Peikert's C-Sieves paper

Shahed Sharif

October 2, 2019

1 Basics of quantum computing

In quantum computing, data takes the form of elements of $\mathbb{P}^{N-1}_{\mathbb{C}}$ for some N. Quantum operations are elements of $PU(N)$, the projective unitary group. Most of these cannot be implemented efficiently (that is, in quantum polynomial time), but in the course of this talk we will only encounter efficient operations. We write elements of $\mathbb{P}^{N-1}_{\mathbb{C}}$ as vectors, or *quantum states*,

$$|\psi\rangle = \sum_{j=1}^{N} \alpha_j |j\rangle .$$

We cannot access the coefficients α_j directly. Instead, we have a *measurement* operation, which works as follows. Lift $|\psi\rangle$ to $\mathbb{C}^N$; without loss of generality the lift v has length 1. Choose an orthonormal basis $w_1, \cdots, w_N$ for $\mathbb{C}^N$, and write

$$v = \sum \beta_j w_j.$$

Then measurement is a probabilistic operator which takes $|\psi\rangle$ as input and outputs w_j with probability $|\beta_j|^2$. It is also a *destructive* process; $|\psi\rangle$ becomes w_j at the conclusion, and so successive measurements give no additional information. We can't choose just any basis w_i, but the set of permissible bases won't concern us at this time.

We can also tensor state vectors. We usually write $|i\rangle |j\rangle$ or $|ij\rangle$ in place of $|i\rangle \otimes |j\rangle$. The tensor product, on the level of projective spaces, is given by the Segre embedding $\mathbb{P}^{M-1} \times \mathbb{P}^{N-1} \to \mathbb{P}^{NM-1}$.

There is a way of combining *partial* measurement with unitary operators to obtain the following.

Proposition 1.1. *Given an efficient classical function $f : \{1, \cdots, N\} \to X$ for some set X, there is a probabilistic quantum algorithm that takes as input*

$$|\psi\rangle = \sum_{j=1}^{N} \alpha_j |j\rangle ,$$

and outputs a random element $\kappa \in \mathrm{im}(f)$*, while also transforming* $|\psi\rangle$ *to*

$$\sum_{f(j)=\kappa} \alpha_j |j\rangle .$$

In the literature, this procedure is often referred to as "measuring f."

Lastly, the most important quantum operator is the *quantum Fourier transform*. Let $\chi : \mathbb{Z}/N\mathbb{Z} \to \mathbb{C}$ be a primitive character; for example, χ descends from

$$m \mapsto e^{2\pi i m/N}.$$

Then the QFT on $\mathbb{P}^{N-1}_{\mathbb{C}}$ is the unique element Q of $PU(N)$ for which

$$Q|j\rangle = \sum_{w=1}^{N} \chi(-jw)|w\rangle .$$

2 Computational problems: CSIDH

In this section, I will describe the mathematics underlying the cryptosystem CSIDH, and fit the mathematics into the framework of known problems in quantum computing. Choose a prime p and a supersingular curve E over $\mathbb{F}_p$. Then the endomorphism ring of E is the ring of integers in an iqf K. Let $\mathrm{Isog}(E)$ denote the set of elliptic curves which are isogenous to E over $\mathbb{F}_p$. It is well-known that the class group $\mathrm{Cl}(K)$ acts freely and transitively on $\mathrm{Isog}(E)$. For an ideal class I, denote this action as $I * E$. The security of CSIDH is based on the problem of deducing I, given the elliptic curves E and $I * E$.

Let A be an abelian group, and let $s \in A$ be a secret element. Let $f_0, f_1 : A \to X$ be efficiently computable injective functions such that

$$f_1(x) = f_0(x + s).$$

The *Hidden Shift Problem* is to find s.

It turns out the CSIDH problem above is an instance of the Hidden Shift Problem. For let $A = \mathrm{Cl}(K)$, $E' = I * E$, and define $f_i : A \to \mathbb{F}_p$ by

$$f_0(J) = j(J * E)$$
$$f_1(J) = j(J * E').$$

Our secret element s is none other than I!

3 Kuperberg's algorithm and collimation

Kuperberg in 2003 published an algorithm for solving the HSP over $A = \mathbb{Z}/N\mathbb{Z}$, which he has since improved [Kup11]. This was adapted by Childs, Jao, and Soukharev in 2010 to a quantum attack on CSIDH. (Well, not CSIDH itself, but a related cryptosystem; their attack also applies to CSIDH.) The bottleneck, it

turns out, is computing $*$ efficiently. However, following Peikert [Pei19], we will focus on the HSP part of the algorithm.

Suppose for convenience that $N = 2^n$. Let $\chi : A \to \mathbb{C}$ be a primitive character. The main point of Kuperberg's algorithm is that it produces many pairs $(b, |\psi\rangle)$, where $b \in A$ is random and

$$|\psi\rangle = |0\rangle + \chi(bs)\,|1\rangle\,.$$

Unfortunately, this is not enough to deduce s. Kuperberg takes many such $|\psi\rangle$ and *collimates* them; that is, he applies quantum operations to probabilistically combine them. If

$$|\psi_1\rangle = |0\rangle + \chi(b_1 s)\,|1\rangle \text{ and}$$
$$|\psi_2\rangle = |0\rangle + \chi(b_2 s)\,|1\rangle$$

then Kuperberg showed that one can obtain a new state

$$|0\rangle + \chi(cs)\,|1\rangle$$

where c is either $b_1 \pm b_2$, each with 50% probability. Iterating until one obtains $c = N/2$, we have the state

$$|\varphi\rangle = |0\rangle + (-1)^s\,|1\rangle\,.$$

But observe that $|0\rangle + |1\rangle\,, |0\rangle - |1\rangle$ form an orthogonal basis for $\mathbb{C}^2$. Measuring $|\varphi\rangle$ with respect to the normalized basis, we can deduce the parity of s, and hence its last bit. We can then iterate with $A = \mathbb{Z}/(N/2)\mathbb{Z}$ to find the next to last bit of s, and so on.

4 Peikert's improvements

Peikert adapted Kuperberg's idea to detect *most significant bits* of s. The algorithm takes as input Kuperberg's oracle; that is, an oracle that produces pairs

$$(b, |0\rangle + \chi(bs)\,|1\rangle).$$

Also as part of the input is a parameter t. The algorithm outputs the t most significant bits of s.

The idea is to apply collimation to produce a state of the form

$$|\psi\rangle = \sum_{j=0}^{T-1} \chi(js)\,|j\rangle\,,$$

where $T = 2^t$ (or approximately 2^t). We will show below how to obtain such a state. For convenience assume that $T \mid N$. Applying the QFT with respect to T, we then get

$$\sum_{w=0}^{T-1} \sum_{j=0}^{T-1} \chi(j(s - wN/T))\,|w\rangle\,.$$

As w varies, most of the coefficients of $|w\rangle$ will be close to 0, since they will be essentially equidistributed around the unit circle. The only situation where the coefficient will be large is when $s \approx wN/T$. This tells us approximately $\log T$ bits of s. The condition $T \mid N$ is not necessary, and in fact $T > N$ is possible, in which case we get s exactly.

4.1 Peikert's collimation

The main idea for Peikert's collimation is to take k states $|\psi_i\rangle$ from Kuperberg's oracle—say,

$$|\psi_i\rangle = |0\rangle + \chi(b_i s)\,|1\rangle\,.$$

Write b_i as a function $\{0,1\} \to \mathbb{Z}/N\mathbb{Z}$ via $b_i(0) = 0$, $b_i(1) = b_i$. Let $|\psi\rangle = \otimes\,|\psi_i\rangle$. If $\vec{j} \in \{0,1\}^k$, let $b(\vec{j}) = \sum b_i(j_i)$. Thus

$$|\psi\rangle = \sum \chi(b(\vec{j})s)\,|\vec{j}\rangle\,.$$

(1) Restrict b values. Choose T. Let $q : \{0,1\}^k \to \mathbb{Z}$ be defined by

$$q(\vec{j}) = \lfloor b(\vec{j})/T \rfloor.$$

Measure $|\psi\rangle$ with respect to q. One obtains

$$|\psi\rangle = \sum \chi(b(\vec{j})s)\,|\vec{j}\rangle$$

where $\vec{j}$ varies over elements satisyfing $b(\vec{j}) \div T$ has a fixed quotient q_0. We may therefore factor out $\chi(q_0 T)$ from every term. Since $|\psi\rangle \in \mathbb{P}^{N-1}_{\mathbb{C}}$, we can ignore this factor and instead simply assume that $0 \le b(\vec{j}) \le T-1$.

(2) Regularize. The measurement outputs q_0 as well, and so we can compute the list of $\vec{j}$ that appear in $|\psi\rangle$ above. Compute a set X of $\vec{j}$s for which $b(\vec{j})$, $\vec{j} \in X$, attains every value between 0 and $T-1$ exactly once. If this is impossible (say, if no value satisfies $b(\vec{j}) = T-1$), start over and recompute $|\psi\rangle$.

Let f be an indicator function for X; that is $f(\vec{j}) = 1$ if $\vec{j} \in X$, and $f(\vec{j}) = 0$ otherwise. Measure $|\psi\rangle$ with respect to f. If we get 0, choose a new X disjoint from the previous choice and try again. Otherwise, $|\psi\rangle$ is now of the form

$$\sum_{\vec{j} \in X} \chi(b(\vec{j})s)\,|\vec{j}\rangle\,.$$

(3) Renumber. We have $b : X \to \{0, 1, \ldots, T-1\}$ is a bijection. We use b to relabel the basis states, so that $|\psi\rangle$ now looks like

$$\sum_{j=0}^{T-1} \chi(js)\,|j\rangle\,.$$

We can do this since the set X and the values of b on this set are known.

There are a couple of other computational tricks that Peikert uses, but I omit them.

5 Kuperberg's oracle

We now describe Kuperberg's oracle; that is, given a hidden shift problem on $\mathbb{Z}/N\mathbb{Z}$ with shift a, how Kuperberg produces states of the form

$$|0\rangle + \chi(bs)\,|1\rangle\,.$$

Let $D = \mathbb{Z}/N\mathbb{Z} \times \{0,1\}$. Let $f_0, f_1 : \mathbb{Z}/N\mathbb{Z} \to X$ be the hidden shift oracles; that is, they are injective functions with $f_1(x) = f_0(x+s)$. Define

$$f : D \to X$$

by $f(a,i) = f_i(a)$.

We work over $\mathbb{C}^D$, the free vector space on D, and its associated projective space. We may start with the state

$$|\psi\rangle = \sum_{(a,i)\in D} |ai\rangle\,.$$

We then measure this state with respect to f to obtain some $x \in X$ and

$$|\psi\rangle = \sum_{f(a,i)=x} |ai\rangle\,.$$

But the f_i are injective, so the above equals

$$|a_0 0\rangle + |(a_0 - s)1\rangle$$

for some a_0. Now apply the QFT with respect to $\mathbb{Z}/N\mathbb{Z}$ to obtain

$$\sum_w \chi(-a_0 w)\,|w0\rangle + \sum_w \chi(-a_0 w)\chi(sw)\,|w1\rangle\,.$$

Finally we measure w, obtaining some value b and the state

$$\chi(-a_0 b)\,|b0\rangle + \chi(-a_0 b)\chi(sb)\,|b1\rangle\,.$$

Since we are in projective space, we may omit $\chi(-a_0 b)$. We will also forget the first coordinate inside the $|\rangle$, and thus have the state

$$|0\rangle + \chi(sb)\,|1\rangle$$

as desired.

6 Childs-Jao-Soukharev

As mentioned earlier, the bottleneck in applying Kuperberg's algorithm to attacking CSIDH is computing the action of the class group $\mathrm{Cl}(K)$ on the isogeny class $\mathrm{Isog}(E)$. We describe the approach of [CJS10].

The group action is as follows. For a given ideal class in $\mathrm{Cl}(K)$, let J be an integral ideal representing it. Let $m = \mathrm{Nm}(J)$. Then there is some $\alpha \in \mathscr{O}_K$ such that J is generated by m and α. Typically, J is represented by a quadratic form, so it is not too difficult to compute both m and α. Any elliptic curve $C \in \mathrm{Isog}(E)$ comes equipped with an action of $\mathscr{O}_K$, and so both m, α act on C. Define

$$C[J] = C[m] \cap C[\alpha].$$

Then $J * C$ is defined to be $C/C[J]$.

However, directly computing $J * C$ is not efficient! The problem is that $C[J]$ has cardinality m, and if m is large, then computing the quotient $C/C[J]$ is inefficient. Instead, Childs-Jao-Soukharev proceed as follows.

Let π be the canonical quotient map

$$\pi : C \to C/C[J] = J * C,$$

First, compute prime ideals with small norm $\mathfrak{p}_1, \ldots, \mathfrak{p}_r$ such that their product is in the ideal class of J. This is done by combining algorithms of Bernstein and Seysen. Let $C_0 = C$, and recursively compute

$$\pi_i : C_{i-1} \to C_i := \mathfrak{p}_i * C_{i-1}.$$

Then we have $\pi = \pi_r \circ \cdots \circ \pi_1$, and in particular $C_r = J * C$. While not precisely *efficient*, it is shown that if GRH holds, then the algorithm to compute $J * C$ is subexponential in $\log p$, where $\mathbb{F}_p$ is our base field.

References

[CJS10] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. 2010, arXiv:1012.4019. 5

[Kup11] Greg Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. 2011, arXiv:1112.3333. 2

[Pei19] Chris Peikert. He gives c-sieves on the csidh. Cryptology ePrint Archive, Report 2019/725, 2019. https://eprint.iacr.org/2019/725. 3