A Quantum Algorithm for Computing the Unit Group of a Number Field:

joint w/ Sean Hallgren, Alexei Kitaev, Fang Song.

Main Computational problem for number fields: Given a number field K and ring of integers $\mathcal{O}$, compute

1) the unit group $\mathcal{O}^\times$ (special case is solving Pell's equation $x^2 - dy^2 = 1$ $\leftrightarrow$ find units in $\mathbb{Q}(\sqrt{d})$, $d > 0$)

2) the class group of K

3) solve the principal ideal problem (PIP). Given an ideal $I \subseteq \mathcal{O}$, determine if it is principal. If so, find a generator for I.

Example: $K = \mathbb{Q}(\sqrt{d})$, $d > 0$. d square-free

$\mathcal{O} = \mathbb{Z}[\sqrt{d}]$ $\quad d \equiv 2, 3 \pmod 4$

$\mathcal{O} = \mathbb{Z}\left[\frac{1+\sqrt{d}}{2}\right]$ $\quad d \equiv 1 \pmod 4$

$\mathcal{O}^\times$ = invertible elements of $\mathcal{O}$

Best known classical algorithms take time $\exp(\alpha^{1/2})$

where $\alpha = \log|D|$, $D = \mathrm{disc}(\mathbb{Q}(\sqrt{D}))$.

input size = $\log d$.

for arbitrary number fields: input size is $\log|D|$ & $\log n$ where $n = [K:\mathbb{Q}]$.

What can quantum algorithms do?

We have exponential speed ups for:

- factoring (Shor, 94)

- computing unit groups, class groups, and solve PIP in real quadratic number field (Hallgren '02)
- same for number fields of constant degree (Hallgren 05, Schmidt-Vollmer)
- same for function fields (E-Hallgren 12).

In the previous approach, when letting the degree vary one could not control the rounding errors from embedding the number fields into $\mathbb{R}^n$ or $\mathbb{C}^n$.

Theorem: Let K be a number field of degree n, $\mathcal{O}$ its ring of integers.

There is a polynomial time quantum algorithm for computing $\mathcal{O}^\times$, i.e. compute generators for $\mathcal{O}^\times$.

We'll give generators for the lattice $\mathrm{Log}(\mathcal{O}^\times) \subseteq \mathbb{R}^{s+t-1}$ where

s = # real embeddings, t = # complex conjugate pairs.

One ingredient for algorithm: Need to solve a generalization of the Hidden Subgroup problem (HSP).

HSP: Let G be a group G, S a set. Given $f: G \to S$ that is constant on cosets of a subgroup H of G and distinct on distinct cosets, find H.

If G is finite:
Want to solve this problem by making $\log|G|$ queries to f.

G

H
$g+H$
$\vdots$
$g'+H$

Example: When G is a finite abelian group, there is an efficient algorithm for solving the HSP.

Application: The discrete log problem reduces to HSP over a finite abelian group.

DLP: given a prime p and a generator g of $(\mathbb{Z}/p\mathbb{Z})^*$ and $x \in (\mathbb{Z}/p\mathbb{Z})^*$, find r s.t. $g^r \equiv x \pmod p$.

HSP setup: Define $f: \mathbb{Z}/(p-1)\mathbb{Z} \times \mathbb{Z}/(p-1)\mathbb{Z} \to (\mathbb{Z}/p\mathbb{Z})^*$ by letting $f(a,b) = g^a x^{-b} \pmod p$. Then $f(a,b) = 1$ iff $a = br$

$f(a,b) = 1$ iff $(a,b) \in \langle (r,1) \rangle$. Similarly $f(a,b) = g^k$ for $(a,b) \in \langle (r,1) \rangle + (k,0)$. So f is constant on cosets of $H = \langle (r,1) \rangle$.

Since there is an efficient quantum algorithm for HSP on $\mathbb{Z}/(p-1)\mathbb{Z} \times \mathbb{Z}/(p-1)\mathbb{Z}$, can solve the discrete log problem on a quantum computer efficiently.

Overview of Quantum Algorithm for unit group:

(will simplify and only get generators for $\mathrm{Log}(\mathcal{O}^\times)$)

Input: number field $K = \mathbb{Q}(\vartheta)$.

$$K \xrightarrow{\text{quantum reduction}} \left(\begin{array}{c} \text{Continuous HSP} \\ f: \mathbb{R}^{s+t-1} \to S \end{array} \right) \xrightarrow[\text{HSP alg.}]{\text{quantum}} \mathrm{Log}\,\mathcal{O}^\times \subseteq \mathbb{R}^{s+t-1}.$$

$$\mathrm{Log}: \mathcal{O}^\times \longrightarrow \mathbb{R}^{s+t-1}$$
$$z \longmapsto (\log|\tau_1(z)|, \dots, \log|\tau_s(z)|, 2\log|\tau_{s+1}(z)|, \dots, 2\log|\tau_{s+t}(z)|)$$

Then $\mathrm{Log}\,\mathcal{O}^\times$ is the hidden subgroup of $\mathbb{R}^{s+t-1}$ which corresponds to the function.

$$f: \mathbb{R}^{s+t-1} \longrightarrow \{\text{lattices in } \mathbb{R}^s \times \mathbb{C}^t\}$$

$$u \longmapsto e^u \tau(\mathcal{O}).$$

where

$$\tau: \mathcal{O} \longrightarrow \mathbb{R}^s \times \mathbb{C}^t$$

$$z \longmapsto (\tau_1(z), \dots, \tau_s(z), \tau_{s+1}(z), \dots, \tau_{s+t}(z)).$$

$u \in \mathbb{R}^{s+t-1} \longleftrightarrow u \in \mathbb{R}^n$

$\tau(\mathcal{O})$ is a lattice with basis $z_1, \dots, z_n$, $z_1 = \begin{pmatrix} z_{11} \\ \vdots \\ z_{n1} \end{pmatrix}, \dots, z_n = \begin{pmatrix} z_{1n} \\ \vdots \\ z_{nn} \end{pmatrix}$

then $e^u \tau(\mathcal{O})$ has basis

$$\begin{pmatrix} e^{u_1} z_{11} \\ \vdots \\ e^{u_n} z_{n1} \end{pmatrix}, \dots, \begin{pmatrix} e^{u_1} z_{1n} \\ \vdots \\ e^{u_n} z_{nn} \end{pmatrix}.$$

Example: $K = \mathbb{Q}(\sqrt{d})$, $d > 0$ square-free, $d \equiv 2, 3 \pmod 4$.

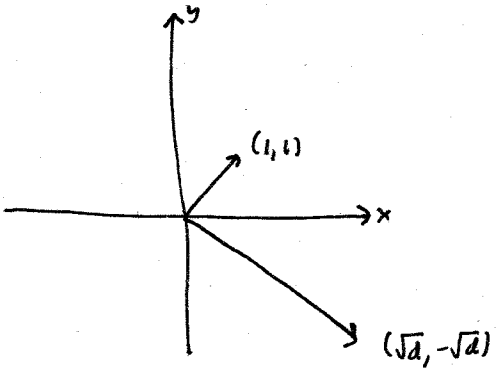
$\mathbb{Z}$-basis for $\mathcal{O}$ is $1, \sqrt{d}$. Lattice-basis for $\tau(\mathcal{O})$ is

$$\tau(1) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \tau(\sqrt{d}) = \begin{pmatrix} \sqrt{d} \\ -\sqrt{d} \end{pmatrix}$$

$$f: \mathbb{R} \longrightarrow \{\text{lattices in } \mathbb{R}^2\}$$

$$t \longmapsto e^t \tau(\mathcal{O})$$

$e^t \tau(\mathcal{O})$ has basis $b_1 = \begin{pmatrix} e^t \\ e^{-t} \end{pmatrix}$ and $b_2 = \begin{pmatrix} e^t \sqrt{d} \\ e^{-t} \sqrt{d} \end{pmatrix}$.



Have $\varepsilon\mathcal{O} = \mathcal{O}$ iff ε is a unit

$e^t\mathcal{O} = \mathcal{O}$ iff $t = \log|\text{unit}|$.

The function f is

(1) periodic on the unit lattice $f(x) = f(x+u)$ for $u \in \text{Log}\,\mathcal{O}^\times$

(2) f is distinct on cosets of $\text{Log}\,\mathcal{O}^\times$.

So f hides $\text{Log}\,\mathcal{O}^\times \subseteq \mathbb{R}$ (and also in the general case, $\text{Log}\,\mathcal{O}^\times \subseteq \mathbb{R}^{s+t-1}$.)

Problems with this approach:

1) there is no distinguished basis for the lattice $e^t\tau(\mathcal{O})$ but the HSP algorithm needs a unique representative.

To deal with this, compose f with another function g that maps L to a uniquely defined quantum superposition $|g(L)\rangle$ that encodes the lattice.

2) we need to compute a basis for $e^t\tau(\mathcal{O})$ in poly. time, but e^t can be doubly exponentially large

$\rightsquigarrow$ need to split up computation and use modification of LLL and Buchmann Pohst.

Let $h = g \circ f$.

(1) periodic $\langle h(x) | h(x+w) \rangle = 1 \quad \forall w \in Log\, \mathcal{O}^\times, x \in \mathbb{R}^{s+t-1}$

$\langle h(x) | h(x+t) \rangle \leq e^{-dist(t, Log\, \mathcal{O}^\times)}$

$\| h(x) \rangle - h(y) \rangle \| \leq a \|x-y\|$.

Applications to (breaking) lattice based cryptosystems:

- Some commonly used problems in lattice crypto.

BDD: Given a lattice and a target point that is promised to be "close" to some lattice vector v, find v.

Ideal-BDD: same as above but specialized to an ideal lattice

Ideal-SVP: given an ideal lattice I, find a vector that is (almost) as short as possible.

Principal-SVP: same as ideal SVP, but specialized to principal ideal lattices.

Ideal BDD seems hardest, get easier as you go down. There are problems with key size and efficiency. If you use ideal-BDD, ideal-SVP, etc the key sizes and efficiency are better.

SOLILOQUY

Smart-Vercauteren scheme

ideal-lattice based post quantum encryption schemes. The public-key is a principal ideal lattice $g\mathcal{O}$ that has generator g (short)

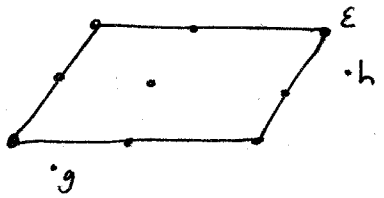
Secret key generator g.

Attack on Soliloquy:

Step 1: $\rightsquigarrow$ solve PIP. to get some generator h of gO.; $hO = gO$.

Step 2: adjust h by a unit to get a short generator g. Which unit?

Step 3: Solve BDD in the unit lattice for 2-power cyclotomic



$g = \varepsilon^{-1} h$

$\text{Log}(g) + \text{Log}(\varepsilon) = \text{Log}(h)$.

This breaks these two encryption schemes.