}, \frac{f_y(x,y)}{g_y(x,y)} \right)$$

(f_x, f_y, g_x, g_y are polynomials) that is also a group homomorphism:

$$\phi((x,y)+(x',y')) = \phi(x,y) + \phi(x',y')$$

Example ($\mathbb{F} = \mathbb{F}_{109}$):

$$E_0 : \; y^2 = x^3 + 2x + 2 \qquad \xrightarrow{\phi} \qquad E_1 : \; y^2 = x^3 + 34x + 45$$

$$\phi(x,y) = \left(\frac{x^3 + 20x^2 + 50x + 6}{x^2 + 20x + 100}, \frac{(x^3 + 30x^2 + 23x + 52)y}{x^3 + 30x^2 + 82x + 19} \right)$$

Deciding isogeny

Theorem [Tate 66]: Two elliptic curves over a finite field are isogenous if and only if they have the same number of points.

There is a polynomial-time classical algorithm that counts the points on an elliptic curve [Schoof 85].

Thus a classical computer can decide isogeny in polynomial time.

The endomorphism ring

The set of isogenies from E to itself (over $\bar{\mathbb{F}}$) is denoted $\mathrm{End}(E)$

We assume E is *ordinary* (i.e., not *supersingular*), which is the case arising in proposed cryptosystems; then $\mathrm{End}(E) \cong \mathcal{O}_\Delta = \mathbb{Z}[\frac{\Delta+\sqrt{\Delta}}{2}]$ is an imaginary quadratic order of discriminant $\Delta < 0$

We also assume that $\mathrm{End}(E_0) = \mathrm{End}(E_1)$ (again, as in proposed cryptosystems)

Let $\mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$ denote the set of elliptic curves over $\mathbb{F}_q$ with n points and endomorphism ring $\mathcal{O}_\Delta$, up to isomorphism

Represent curves up to isomorphism by their j-invariants

$$E: \; y^2 = x^3 + ax + b \quad \Rightarrow \quad j(E) = 12^3 \frac{4a^3}{4a^3 + 27b^2}$$

Representing isogenies

The degree of an isogeny can be exponential (in $\log q$)

Example: The multiplication by m map,

$$(x, y) \mapsto \underbrace{(x, y) + \cdots + (x, y)}_{m}$$

is an isogeny of degree m^2

Thus we cannot even write down the rational map explicitly in polynomial time

Fact: Isogenies between elliptic curves with the same endomorphism ring can be represented by elements of a finite abelian group, the *ideal class group* of the endomorphism ring, denoted $\mathrm{Cl}(\mathcal{O}_\Delta)$

A group action

Thus we can view isogenies in terms of a group action

$$*\colon\ \mathrm{Cl}(\mathcal{O}_\Delta) \times \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta) \to \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$$
$$[\mathfrak{b}] * j(E) = j(E_\mathfrak{b})$$

where $E_\mathfrak{b}$ is the elliptic curve reached from E by an isogeny corresponding to the ideal class $[\mathfrak{b}]$

and $j(E)$ is the j-invariant of E

This action is regular [Waterhouse 69]:
for any E_0, E_1 there is a unique $[\mathfrak{b}]$ such that $[\mathfrak{b}] * j(E_0) = j(E_1)$

Isogeny-based cryptography

Example: Key exchange

Public parameters: field $\mathbb{F}_q$

elliptic curve $E \in \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$

Private key generation: choose an ideal $\mathfrak{b} = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_k^{e_k}$

where $\mathfrak{p}_1, \ldots, \mathfrak{p}_k$ have small norm

and $e_1, \ldots, e_k$ are small

Public key: $[\mathfrak{b}] * j(E)$

To establish a shared private key,

Alice publishes $[\mathfrak{b}_A] * j(E)$

Bob publishes $[\mathfrak{b}_B] * j(E)$

Alice computes $[\mathfrak{b}_A] * [\mathfrak{b}_B] * j(E)$

Bob computes $[\mathfrak{b}_B] * [\mathfrak{b}_A] * j(E)$

$$= [\mathfrak{b}_A] * [\mathfrak{b}_B] * j(E)$$

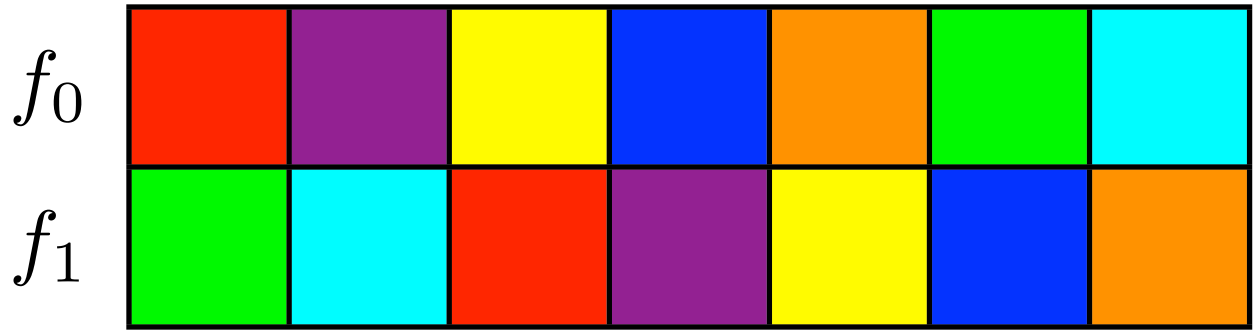
The abelian hidden shift problem

Let A be a known finite abelian group

Let $f_0 : A \to R$ be an injective function (for some finite set R)

Let $f_1 : A \to R$ be defined by $f_1(x) = f_0(xs)$ for some unknown $s \in A$

Problem: find s



For A cyclic, this is equivalent to the dihedral hidden subgroup problem

More generally, this is equivalent to the HSP in the generalized dihedral group $A \rtimes \mathbb{Z}_2$

Isogeny construction as a hidden shift problem

Define $f_0, f_1 : \mathrm{Cl}(\mathcal{O}_\Delta) \to \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$ by

$$f_0([\mathfrak{b}]) = [\mathfrak{b}] * j(E_0)$$
$$f_1([\mathfrak{b}]) = [\mathfrak{b}] * j(E_1)$$

E_0, E_1 are isogenous, so there is some $[\mathfrak{s}]$ such that

$$[\mathfrak{s}] * j(E_0) = j(E_1)$$

Since $*$ is a group action, $f_1([\mathfrak{b}]) = f_0([\mathfrak{b}][\mathfrak{s}])$

Since $*$ is regular, f_0 is injective

So this is an instance of the hidden shift problem in $\mathrm{Cl}(\mathcal{O}_\Delta)$ with hidden shift $[\mathfrak{s}]$

Kuperberg's algorithm

Theorem [Kuperberg 03]: There is a quantum algorithm that solves the abelian hidden shift problem in a group of order N with running time $\exp[O(\sqrt{\ln N})] = L_N(\frac{1}{2}, 0)$.

Main idea: Clebsch-Gordan sieve on coset states

Thus there is a quantum algorithm to construct an isogeny with running time

$$L_N(\tfrac{1}{2}, 0) \times c(N)$$

where $c(N)$ is the cost of evaluating the action

The same approach works for any group action (cf. "hard homogeneous spaces" [Couveignes 97])

Computing the action

Problem: Given E, Δ, $\mathfrak{b} \in \mathcal{O}_\Delta$, compute $[\mathfrak{b}] * j(E)$

Direct computation (using modular polynomials) takes time $O(\ell^3)$ for an ideal of norm ℓ

Instead we use an indirect approach:

- Choose a factor base of small prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_f$
- Find a factorization $[\mathfrak{b}] = [\mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_f^{e_f}]$ where $e_1, \ldots, e_f$ are small
- Compute $[\mathfrak{b}] * j(E)$ one small prime at a time

By optimizing the size of the factor base, this approach can be made to work in time $L(\frac{1}{2}, \frac{\sqrt{3}}{2})$.

Removing heuristic assumptions

Similar ideas appear in previous (classical) algorithms for isogenies:

- Galbraith, Hess, Smart 02: introduced idea of working in the ideal class group to compute the isogeny for a given ideal in time $q^{1/4}$
- Bisson, Sutherland 09: compute $\mathrm{End}(E)$ in subexponential time
- Jao, Soukharev 10: compute the isogeny for a given ideal in subexponential time

All of these results require heuristic assumptions in addition to the Generalized Riemann Hypothesis

We use a result on expansion properties of Cayley graphs of the ideal class group [Jao, Miller, Venkatesan 09] to avoid extra heuristics: our result assumes *only* GRH

The same technique works to remove the heuristic assumptions (except GRH) from the algorithm for isogeny computation [Jao, Soukharev 10]

Unknown endomorphism ring

Computing in $\mathrm{Cl}(\mathcal{O}_\Delta)$ requires us to know Δ

All proposed isogeny-based cryptosystems take $\mathcal{O}_\Delta$ to be a maximal order, so we can compute Δ as follows:

- Compute $t(E) := q + 1 - \#E$
- Factor $t(E)^2 - 4q = v^2 D$ where D is squarefree
- Then $\Delta = D$

But what if Δ is unknown?

Bisson, Sutherland 09: compute $\mathrm{End}(E)$ in time $L(\frac{1}{2}, \frac{\sqrt{3}}{2})$ (under significant heuristic assumptions)

Bisson 11 (using our expander graph idea): compute $\mathrm{End}(E)$ in time $L(\frac{1}{2}, \frac{1}{\sqrt{2}})$ under only GRH; also gives a new idea that improves the exponent of the group action computation from $\frac{\sqrt{3}}{2}$ to $\frac{1}{\sqrt{2}}$

Polynomial space

Kuperberg's algorithm uses space $\exp[\Theta(\sqrt{\ln N})]$

Regev 04 presented a modified algorithm using only polynomial space for the case $A = \mathbb{Z}_{2^n}$, with running time

$$\exp[O(\sqrt{n \ln n})] = L_{2^n}(\tfrac{1}{2}, O(1))$$

Combining Regev's ideas with techniques used by Kuperberg for the case of a general abelian group (of order N), and performing a careful analysis, we find an algorithm with running time $L_N(\frac{1}{2}, \sqrt{2})$

Thus there is a quantum algorithm to construct elliptic curve isogenies using only polynomial space in time $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2} + \sqrt{2})$

→

Kuperberg's approach

Consider the hidden shift problem in $\mathbb{Z}_N$

Standard approach to the hidden shift problem makes states

$$|\psi_x\rangle := \tfrac{1}{\sqrt{2}}(|0\rangle + \omega^{sx}|1\rangle) \qquad\qquad \omega := e^{2\pi i/N}$$

with $x \in \mathbb{Z}_N$ uniformly random

Suppose we can make $|\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_4\rangle \otimes \cdots \otimes |\psi_{2^{\lfloor \log_2 N \rfloor}}\rangle$; then a QFT reveals s

Idea: Combine states to make ones with more desirable labels

Measure parity:

$$|\psi_x\rangle \otimes |\psi_{x'}\rangle \xrightarrow{\text{even}} \tfrac{1}{\sqrt{2}}(|00\rangle + \omega^{s(x+x')}|11\rangle) \cong |\psi_{x+x'}\rangle$$

$$|\psi_x\rangle \otimes |\psi_{x'}\rangle \xrightarrow{\text{odd}} \tfrac{1}{\sqrt{2}}\omega^{sx'}(|01\rangle + \omega^{s(x-x')}|10\rangle) \cong |\psi_{x-x'}\rangle$$

This gives an algorithm with running time $2^{O(\sqrt{\log N})}$, but we have to store many states at once

Regev's approach: Combining more states

New idea: combine $k \gg 1$ states at a time

To cancel ℓ bits of the label:

$$\frac{1}{\sqrt{2^k}} \sum_{y \in \{0,1\}^k} \omega^{s(x \cdot y)} |y\rangle \underbrace{|x \cdot y \bmod 2^\ell\rangle}_{\text{measurement gives } r}$$

$$\mapsto \frac{1}{\sqrt{|Y_r|}} \sum_{y \in Y_r} \omega^{s(x \cdot y)} |y\rangle$$

$$Y_r = \{y \in \{0,1\}^k : x \cdot y \bmod 2^\ell = r\}$$

Compute the set Y_r (takes time 2^k)

Project onto $\mathrm{span}\{|y_1\rangle, |y_2\rangle\}$ and relabel:

$$\mapsto \frac{1}{\sqrt{2}} \Big(|0\rangle + \omega^{s(x \cdot y_2 - x \cdot y_1)} |1\rangle \Big)$$

Success probability is reasonable provided $k \gg \ell$

Note: it is not necessary to have $|Y_r| = O(1)$

Regev's approach: The pipeline of routines

For $j = 0, 1, \ldots, m$, let S_j include the states with last $j\ell$ bits canceled

Repeat
  While for all j there are fewer than k states from S_j
    Make a state from S_0
  End while
  Combine k states from some S_j to make a state from S_{j+1}
Until there is a state from S_m

We never store more than $O(mk)$ states at a time

If combinations work perfectly, we need to eventually make

$$1 + k + k^2 + \cdots + k^m \approx k^m \text{ states}$$

By Chernoff bounds, even if the combinations only succeed with constant probability, we only need $k^{(1+o(1))m}$ states

Optimizing the tradeoff

Cancel k bits in each of m stages: $mk \approx \log_2 N$

Running time of combination procedure: $\approx 2^k$

Total number of combinations: $\approx k^m$

Overall running time: $\approx 2^k k^m = 2^{k+m\log k}$

Let $k = c\sqrt{\log N \log\log N}$

Then $2^{k+m\log k} = L(\frac{1}{2}, c + \frac{1}{2c})$

Optimized with $c = \frac{1}{\sqrt{2}}$, giving running time $L(\frac{1}{2}, \sqrt{2})$

Making smaller labels

Given: states with labels in $\{0, 1, \ldots, B-1\}$ (uniformly random)

Produce: states with labels in $\{0, 1, \ldots, B'-1\}$ (uniformly random)

$$\frac{1}{\sqrt{2^k}} \sum_{y \in \{0,1\}^k} \omega^{s(x \cdot y)} |y\rangle \underbrace{|\lfloor (x \cdot y)/2B' \rfloor\rangle}_{\text{measurement gives } q}$$

$$\mapsto \frac{1}{\sqrt{|Y_q|}} \sum_{y \in Y_q} \omega^{s(x \cdot y)} |y\rangle$$

Compute the set $Y_q = \{y \in \{0,1\}^k \colon \lfloor (x \cdot y)/2B' \rfloor = q\}$

Project onto $\mathrm{span}\{|y_1\rangle, |y_2\rangle\}$ or $\mathrm{span}\{|y_3\rangle, |y_4\rangle\}$ or ...

Use rejection sampling to ensure that the distribution over the resulting label is uniform over $\{0, 1, \ldots, B'-1\}$

Lemma: This succeeds with constant probability if $4k \le \frac{B}{B'} \le \frac{2^k}{k}$

Reducing to the cyclic case

For a general abelian group $\mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_t}$, hidden shift states have the form

$$|\psi_x\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle + \exp\left[2\pi i\left(\frac{s_1 x_1}{N_1} + \cdots + \frac{s_t x_t}{N_t}\right)\right]|1\rangle\right)$$

If we can produce states with all components of x but one (say, the tth) equal to zero, we reduce to the cyclic case

Combination procedure: similar to the one for making smaller labels, using the quantity

$$\mu(x) := \sum_{j=1}^{t-1} x_j \prod_{j'=1}^{j-1} N_{j'}$$

Procedure and its analysis are simplified since we don't need to maintain a uniform distribution

Overall algorithm

Write $A = \mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_t}$ where each N_i is either odd or a power of 2

To determine s_i:

- For each j, make the state $|\psi_{2^j}\rangle$ as follows:
  - Sieve away components other than the ith
  - If N_i is odd
    - Under the automorphism $x \mapsto 2^{-j}x$, sieve toward smaller labels, making a state with label 1
  - If N_i is a power of 2
    - Sieve away the $j-1$ lowest-order bits, then sieve toward smaller labels

Theorem: With carefully chosen parameters, this algorithm has running time $L(\frac{1}{2}, \sqrt{2})$.

Open problems

- Breaking isogeny-based cryptography in polynomial time?
- Quantum algorithms for properties of a single curve:
  - computing the ideal class group
  - computing the endomorphism ring
- Generalizations:
  - evaluating/constructing isogenies between curves of different endomorphism ring
  - constructing isogenies between supersingular curves