Quantum Security Analysis of CSIDH

Xavier Bonnetain[1], André Schrottenloher[2]

[1] University of Waterloo, Waterloo, ON, Canada
[2] Cryptology Group, WI

Summary

CSIDH-512 does not reach NIST level 1 security.

- It does not change anything asymptotically, and it's just a matter of choosing higher parameter sizes.
- We can use the same algorithms to estimate the security of higher instances.

There are several trade-offs possible, and the security depends on many assumptions:

- what are the respective costs of classical / quantum operations?
- do we have QRACM?
- etc.

Outline

Preliminaries

Cost metrics

NIST level 1 security

Breaking CSIDH-512 should be "as hard as a key-recovery on AES-128".

We interpret this as:

One shouldn't break CSIDH-512 using less than 2^{128} classical operations and $2^{83.4}$ Clifford+T gates (from [JNRV20]).

- The algorithms that we consider are **hybrid**: quantum and classical costs may be different
- We avoid QRACM (contrary to [Peikert20])

Jaques, Naehrig, Roetteler, Virdia, "Implementing Grover Oracles for Quantum Key Search on AES and LowMC", EUROCRYPT 2020

Peikert, "He Gives C-Sieves on the CSIDH", EUROCRYPT 2020

Attack outline

Problem

Given two curves E_A, E_B, find an isogeny between E_A and E_B $\iff$ find an $s \in \mathcal{C}\ell(\mathcal{O})$ such that $E_B = s \cdot E_A$.

This reduces to a hidden shift problem (or dihedral hidden subgroup, DHS).

Given f, g s.t $f(x) = g(x+s)$, find s. Here $g(x) = x \cdot E_A$ and $f(x) = x \cdot E_B$.

- **Step 1**: use a quantum DHS algorithm, find the number of oracle queries & additional computations
- **Step 2**: bound the cost of an oracle query: **evaluate the class group action** (see BLMP20 for details)

Bernstein, Lange, Martindale, Panny, "Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies", EUROCRYPT 2020

DHS Algorithms

Hidden shift problem

Hidden shift problem

- $f, g : \mathcal{G} \to X$ injective
- $s \in \mathcal{G}$
- $f(x) = g(x+s)$
- Goal : find s, given oracle access to f and g.

Classical resolution

Find a collision, in $\Omega\left(2^{n/2}\right)$ samples.

Kuperberg's algorithm: labeled qubits (in $\mathbb{Z}/(2^n)$)

- Start with $|0\rangle |0\rangle |0\rangle$
- Apply $H : \sum_{b=0}^{1}, \sum_{x=0}^{2^n-1} |b\rangle |x\rangle |0\rangle$
- Apply the quantum oracles

$$\sum_x |0\rangle |x\rangle |f(x)\rangle + |1\rangle |x\rangle |g(x)\rangle$$

- Measure in the last register $y = f(x_0) = g(x_0 + s)$

$$|0\rangle |x_0\rangle + |1\rangle |x_0 + s\rangle$$

- Apply a quantum Fourier Transform

$$\sum_\ell \exp\left(2i\pi \frac{x_0 \ell}{2^n}\right) |0\rangle |\ell\rangle + \exp\left(2i\pi \frac{(x_0 + s)\ell}{2^n}\right) |1\rangle |\ell\rangle$$

- Measure ℓ

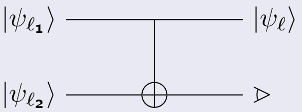
$$|\psi_\ell\rangle = |0\rangle + \exp\left(2i\pi \frac{s\ell}{2^n}\right) |1\rangle$$

Combining qubits

Labeled qubits

- $|\psi_\ell\rangle = |0\rangle + \exp\left(2i\pi \frac{s\ell}{2^n}\right)|1\rangle$
- $|\psi_{2^{n-1}}\rangle = |0\rangle + (-1)^s |1\rangle$

Combination: CNOT [Kup05]



$$(\ell_1, \ell_2) \mapsto \ell_1 \pm \ell_2 \mod 2^n$$

Let's say we want to obtain **the label** $1 \mod 2^n$: at each combination, we reduce the value by a factor $2^{\mathcal{O}(\sqrt{n})}$.

Complexity

Asymptotic complexity $\widetilde{\mathcal{O}}\left(2^{\sqrt{2\log_2(3)n}}\right)$ quantum time and memory

For cyclic groups (odd order)

Assume that the group is cyclic of order $N \simeq 2^n$. There is a standard technique:

- We produce label qubits $|\psi_{2^i \bmod N}\rangle$ for all $i \leq n$
- We do a QFT modulo 2^n on the result
- With probability $\geq \frac{4}{\pi^2}$ we will obtain s directly

Focusing on 1

- Having a generic cyclic group changes nothing for the production of 1
- If N is odd, being able to produce the label 1 is enough (to produce 2^i, multiply all the labels by $2^{-i} \bmod N$)

Numbers!

We can simulate the whole algorithm by simulating the combination:

- we start from random labels
- we combine ℓ_1, ℓ_2 by taking $\ell_1 \pm \ell_2$ at random

Results for odd cyclic groups

From simulations, a cost in $2^{1.8\sqrt{n}+4.3}$ queries overall (as with $N = 2^n$ in [BNP18], the polynomial factor is a constant in practice).

CSIDH p size	n	Queries	Qubits
512	256	2^{33}	2^{31}
1024	512	2^{45}	2^{43}
1792	896	2^{58}	2^{56}

Bonnetain, Naya-Plasencia, "Hidden Shift Quantum Cryptanalysis and Implications", ASIACRYPT 2018

Regev's (and CJS) variant

Choose some value B.

- Start with $|\psi_{\ell_1}\rangle, \ldots |\psi_{\ell_k}\rangle$

$$\bigotimes_j |\psi_{\ell_j}\rangle = \sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{N}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

- Compute $\lfloor \sum_j \ell_j b_j / B \rfloor$

$$\sum_{b_j \in \{0,1\}} \exp\left(2i\pi \frac{s}{N}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle \, |\lfloor \sum_j \ell_j b_j / B \rfloor\rangle$$

- Measure a value V

$$\sum_{b_j \in \{0,1\}, \lfloor \sum_j \ell_j b_j / B \rfloor = V} \exp\left(2i\pi \frac{s}{N}\left(\sum \ell_j b_j\right)\right) |b_1\rangle \ldots |b_k\rangle$$

Regev, "A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space", http://arxiv.org/abs/quant-ph/0406151

Regev's variant (ctd)

- Measure a value V

$$\sum_{b_j \in \{0,1\}, \lfloor \sum_j \ell_j b_j / B \rfloor = V} \exp\left(2i\pi \frac{s}{N} \left(\sum \ell_j b_j\right)\right) |b_1\rangle \dots |b_k\rangle$$

- Find two solutions $(b_1, \dots, b_k), (b'_1, \dots, b'_k)$ of

$$\lfloor \sum_j \ell_j b_j / B \rfloor = V$$

- Project on them
- Map $(b_1, \dots, b_k)$ to 0, $(b'_1, \dots, b'_k)$ to 1

$$|0\rangle + \exp\left(2i\pi \frac{s}{N} \left(\sum \ell_j b'_j - \sum \ell_j b_j\right)\right) |1\rangle$$

- New labeled qubit $|\psi_\ell\rangle$, with $\ell = \sum \ell_j b'_j - \sum \ell_j b_j$, $\ell \leq B$

Trade-offs

We have a routine that, from $|\psi_{\ell_1}\rangle, \ldots |\psi_{\ell_k}\rangle$ and B, produces $|\psi_\ell\rangle$ with $\ell \leq B$ (we take $k \simeq \log_2(N/B)$ generally). It requires to solve $\lfloor \sum_j \ell_j b_j / B \rfloor = V$.

Regev and CJS use this recursively to get $\widetilde{\mathcal{O}}\left(2^{\sqrt{2n \log_2(n)}}\right)$ queries and $\mathcal{O}(n)$ quantum memory.

Minimal quantum cost: the best for CSIDH-512

Produce $\ell = 1$ in **only one step** ($k = n, B = 2$). Costs $8n^2$ queries, and requires to solve an n-bit classical subset-sum problem ($\widetilde{\mathcal{O}}\left(2^{0.283n}\right)$ classical time/memory).

- For CSIDH-512, $n = 256$: this gives $2^{72.5}$ to repeat 256 times (all labels), then 3 times (final Fourier transform) $\implies 2^{82}$ operations.
- We're cheating by omitting the polynomial factor, but the cost **will be** below $2^{256/2} = 2^{128}$ anyway.

Kuperberg's second algorithm

Now, instead of a single label, we use multi-labeled qubit registers:

$$|\psi_{(\ell_0,\ldots,\ell_{k-1})}\rangle = \frac{1}{\sqrt{k}} \sum_{0\leq i\leq k-1} \exp\left(2i\pi\frac{s\ell_i}{N}\right)|i\rangle \quad .$$

We'll use much more classical memory to store the labels ℓ_i.

Combination:

- Start with $\left(|\psi_{(\ell_0,\ldots,\ell_{M-1})}\rangle, |\psi_{(\ell'_0,\ldots,\ell'_{M-1})}\rangle\right) : \forall i, \ell_i < 2^a, \ell'_i < 2^a$
- Tensor

$$|\psi_{(\ell_0,\ldots,\ell_{M-1})}\rangle |\psi_{(\ell'_0,\ldots,\ell'_{M-1})}\rangle = \sum_{i,j=0}^{M-1} \exp\left(2i\pi\frac{s(\ell_i+\ell'_j)}{N}\right)|i\rangle|j\rangle$$

Kuperberg, "Another Subexponential-time Quantum Algorithm for the Dihedral Hidden Subgroup Problem", TQC 2013

Kuperberg's second algorithm (ctd.)

- Add an ancilla register, apply $|i\rangle|j\rangle|0\rangle \mapsto |i\rangle|j\rangle|\lfloor(\ell_i+\ell'_j)/2^{a-r}\rfloor\rangle$
- Measure the ancilla register, leaving with

$$V \text{ and } \sum_{i,j:\lfloor(\ell_i+\ell'_j)/2^{a-r}\rfloor=V} \exp\left(2i\pi\frac{s(\ell_i+\ell'_j)}{N}\right)|i\rangle|j\rangle$$

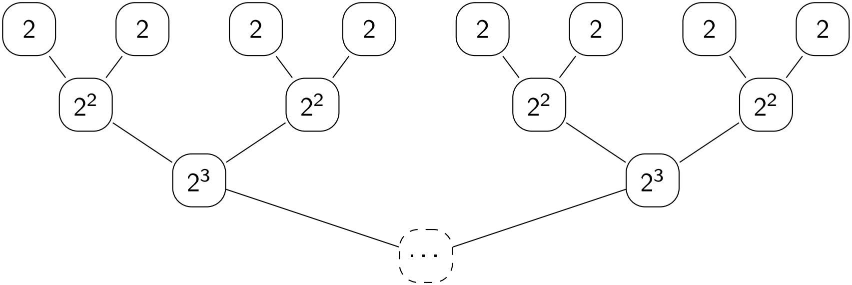
- Compute the M' corresponding pairs (i,j)
  - classical time $\max(M, M')$ by **classical merging**
- Apply to the state a transformation f from (i,j) to $[0; M'-1]$
  - quantum time $\mathcal{O}(M')$ without QRACM
- Return the state and the vector v with $v_{f(i,j)} = \ell_i+\ell'_j$:

$$|\psi_{(v_0,\ldots,v_{M'-1})}\rangle : \forall i, v_i < 2^{a-r}$$

Heuristic complexity

Here is a way to recover Kuperberg's heuristic $2^{\sqrt{2n}}$

- see this as a **merging tree** where each node is the list of labels of a qubit register
- start with lists of size 2
- merge the lists pairwise recursively as follows:



Heuristic complexity (ctd.)

- At each level, we merge two lists of size 2^i into a list of size 2^{i+1}: we eliminate $2i - (i+1) = i - 1$ bits.
- At the end we must have eliminated all the bits.
- Thus we need $k+1$ levels where $1 + 2 + \ldots + k = n \implies k \simeq \sqrt{2n}$
- Thus we have started with $2^{\sqrt{2n}}$ lists of size 2. Quantum time = classical time = classical memory = $2^{\sqrt{2n}}$

But many other trade-offs are possible! In particular, we can make the classical time much bigger than the quantum time.

Second trade-off

Before: 2-list merging

We merge two lists of size 2^i into a list of size 2^{i+1}, in time 2^{i+1}.

- On two levels we would gain: $i - 1 + i + 1 - 1 = 2i - 1$ bits

After: 4-list merging

We take 4 lists of size 2^i and merge into a list of size 2^{i+2}.

- We gain $4i - (i + 2) = 3i - 2$ bits
- This is much more, because we have bypassed a merging step
- The time complexity increases to 2^{2i}
- The memory complexity remains 2^i (Schroeppel-Shamir)

Thus we need $k + 1$ levels where

$$1 + 4 + \ldots + (3k - 2) = n \implies \frac{3}{4}k^2 \simeq n \implies k \simeq 2\sqrt{n/3}$$

$\implies 2^{2\sqrt{n/3}}$ quantum queries and quantum time; $2^{4\sqrt{n/3}}$ classical time.

The Oracle

Group action oracle vs. CSIDH action oracle

- We need an oracle to evaluate $f(x) = x * E$ for any $x \in \mathcal{C}\ell\mathcal{O}$
- In general, that's difficult (see previous talks)
- But we are given many small-degree isogenies $\mathfrak{l}_i$, such that $\{\prod \mathfrak{l}_i^{e_i}\}$ (are expected to) span the class group

The CSIDH action oracle

Compute $\prod \mathfrak{l}_i^{e_i} * E$ given the exponents e_i.

From x to an exponent sequence

Precomputation

- The class group structure, with Shor's algorithm
- An approximate short basis B of the **relation lattice** $\mathcal{L}$:

$$(e_1, \dots, e_u), \prod \mathfrak{l}_i^{e_i} = 1$$

A new x arrives (in superposition).

- Decompose x on the $\mathfrak{l}_i$ with Shor's algorithm (easy)
- We now have a vector: $(t_1, \dots, t_u)$ representing x, but the exponents can be high
- Find a close vector $(v_1, \dots, v_u)$ in the relation lattice using Babai's algorithm
- Now we have $x = \prod \mathfrak{l}_i^{v_i - t_i}$

The ugly heuristic

We'll have to compute $L_1(\bar{v} - \bar{t}) = \sum_i |v_i - t_i|$ isogenies.

- By Babai's algorithm, this is upper bounded by: $\frac{\sqrt{u}}{2}\sqrt{\sum_i \|b_i^*\|^2}$ where the b_i^* are the Gram-Schmidt orthogonalization of the short basis of $\mathcal{L}$.
- We didn't really know how to estimate that, so we took relation lattices of random elements in random cyclic groups
- In general, there must be some bound w.r.t. some standard heuristic

On average for CSIDH-512, we estimated 1300 isogenies (the legitimate CSIDH group action does 370).

Later on, the CSIDH-512 class group (cyclic) and relation lattices were computed by [BKV19].

Beullens, Kleinjung, Vercauteren, "CSI-FiSh: Efficient isogeny based signatures through class group computations", ASIACRYPT 2019

The CSIDH action oracle

Now, let's implement:

$$|e_1, \ldots, e_u\rangle |A\rangle |0\rangle \mapsto |e_1, \ldots, e_u\rangle |A\rangle |L_{\ell_1}^{e_1} \circ \ldots \circ L_{\ell_u}^{e_u}(A)\rangle \quad ,$$

where:

- $A \in \mathbb{F}_p$ represents a CSIDH Montgomery curve
- L_{ℓ_i} consists in applying $[\mathfrak{l}_i]$ (with negative exponents, it's the inverse)

We follow [BLMP20], which has all the algebraic details, and we make everything into **a quantum circuit**.

- In [BLMP20], you'll find "537503414" logical qubits for CSIDH-512
- Actually we get 40 000 qubits

Bernstein, Lange, Martindale, Panny, "Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies", EUROCRYPT 2020

Reducing to a single isogeny circuit

- Given A, we want to compute $B = L_\ell(A)$ for some ℓ
- The isogeny is invertible, so we'll make a circuit: $|A\rangle \mapsto |L_\ell(A)\rangle$
- No memory overhead for the sequence of 1300 isogenies

Finding a point of order ℓ

To compute the ℓ-isogenous curve, we need a point on E_A of order ℓ.

- In CSIDH: find $P \in E_A$, then compute $Q = ((p+1)/\ell)P$. If $Q \neq \infty$, then it generates a subgroup of order ℓ.
- This multiplication is the most costly part.

The simple method

- Sample $x \in \mathbb{F}_p^*$: the x-coordinate of a point P
- Check that $x^3 + Ax^2 + x$ is a square
- Compute $((p+1)/\ell)P$; if this is ∞, then repeat

BLMP show that it's difficult to have a good success prob. in constant-time.

Finding a point (ctd.)

The quantum method

The probability of success of the simple method is exactly: $\frac{1}{2}\left(1 - \frac{1}{\ell}\right)$.

- We do an exact quantum search of $x \in \mathbb{F}_p^*$ with a partial rotation
- This requires a controlled phase-operator, approximated by Soloway-Kitaev
- We get a failure probability 2^{-50} with a negligible cost overhead

---

Chi, Kim, "Quantum database search by a single query", NASA International Conference on Quantum Computing and Quantum Communications

Summary

- From A, obtain the good points P (detectable failures happen here)
- From A and P, compute $Q = ((p+1)/\ell)P$ (overhead w.r.t. classical circuit due to reversibility)
- From A and Q, obtain $B = L_\ell(A)$
- Uncompute Q
- Uncompute P (detectable failures happen here)

All of this is counted in multiplications in $\mathbb{F}_p$ (also more costly than classical).

Failures

- **All the failures are detected**: they do not impact the sieving step.
- A probability of failure of 2^{-50} is more than enough.

Numbers!

Bit-size n of p	Number of isog.	T_M	Toffoli gates	T-gates	Ancilla qubits
512	1300	2^{20}	$2^{49.6}$	$2^{52.4}$	$< 40\ 000$
1024	4000	2^{22}	$2^{56.2}$	$2^{59.0}$	$< 60\ 000$
1024	4000	2^{22}	$2^{54.8}$	$2^{57.6}$	$< 80\ 000$
1792	10 000	$2^{23.6}$	$2^{60.1}$	$2^{62.9}$	$< 110\ 000$
1792	10 000	$2^{23.6}$	$2^{58.9}$	$2^{61.7}$	$< 140\ 000$

All of this can be improved, for example:

- The "pebbling" game for reversibility (`https://algassert.com/post/1905`)
- The multiplication circuit
- Legendre symbol computations are actually easy

Concluding Remarks

Attacking CSIDH

For CSIDH-512:

- 2^{19} queries with Regev's algorithm (single subset-sum), less than 2^{128} classical time
- $2^{71.6}$ T-gates (about 1000 times less than Grover)
- $< 2^{15.3}$ qubits (also overestimated)

The circuit is **by far** an overestimation.

Safe instances

Up to interpretation. We proposed two sets of parameters for NIST-1.

Aggressive parameters

If NIST-1 is a classical time-memory product at 2^{128} and the oracle allows for 2^{20} queries, $p \simeq 2260$ bits would be enough.

Conservative parameters

If NIST-1 allows for 2^{128} classical time and 2^{64} classical memory and the oracle allows for 2^{40} queries, $p \simeq 5280$ bits would be enough.

Conclusion

- The quantum attacks have many degrees of freedom, which allows many trade-offs, using different approaches.
- Subexponential algorithms means safe instances are harder to estimate.
- The NIST levels are also tricky to work with, as they depend on time and not on the number of queries.

Thank you!