Quantum attacks against isogenies

Daniel J. Bernstein

1994 Shor discrete-log algorithm:
Input prime \(p\); \(g \in \mathbb{F}_p^*\); \(h \in g^\mathbb{Z}\).

Define \(\varphi : \mathbb{Z} \times \mathbb{Z} \to \mathbb{F}_p^*\) by
\(\varphi(a, b) = g^ah^b\). Fast function.

If \(h = g^s\) and \(g\) has order \(N\)
then \(\text{Ker} \varphi = \mathbb{Z}(N, 0) + \mathbb{Z}(s, -1)\).

Shor computes \(\varphi\) on quantum superposition of many \((a, b)\);
deduces \(\text{Ker} \varphi\); deduces \(s\) in \(\mathbb{Z}/N\).

Shor also generalizes from \(\mathbb{F}_p^*\) to other finite groups with fast computations.
e.g. \(\mathbb{F}_q^*\) for prime power \(q\);
\(E(\mathbb{F}_q)\) for elliptic curve \(E/\mathbb{F}_q\).
Quantum attacks against isogenies

Daniel J. Bernstein

1994 Shor discrete-log algorithm:
Input prime \( p \); \( g \in \mathbb{F}_p^* \); \( h \in g^\mathbb{Z} \).

Define \( \varphi : \mathbb{Z} \times \mathbb{Z} \to \mathbb{F}_p^* \) by \( \varphi(a, b) = g^a h^b \). Fast function.

If \( h = g^s \) and \( g \) has order \( N \) then \( \text{Ker } \varphi = \mathbb{Z}(N, 0) + \mathbb{Z}(s, -1) \).

Shor computes \( \varphi \) on quantum superposition of many \((a, b)\);
deduces \( \text{Ker } \varphi \); deduces \( s \) in \( \mathbb{Z}/N \).

Shor also generalizes from \( \mathbb{F}_p^* \) to other finite groups with fast computations.

1995 Boneh–Lipton:
Find “hidden” lattice \( L \subseteq \mathbb{Z}^n \),
given fast function \( \varphi : \mathbb{Z}^n \to X \) that induces \( \mathbb{Z}^n/L \hookrightarrow X \).

e.g. \( \mathbb{F}_q^* \) for prime power \( q \);
\( E(\mathbb{F}_q) \) for elliptic curve \( E/\mathbb{F}_q \).
Quantum attacks
against isogenies

Daniel J. Bernstein

1994 Shor discrete-log algorithm:
Input prime $p$; $g \in F_p^*$; $h \in g^Z$.

Define $\varphi : Z \times Z \to F_p^*$ by $\varphi(a, b) = g^a h^b$. Fast function.

If $h = g^s$ and $g$ has order $N$ then $\text{Ker} \varphi = Z(N, 0) + Z(s, -1)$.

Shor computes $\varphi$ on quantum superposition of many $(a, b)$; deduces $\text{Ker} \varphi$; deduces $s$ in $Z/N$.

Shor also generalizes from $F_p^*$ to other finite groups with fast computations. e.g. $F_q^*$ for prime power $q$; $E(F_q)$ for elliptic curve $E/F_q$.

1995 Boneh–Lipton:
Find “hidden” lattice $L \subseteq Z^n$, given fast function $\varphi : Z^n \to X$ that induces $Z^n/L \leftrightarrow X$.

Non-commutative generalizations:
e.g. find hidden subgroup $H \subseteq S_n$, given fast function $\varphi : S_n \to X$ that induces $S_n/H \leftrightarrow X$?

Some progress, some obstacles.
Shor also generalizes from $\mathbf{F}_p^*$ to other finite groups with fast computations. E.g. $\mathbf{F}_q^*$ for prime power $q$; $E(\mathbf{F}_q)$ for elliptic curve $E/\mathbf{F}_q$.

1995 Boneh–Lipton: Find “hidden” lattice $L \subseteq \mathbf{Z}^n$, given fast function $\varphi : \mathbf{Z}^n \rightarrow X$ that induces $\mathbf{Z}^n/L \leftrightarrow X$.

Non-commutative generalizations: E.g. find hidden subgroup $H \subseteq S_n$, given fast function $\varphi : S_n \rightarrow X$ that induces $S_n/H \leftrightarrow X$?

Some progress, some obstacles.

The hidden-shift problem
Given $N \in \mathbf{Z}$, $N > 0$; $f_0 : \mathbf{Z}/N \rightarrow X$; $f_1 : \mathbf{Z}/N \rightarrow X$; $f_1(a) = f_0(a + s)$ for all $a \in \mathbf{Z}/N$.

Goal: Find $s \in \mathbf{Z}/N$. 

For $\varphi = \mathbf{Z}(N, 0) + \mathbf{Z}(s, -1)$.

Computes $\varphi$ on quantum superposition of many $(a, b)$; $\mathbf{Ker} \varphi$; deduces $s$ in $\mathbf{Z}/N$. 

Quantum attacks against isogenies
Daniel J. Bernstein

1994 Shor discrete-log algorithm:
Input prime $p$; $g \in \mathbf{F}_p^*$; $h \in g^Z$. Define $'$ : $\mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{F}_p^*$ by $'= g^a h^b$. Fast function.

If $h = g^s$ and $g$ has order $N$ for $\varphi = \mathbf{Z}(N, 0) + \mathbf{Z}(s, -1)$.

Shor computes $'$ on quantum superposition of many $(a, b)$; $\mathbf{Ker} \varphi$; deduces $s$ in $\mathbf{Z}/N$. 

Shor also generalizes from $\mathbf{F}_p^*$ to other finite groups with fast computations.
E.g. $\mathbf{F}_q^*$ for prime power $q$; $E(\mathbf{F}_q)$ for elliptic curve $E/\mathbf{F}_q$.
Quantum attacks against isogenies
Daniel J. Bernstein
1994 Shor discrete-log algorithm:
Input prime \( p \); \( g \in \mathbb{F}_p^* \); \( h \in g^\mathbb{Z} \).
Define \( \cdot \) : \( \mathbb{Z} \times \mathbb{Z} \to \mathbb{F}_p^* \) by \( (a; b) = g^a h^b \).
Fast function.
If \( h = g^s \) and \( g \) has order \( N \) then \( \ker \cdot = \mathbb{Z}(N; 0) + \mathbb{Z}(s; -1) \).
Shor computes \( \cdot \) on quantum superposition of many \( (a; b) \);
deduces \( \ker \cdot \); deduces \( s \) in \( \mathbb{Z} = N \).

Shor also generalizes from \( \mathbb{F}_p^* \) to other finite groups with fast computations.
e.g. \( \mathbb{F}_q^* \) for prime power \( q \); \( E(\mathbb{F}_q) \) for elliptic curve \( E/\mathbb{F}_q \).
1995 Boneh–Lipton:
Find “hidden” lattice \( L \subseteq \mathbb{Z}^n \),
given fast function \( \varphi : \mathbb{Z}^n \to X \) that induces \( \mathbb{Z}^n/L \to X \).
Non-commutative generalizations:
e.g. find hidden subgroup \( H \subseteq S_n \),
given fast function \( \varphi : S_n \to X \) that induces \( S_n/H \to X \)?
Some progress, some obstacles.

The hidden-shift problem
Given \( N \in \mathbb{Z} \), \( N > 0 \);
\( f_0 : \mathbb{Z}/N \to X \); \( f_1 : \mathbb{Z}/N \to X \);
\( f_1(a) = f_0(a + s) \) for all \( a \in \mathbb{Z}/N \).
Goal: Find \( s \in \mathbb{Z}/N \).
Shor also generalizes from $\mathbb{F}_p^*$ to other finite groups with fast computations. 
e.g. $\mathbb{F}_q^*$ for prime power $q$; 
$E(\mathbb{F}_q)$ for elliptic curve $E/\mathbb{F}_q$.

1995 Boneh–Lipton:
Find “hidden” lattice $L \subseteq \mathbb{Z}^n$, 
given fast function $\varphi : \mathbb{Z}^n \rightarrow X$ 
that induces $\mathbb{Z}^n / L \hookrightarrow X$.

Non-commutative generalizations:
e.g. find hidden subgroup $H \subseteq S_n$, 
given fast function $\varphi : S_n \rightarrow X$ 
that induces $S_n / H \hookrightarrow X$?
Some progress, some obstacles.
Shor also generalizes from $\mathbb{F}_p^*$ to other finite groups with fast computations.
e.g. $\mathbb{F}_q^*$ for prime power $q$;
$E(\mathbb{F}_q)$ for elliptic curve $E/\mathbb{F}_q$.

1995 Boneh–Lipton:
Find “hidden” lattice $L \subseteq \mathbb{Z}^n$,
given fast function $\varphi : \mathbb{Z}^n \rightarrow X$
that induces $\mathbb{Z}^n/L \hookrightarrow X$.

Non-commutative generalizations:
e.g. find hidden subgroup $H \subseteq S_n$,
given fast function $\varphi : S_n \rightarrow X$
that induces $S_n/H \hookrightarrow X$?
Some progress, some obstacles.

The hidden-shift problem
Given $N \in \mathbb{Z}$, $N > 0$;
$f_0 : \mathbb{Z}/N \hookrightarrow X$; $f_1 : \mathbb{Z}/N \hookrightarrow X$;
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.
Goal: Find $s \in \mathbb{Z}/N$. 
Shor also generalizes from $\mathbb{F}_p^*$ to other finite groups with fast computations. e.g. $\mathbb{F}_q^*$ for prime power $q$; $E(\mathbb{F}_q)$ for elliptic curve $E/\mathbb{F}_q$.

1995 Boneh–Lipton:
Find “hidden” lattice $L \subseteq \mathbb{Z}^n$, given fast function $\varphi : \mathbb{Z}^n \rightarrow X$ that induces $\mathbb{Z}^n/L \hookrightarrow X$.

Non-commutative generalizations:
e.g. find hidden subgroup $H \subseteq S_n$, given fast function $\varphi : S_n \rightarrow X$ that induces $S_n/H \hookrightarrow X$?
Some progress, some obstacles.

The hidden-shift problem
Given $N \in \mathbb{Z}$, $N > 0$;
$f_0 : \mathbb{Z}/N \hookrightarrow X$; $f_1 : \mathbb{Z}/N \hookrightarrow X$;
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.
Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:
$(a, b)(c, d) = (a + (-1)^b c, b + d)$. \[\]
Shor also generalizes from $F_p^*$ to other finite groups with fast computations. e.g. $F_q^*$ for prime power $q$; $E(F_q)$ for elliptic curve $E/F_q$.

1995 Boneh–Lipton:
Find “hidden” lattice $L \subseteq \mathbb{Z}^n$, given fast function $\varphi : \mathbb{Z}^n \rightarrow X$ that induces $\mathbb{Z}^n/L \hookrightarrow X$.

Non-commutative generalizations:
e.g. find hidden subgroup $H \subseteq S_n$, given fast function $\varphi : S_n \rightarrow X$ that induces $S_n/H \hookrightarrow X$?
Some progress, some obstacles.

The hidden-shift problem
Given $N \in \mathbb{Z}$, $N > 0$; $f_0 : \mathbb{Z}/N \hookrightarrow X$; $f_1 : \mathbb{Z}/N \hookrightarrow X$; $f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$: $(a, b)(c, d) = (a + (-1)^b c, b + d)$.

Define $\varphi : D_N \rightarrow X$ by $\varphi(a, i) = f_i(a)$. Then $\varphi$ hides subgroup $\{(0, 0), (s, 1)\}$ of $D_N$. 

Shor also generalizes from $F_p^*$ to other finite groups with fast computations. 
e.g. $F_q^*$ for prime power $q$; $E(F_q)$ for elliptic curve $E/F_q$.

1995 Boneh–Lipton:
Find “hidden” lattice $L \subseteq \mathbb{Z}^n$, given fast function $\varphi : \mathbb{Z}^n \rightarrow X$ that induces $\mathbb{Z}^n/L \leftrightarrow X$.

Non-commutative generalizations: 
e.g. find hidden subgroup $H \subseteq S_n$, given fast function $\varphi : S_n \rightarrow X$ that induces $S_n/H \leftrightarrow X$? 
Some progress, some obstacles.

The hidden-shift problem

Given $N \in \mathbb{Z}$, $N > 0$;
$f_0 : \mathbb{Z}/N \leftrightarrow X$; $f_1 : \mathbb{Z}/N \leftrightarrow X$;
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:
$(a, b)(c, d) = (a + (-1)^b c, b + d)$.

Define $\varphi : D_N \rightarrow X$ by
$\varphi(a, i) = f_i(a)$. Then $\varphi$ hides subgroup $\{(0, 0), (s, 1)\}$ of $D_N$.

These are the only “Shor-hard” hidden subgroups of $D_N$.
Shor also generalizes from \( F^*_p \) to other finite groups with fast computations. For prime power \( q \); for elliptic curve \( E/F_q \).

Boneh–Lipton: Find "hidden" lattice \( L \subseteq \mathbb{Z}^n \), given fast function \( \varphi : \mathbb{Z}^n \rightarrow X \) reduces \( \mathbb{Z}^n/L \rightarrow X \).

Non-commutative generalizations: Find hidden subgroup \( H \subseteq S_n \), given fast function \( \varphi : S_n \rightarrow X \) reduces \( S_n/H \rightarrow X \)?

Some progress, some obstacles.

The hidden-shift problem

Given \( N \in \mathbb{Z}, N > 0; f_0 : \mathbb{Z}/N \rightarrow X; f_1 : \mathbb{Z}/N \rightarrow X; f_1(a) = f_0(a + s) \) for all \( a \in \mathbb{Z}/N \).

Goal: Find \( s \in \mathbb{Z}/N \).

Dihedral group \( D_N = \mathbb{Z}/N \times \mathbb{Z}/2 \): \((a, b)(c, d) = (a + (-1)^b c, b + d)\).

Define \( \varphi : D_N \rightarrow X \) by \( \varphi(a, i) = f_i(a) \). Then \( \varphi \) hides subgroup \( \{(0, 0), (s, 1)\} \) of \( D_N \).

These are the only "Shor-hard" hidden subgroups of \( D_N \).

1998 Ettinger–Høyer: Solve hidden-shift problem using \( O(\log N) \) quantum evaluations, huge \( \varphi \)-independent computation.
Shor also generalizes from $F^*_p$ to other finite groups with fast computations. e.g. $F^*_q$ for prime power $q$; curve $E/F_q$.

1995 Boneh–Lipton: Find "hidden" lattice $L \subseteq \mathbb{Z}^n$, given fast function $' : \mathbb{Z}^n \rightarrow X$ that induces $\mathbb{Z}^n = L \rightarrow X$.

Non-commutative generalizations: e.g. find hidden subgroup $H \subseteq S_n$, given fast function $' : S_n \rightarrow X$ that induces $S_n = H \rightarrow X$?

Some progress, some obstacles.

The hidden-shift problem

Given $N \in \mathbb{Z}$, $N > 0$;
$f_0 : \mathbb{Z}/N \rightarrow X$; $f_1 : \mathbb{Z}/N \rightarrow X$;
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:
$(a, b)(c, d) = (a + (−1)^b c, b + d)$.

Define $\varphi : D_N \rightarrow X$ by
$\varphi(a, i) = f_i(a)$. Then $\varphi$ hides subgroup $\{(0, 0), (s, 1)\}$ of $D_N$.

These are the only "Shor-hard" hidden subgroups of $D_N$.

The hidden-shift problem

Given $N \in \mathbb{Z}$, $N > 0$;

$f_0 : \mathbb{Z}/N \hookrightarrow X$; $f_1 : \mathbb{Z}/N \hookrightarrow X$;

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:

$(a, b)(c, d) = (a + (-1)^b c, b + d)$.

Define $\varphi : D_N \rightarrow X$ by

$\varphi(a, i) = f_i(a)$. Then $\varphi$ hides subgroup $\{(0, 0), (s, 1)\}$ of $D_N$.

These are the only “Shor-hard” hidden subgroups of $D_N$.

1998 Ettinger–Høyer:

Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.

1995 Boneh–Lipton:

Find “hidden” lattice $L \subseteq \mathbb{Z}^n$,
given fast function $\varphi : \mathbb{Z}^n \rightarrow X$ that induces $\mathbb{Z}^n = L$.

Non-commutative generalizations:

e.g. find hidden subgroup $H \subseteq S_n$,
given fast function $\varphi : S_n \rightarrow X$ that induces $S_n = H$.

Some progress, some obstacles.
The hidden-shift problem

Given $N \in \mathbb{Z}, N > 0$;
$f_0 : \mathbb{Z}/N \to X$; $f_1 : \mathbb{Z}/N \to X$;
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:
$(a, b)(c, d) = (a + (-1)^b c, b + d)$.

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then $\varphi$ hides
subgroup $\{(0, 0), (s, 1)\}$ of $D_N$.

These are the only “Shor-hard”
hidden subgroups of $D_N$.
The hidden-shift problem

Given $N \in \mathbb{Z}, N > 0$;

$f_0 : \mathbb{Z}/N \rightarrow X$; $f_1 : \mathbb{Z}/N \rightarrow X$;

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:

$$(a, b)(c, d) = (a + (-1)^b c, b + d).$$

Define $\varphi : D_N \rightarrow X$ by

$\varphi(a, i) = f_i(a)$. Then $\varphi$ hides subgroup $\{(0, 0), (s, 1)\}$ of $D_N$.

These are the only “Shor-hard” hidden subgroups of $D_N$.

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.

(1999–2004 Ettinger–Høyer–Knill:
Similarly few evaluations for hidden subgroups of any group.)
The hidden-shift problem

Given $N \in \mathbb{Z}, N > 0$;

$f_0 : \mathbb{Z}/N \leftrightarrow X; f_1 : \mathbb{Z}/N \leftrightarrow X; f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:

$(a, b)(c, d) = (a + (-1)^b c, b + d)$.

Define $\varphi : D_N \rightarrow X$ by

$\varphi(a, i) = f_i(a)$. Then $\varphi$ hides subgroup $\{(0, 0), (s, 1)\}$ of $D_N$.

These are the only “Shor-hard” hidden subgroups of $D_N$.

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.
The hidden-shift problem

Given \( N \in \mathbb{Z}, N > 0; \)
\( f_0 : \mathbb{Z}/N \rightarrow X; f_1 : \mathbb{Z}/N \rightarrow X; \)
\( f_1(a) = f_0(a + s) \) for all \( a \in \mathbb{Z}/N. \)

Goal: Find \( s \in \mathbb{Z}/N. \)

Dihedral group \( D_N = \mathbb{Z}/N \times \mathbb{Z}/2: \)
\( (a, b)(c, d) = (a + (-1)^b c, b + d). \)

Define \( \varphi : D_N \rightarrow X \) by
\( \varphi(a, i) = f_i(a). \) Then \( \varphi \) hides
subgroup \( \{(0, 0), (s, 1)\} \) of \( D_N. \)

These are the only “Shor-hard”
hidden subgroups of \( D_N. \)

1998 Ettinger–Høyer:
Solve hidden-shift problem using
\( O(\log N) \) quantum \( \varphi \) evaluations,
huge \( \varphi \)-independent computation.

(1999–2004 Ettinger–Høyer–Knill:
Similarly few evaluations for
hidden subgroups of any group.)

2003 Kuperberg:
Solve hidden-shift problem using
more quantum \( \varphi \) evaluations,
less \( \varphi \)-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.
The hidden-shift problem

Given \(N \in \mathbb{Z}, N > 0; f_0 : \mathbb{Z}/N \to X; f_1 : \mathbb{Z}/N \to X; f_0(a + s) \) for all \(a \in \mathbb{Z}/N\).

Find \(s \in \mathbb{Z}/N\).

Dihedral group \(D_N = \mathbb{Z}/N \times \mathbb{Z}/2\):

\[
(a; b)(c; d) = (a + (-1)^b c, b + d).
\]

Define \(\varphi : D_N \to X\) by

\(\varphi = f_i(a).\) Then \(\varphi\) hides \(\{0, 0\}, \{s, 1\}\) of \(D_N\).

These are the only “Shor-hard” hidden subgroups of \(D_N\).

1998 Ettinger–Høyer:
Solve hidden-shift problem using \(O(\log N)\) quantum \(\varphi\) evaluations,
huge \(\varphi\)-independent computation.

(1999–2004 Ettinger–Høyer–Knill:
Similarly few evaluations for hidden subgroups of any group.)

2003 Kuperberg:
Solve hidden-shift problem using more quantum \(\varphi\) evaluations,
less \(\varphi\)-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.

Attacking isogenies

CRS/CSIDH: Class group \(G\) acts freely and transitively
on a set \(X\) of curves over \(\mathbb{F}_p\).
The hidden-shift problem

Given $N \in \mathbb{Z}$, $N > 0$;

$f_0 : \mathbb{Z}/N \rightarrow \mathbb{X}$;

$f_1 : \mathbb{Z}/N \rightarrow \mathbb{X}$;

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Goal: Find $s \in \mathbb{Z}/N$.

Dihedral group $D_N = \mathbb{Z}/N \times \mathbb{Z}/2$:

$D_N = ( (-1)^{b} c, b + d)$.

Define $\varphi : D_N \rightarrow \mathbb{X}$ by

Then $\varphi$ hides subgroup $\{(0, 0); (s, 1)\}$ of $D_N$.

These are the only “Shor-hard” hidden subgroups of $D_N$.

1998 Ettinger–Høyer:

Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.

(1999–2004 Ettinger–Høyer–Knill:

Similarly few evaluations for hidden subgroups of any group.)

2003 Kuperberg:

Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.

2004 Regev, 2011 Kuperberg:

More tradeoffs, better tradeoffs.

Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.
The hidden-shift problem

Given \( N \in \mathbb{Z}, N > 0; \)

\( f_0 : \mathbb{Z} = N \rightarrow X; \)

\( f_1 : \mathbb{Z} = N \rightarrow X; \)

\( f_1(a) = f_0(a + s) \) for all \( a \in \mathbb{Z} = N. \)

Goal: Find \( s \in \mathbb{Z} = N. \)

Dihedral group \( D_N = \mathbb{Z} = N \times \mathbb{Z}/2: \)

\((a; b)(c; d) = (a + (−1)b d; b + d). \)

Define \( ' : D_N \rightarrow X \) by \( ' (a; i) = f_i(a). \)

Then \( ' \) hides subgroup \( \{(0; 0); (s; 1)\} \) of \( D_N. \)

These are the only “Shor-hard” hidden subgroups of \( D_N. \)

1998 Ettinger–Høyer:
Solve hidden-shift problem using \( O(\log N) \) quantum \( \varphi \) evaluations, huge \( \varphi \)-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum \( \varphi \) evaluations, less \( \varphi \)-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.

Attacking isogenies

CRS/CSIDH: Class group \( G \) acts freely and transitively on a set \( X \) of curves over \( \mathbb{F}_p. \)
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.


2003 Kuperberg: Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.


Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$. 


2003 Kuperberg: Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.


---

Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \simeq \mathbb{Z}/N$ with $N \approx p^{1/2}$. Compute $N$ by Shor’s algorithm.
1998 Ettinger–Høyer:
Solve hidden-shift problem using \(O(\log N)\) quantum \(\varphi\) evaluations, huge \(\varphi\)-independent computation.

(1999–2004 Ettinger–Høyer–Knill:
Similarly few evaluations for hidden subgroups of any group.)

2003 Kuperberg:
Solve hidden-shift problem using more quantum \(\varphi\) evaluations, less \(\varphi\)-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.

Attacking isogenies
CRS/CSIDH: Class group \(G\) acts freely and transitively on a set \(X\) of curves over \(\mathbb{F}_p\).

Usually \(G \cong \mathbb{Z}/N\) with \(N \approx p^{1/2}\). Compute \(N\) by Shor’s algorithm.
Find ideal \(I\) with \(G = [I]^\mathbb{Z}\).
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$. Compute $N$ by Shor’s algorithm. Find ideal $I$ with $G = [I]^\mathbb{Z}$.

Given $E_0, E_1 \in X$: define $f_0 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_0$; $f_1 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_1$.

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.
1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.

Attacking isogenies
CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.
Compute $N$ by Shor’s algorithm. Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define
$f_0 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_0$;
$f_1 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$. 

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.

Attacking isogenies
CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.
Compute $N$ by Shor’s algorithm.
Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_0$;
$f_1 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$. 
1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum $\varphi$ evaluations, huge $\varphi$-independent computation.


2003 Kuperberg:
Solve hidden-shift problem using more quantum $\varphi$ evaluations, less $\varphi$-independent computation.

2004 Regev, 2011 Kuperberg:
More tradeoffs, better tradeoffs.

Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$. Compute $N$ by Shor’s algorithm. Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define $f_0 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_0$; $f_1 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$. $f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$. Find the hidden shift $s$ in $f_0, f_1$. 
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Compute $N$ by Shor’s algorithm.

Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define

$$f_0 : \mathbb{Z}/N \leftrightarrow X \text{ by } a \mapsto [I]^a E_0;$$
$$f_1 : \mathbb{Z}/N \leftrightarrow X \text{ by } a \mapsto [I]^a E_1.$$ 

$$E_1 = [I]^s E_0 \text{ for some } s \in \mathbb{Z}/N.$$

$$f_1(a) = f_0(a + s) \text{ for all } a \in \mathbb{Z}/N.$$

Find the hidden shift $s$ in $f_0, f_1$. 

How many steps in an action?

Steps for CRS/CSIDH users:

Fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$. e.g., $d = 74$ for CSIDH-512.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Compute $N$ by Shor’s algorithm.

Find ideal $I$ with $G = [I]^\mathbb{Z}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \rightarrow X$ by $a \mapsto [I]^a E_0$;

$f_1 : \mathbb{Z}/N \rightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Find the hidden shift $s$ in $f_0, f_1$. 

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$. e.g., $d = 74$ for CSIDH-512.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$. Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$. Given $E_0, E_1 \in X$: define $f_0: \mathbb{Z}/N \to X$ by $a \mapsto [I^a E_0]$, $f_1: \mathbb{Z}/N \to X$ by $a \mapsto [I^a E_1]$. Find ideal $I$ with $G = [I]\mathbb{Z}$.

Steps for CRS/CSIDH users:

- fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.
- e.g., $d = 74$ for CSIDH-512.

How many steps in an action?

Steps for CRS/CSIDH users:

- fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.
- e.g., $d = 74$ for CSIDH-512.

How many steps in an action?

Steps for CRS/CSIDH users:

- fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.
- e.g., $d = 74$ for CSIDH-512.

How many steps in an action?

Steps for CRS/CSIDH users:

- fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.
- e.g., $d = 74$ for CSIDH-512.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \cong p^{1/2}$. Compute $N$ by Shor’s algorithm. Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define
- $f_0 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_0$;
- $f_1 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.
$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.
Find the hidden shift $s$ in $f_0, f_1$.

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$. e.g., $d = 74$ for CSIDH-512.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Compute $N$ by Shor's algorithm.

Find ideal $I$ with $G = [I]^\mathbb{Z}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \hookrightarrow X$ by $a \mapsto [I]^a E_0$;

$f_1 : \mathbb{Z}/N \hookrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Find the hidden shift $s$ in $f_0, f_1$.

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.

e.g., $d = 74$ for CSIDH-512.

Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Compute $N$ by Shor’s algorithm.

Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \hookrightarrow X$ by $a \mapsto [I]^a E_0$;

$f_1 : \mathbb{Z}/N \hookrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Find the hidden shift $s$ in $f_0, f_1$.

---

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.

e.g., $d = 74$ for CSIDH-512.


$[P_1]^{7038304916}$: 7038304916 steps.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Compute $N$ by Shor’s algorithm.

Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \hookrightarrow X$ by $a \mapsto [I]^a E_0$;

$f_1 : \mathbb{Z}/N \hookrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Find the hidden shift $s$ in $f_0, f_1$.

---

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.

e.g., $d = 74$ for CSIDH-512.


$[P_1]^{7038304916}$: 7038304916 steps.

$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Compute $N$ by Shor’s algorithm.

Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \rightarrow X$ by $a \mapsto [I]^a E_0$;

$f_1 : \mathbb{Z}/N \rightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.

Find the hidden shift $s$ in $f_0, f_1$.

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.

e.g., $d = 74$ for CSIDH-512.


$[P_1]^{7038304916}$: 7038304916 steps.

$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.

Approach 1: Compute lattice $L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})$. 
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.
Compute $N$ by Shor’s algorithm.
Find ideal $I$ with $G = [I]^{\mathbb{Z}}$.

Given $E_0, E_1 \in X$: define

$f_0 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_0$;
$f_1 : \mathbb{Z}/N \leftrightarrow X$ by $a \mapsto [I]^a E_1$.

$E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

$f_1(a) = f_0(a + s)$ for all $a \in \mathbb{Z}/N$.
Find the hidden shift $s$ in $f_0, f_1$.

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.

e.g., $d = 74$ for CSIDH-512.

$[P_1]^{7038304916}$: 7038304916 steps.

$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.

Approach 1: Compute lattice $L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})$.

Given $a \in \mathbb{Z}^d$, find close $v \in L$:

distance $\exp((\log N)^{1/2+o(1)})$ using time $\exp((\log N)^{1/2+o(1)})$. 
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$. Compute $N$ by Shor's algorithm.

Find ideal $I$ with $G = [I]$. 

Given $E_0, E_1 \in X$: define $f_0 : \mathbb{Z}/N \to X$ by $a \mapsto [I]^a E_0$; $f_1 : \mathbb{Z}/N \to X$ by $a \mapsto [I]^a E_1$.

$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.

Approach 1: Compute lattice $L = \ker(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})$.

Given $a \in \mathbb{Z}^d$, find close $v \in L$: distance $\exp((\log N)^{1/2+o(1)})$ using time $\exp((\log N)^{1/2+o(1)})$.

Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.
Attacking isogenies

CRS/CSIDH: Class group $G$ acts freely and transitively on a set $X$ of curves over $\mathbb{F}_p$.

Usually $G \cong \mathbb{Z}/N$ with $N \approx p^{1/2}$.

Find ideal $I$ with $G = [I]^Z$.

Given $E_0, E_1 \in X$, define $f_0: \mathbb{Z}/N \to X$ by $a \mapsto [I]^a E_0$;

$f_1: \mathbb{Z}/N \to X$ by $a \mapsto [I]^a E_1$.

Given $E_1 = [I]^s E_0$ for some $s \in \mathbb{Z}/N$.

Approach 1: Compute lattice $L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})$.

Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.

e.g., $d = 74$ for CSIDH-512.

$[P_1]^{5}[P_2]^{4}[P_3]^{1}$: 10 steps.

$[P_1]^{7038304916}$: 7038304916 steps.

$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.

Given $a \in \mathbb{Z}^d$, find close $v \in L$: distance $\exp((\log N)^{1/2+o(1)})$ using time $\exp((\log N)^{1/2+o(1)})$. 

How many steps in an action? Steps for CRS/CSIDH users: fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$. e.g., $d = 74$ for CSIDH-512.

$[P_1]^{5}[P_2]^{4}[P_3]^{1}$: 10 steps.

$[P_1]^{7038304916}$: 7038304916 steps.

$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.

Approach 1: Compute lattice $L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})$.

Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

Given $a \in \mathbb{Z}^d$, find close $v \in L$: distance $\exp((\log N)^{1/2+o(1)})$ using time $\exp((\log N)^{1/2+o(1)})$. 

Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.
How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small \([P_1], [P_2], [P_3], \ldots, [P_d]\).

e.g., \(d = 74\) for CSIDH-512.


\([P_1]^{7038304916}\): 7038304916 steps.

\([P_1]^a\) for huge \(a \in \mathbb{Z}/N\): Hmmm.

Approach 1: Compute lattice \(L = \ker(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})\).

Given \(a \in \mathbb{Z}^d\), find close \(v \in L\):

distance \(\exp((\log N)^{1/2 + o(1)})\)

using time \(\exp((\log N)^{1/2 + o(1)})\).

Approach 2: Increase \(d\) up to \(\exp((\log N)^{1/2 + o(1)})\). Search randomly for small relations.
How many steps in an action?

Steps for CRS/CSIDH users:
fast algorithms for actions of small \([P_1], [P_2], [P_3], \ldots, [P_d]\).
e.g., \(d = 74\) for CSIDH-512.

\([P_1]^{7038304916}\): 7038304916 steps.
\([P_1]^a\) for huge \(a \in \mathbb{Z}/N\): Hmmm.

Approach 1: Compute lattice \(L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})\).

Given \(a \in \mathbb{Z}^d\), find close \(v \in L\):
distance \(\exp((\log N)^{1/2+o(1)})\)
using time \(\exp((\log N)^{1/2+o(1)})\).

Approach 2: Increase \(d\) up to \(\exp((\log N)^{1/2+o(1)})\). Search randomly for small relations.
How many steps in an action?

Steps for CRS/CSIDH users:
fast algorithms for actions of small \([P_1], [P_2], [P_3], \ldots, [P_d]\).
e.g., \(d = 74\) for CSIDH-512.

\([P_1]^{7038304916}\): 7038304916 steps.
\([P_1]^a\) for huge \(a \in \mathbb{Z}/N\): Hmm.

Approach 1: Compute lattice \(L = \ker(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})\).

given \(a \in \mathbb{Z}^d\), find close \(v \in L\):
distance \(\exp((\log N)^{1/2+o(1)})\)
using time \(\exp((\log N)^{1/2+o(1)})\).

Approach 2: Increase \(d\) up to \(\exp((\log N)^{1/2+o(1)})\). Search randomly for small relations.

2010 Childs–Jao–Soukharev:
A. Time \(\exp((\log N)^{1/2+o(1)})\) to compute \(G\) action by Approach 2.
How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small \([P_1], [P_2], [P_3], \ldots, [P_d]\).
e.g., \(d = 74\) for CSIDH-512.

\([P_1]^{7038304916}\): 7038304916 steps.
\([P_1]^a\) for huge \(a \in \mathbb{Z}/N\): Hmmm.

Approach 1: Compute lattice \(L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})\).

Given \(a \in \mathbb{Z}^d\), find close \(v \in L\):
distance \(\exp((\log N)^{1/2+o(1)})\)
using time \(\exp((\log N)^{1/2+o(1)})\).

Approach 2: Increase \(d\) up to \(\exp((\log N)^{1/2+o(1)})\). Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time \(\exp((\log N)^{1/2+o(1)})\) to compute \(G\) action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.
How many steps in an action?

Steps for CRS/CSIDH users: fast algorithms for actions of small \([P_1], [P_2], [P_3], \ldots, [P_d]\).

e.g., \(d = 74\) for CSIDH-512.


\([P_1]^{7038304916}\): 7038304916 steps.

\([P_1]^a\) for huge \(a \in \mathbb{Z}/N\): Hmmm.

Approach 1: Compute lattice \(L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})\).

Given \(a \in \mathbb{Z}^d\), find close \(v \in L\): distance \(\exp((\log N)^{1/2+o(1)})\) using time \(\exp((\log N)^{1/2+o(1)})\).

Approach 2: Increase \(d\) up to \(\exp((\log N)^{1/2+o(1)})\). Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time \(\exp((\log N)^{1/2+o(1)})\) to compute \(G\) action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev): Time \(\exp((\log N)^{1/2+o(1)})\) to find \(g \in G\) with \(gE_0 = E_1\).
How many steps in an action?

Steps for CRS/CSIDH users:
fast algorithms for actions of small $[P_1], [P_2], [P_3], \ldots, [P_d]$.
e.g., $d = 74$ for CSIDH-512.

$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbb{Z}/N$: Hmmm.

Approach 1: Compute lattice $L = \text{Ker}(a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d})$.

Given $a \in \mathbb{Z}^d$, find close $v \in L$: distance $\exp((\log N)^{1/2+o(1)})$
using time $\exp((\log N)^{1/2+o(1)})$.

Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:
A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.
B. Unfixably flawed argument that Approach 2 beats Approach 1.
C. Apply Kuperberg (or Regev): $\text{Time } \exp((\log N)^{1/2+o(1)})$
to find $g \in G$ with $gE_0 = E_1$.
D. Proof assuming only GRH, using provable-factoring ideas.
How many steps in an action?

Steps for CRS/CSIDH users:
fast algorithms for actions of
small \([P_1], [P_2], [P_3], \ldots, [P_d]\).

\(d = 74\) for CSIDH-512.

\([P_1]^4[P_3]^1\): 10 steps.
\(7038304916\): 7038304916 steps.

For huge \(a \in \mathbb{Z}/N\): Hmm.

Approach 1: Compute lattice
\(L = \{a_1, \ldots, a_d \mapsto [P_1]^{a_1} \cdots [P_d]^{a_d}\}\).

Given \(a \in \mathbb{Z}^d\), find close \(v \in L:\)
\(\exp((\log N)^{1/2+\omega(1)})\)

Surely \(g = [P_1]^{a_1} \cdots [P_d]^{a_d}\) is
nearly uniformly distributed in \(G\).

Approach 2: Increase \(d\) up to
\(\exp((\log N)^{1/2+\omega(1)})\). Search
randomly for small relations.

2010 Childs–Jao–Soukharev:
A. Time \(\exp((\log N)^{1/2+\omega(1)})\) to
compute \(G\) action by Approach 2.

B. Unfixably flawed argument that
Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev):
Time \(\exp((\log N)^{1/2+\omega(1)})\)
to find \(g \in G\) with \(gE_0 = E_1\).

D. Proof assuming only GRH,
using provable-factoring ideas.

Approach 3 (mentioned in 2018
Bernstein–Lange–Martindale–
Panny): Uniform \((a_1, \ldots, a_d)\)
in \((-c, \ldots, c)^d\).

Choose \(c\) somewhat larger than users do.

Not much slowdown in action.
Steps for CRS/CSIDH users:

- Fast algorithms for actions of $\left[P_1\right], \ldots, \left[P_d\right]$.

For example, $d = 74$ for CSIDH-512.

- $\left[P_1\right]^{5}, \left[P_2\right]^{4}, \left[P_3\right]^{1}, \ldots, \left[P_d\right]$:
  - 10 steps.
- $\left[P_1\right]^{7038304916}$: 7038304916 steps.

Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev):
  - Time $\exp((\log N)^{1/2+o(1)})$ to find $g \in G$ with $gE_0 = E_1$.

D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny):

- Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$.

- Choose $c$ somewhat larger than users do.

- Not much slowdown in action.

- Surely $g = \left[P_1\right]^{a_1} \cdots \left[P_d\right]^{a_d}$ is nearly uniformly distributed in $G$. 

---

**Questions:**

- How many steps in an action?

**Answers:**

- Steps for CRS/CSIDH users:
  - Fast algorithms for actions of $\left[P_1\right], \ldots, \left[P_d\right]$.
Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev):
\[
\text{Time } \exp((\log N)^{1/2+o(1)})
\] to find $g \in G$ with $gE_0 = E_1$.

D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$. Choose $c$ somewhat larger than users do.

Not much slowdown in action.

Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ nearly uniformly distributed in $G$. 
Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev):
   Time $\exp((\log N)^{1/2+o(1)})$ to find $g \in G$ with $gE_0 = E_1$.

D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$. Choose $c$ somewhat larger than users do.

Not much slowdown in action. Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ is nearly uniformly distributed in $G$. 
Approach 2: Increase $d$ up to \( \exp((\log N)^{1/2+o(1)}) \). Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time \( \exp((\log N)^{1/2+o(1)}) \) to compute $G$ action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev): Time \( \exp((\log N)^{1/2+o(1)}) \) to find $g \in G$ with $gE_0 = E_1$.

D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform \((a_1, \ldots, a_d)\) in \(\{-c, \ldots, c\}^d\). Choose $c$ somewhat larger than users do.

Not much slowdown in action. Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ is nearly uniformly distributed in $G$.

Can quickly compute $gE_b$ and image of $g$ in $\mathbb{Z}/N$. 
Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev): Time $\exp((\log N)^{1/2+o(1)})$ to find $g \in G$ with $gE_0 = E_1$.

D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$. Choose $c$ somewhat larger than users do.

Not much slowdown in action. Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ is nearly uniformly distributed in $G$.

Can quickly compute $gE_b$ and image of $g$ in $\mathbb{Z}/N$.

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.
Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:

A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.

B. Unfixably flawed argument that Approach 2 beats Approach 1.

C. Apply Kuperberg (or Regev): $\exp((\log N)^{1/2+o(1)})$ to find $g \in G$ with $gE_0 = E_1$.

D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$. Choose $c$ somewhat larger than users do.

Not much slowdown in action. Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ is nearly uniformly distributed in $G$.

Can quickly compute $gE_b$ and image of $g$ in $\mathbb{Z}/N$.

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.

How fast are the steps? e.g. CSIDH-512, user distribution on $G$, error rate $< 2^{-32}$ (is this adequate?), nonlinear bit ops: $\approx 2^{51}$ by 2018 Jao–LeGrow–Leonardi–Ruiz-Lopez.
Approach 2: Increase $d$ up to $\exp((\log N)^{1/2+o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:
A. Time $\exp((\log N)^{1/2+o(1)})$ to compute $G$ action by Approach 2.
B. Unfixably flawed argument that Approach 2 beats Approach 1.
C. Apply Kuperberg (or Regev): Time $\exp((\log N)^{1/2+o(1)})$ to find $g \in G$ with $gE_0 = E_1$.
D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$. Choose $c$ somewhat larger than users do.

Not much slowdown in action. Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ is nearly uniformly distributed in $G$.

Can quickly compute $gE_b$ and image of $g$ in $\mathbb{Z}/N$.

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.

How fast are the steps? e.g. CSIDH-512, user distribution on $G$, error rate $< 2^{-32}$ (is this adequate?), nonlinear bit ops: $\approx 2^{51}$ by 2018 Jao–LeGrow–Leonardi–Ruiz-Lopez.
Approach 2: Increase $d$ up to $\exp((\log N)^{1 = 2+ o(1)})$. Search randomly for small relations.

2010 Childs–Jao–Soukharev:
A. Time $\exp((\log N)^{1 = 2+ o(1)})$ to compute $G$ action by Approach 2.
B. Unfixably flawed argument that Approach 2 beats Approach 1.
C. Apply Kuperberg (or Regev):
   Time $\exp((\log N)^{1 = 2+ o(1)})$ to find $g \in G$ with $gE_0 = E_1$.
D. Proof assuming only GRH, using provable-factoring ideas.

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny):
Uniform $(a_1, \ldots, a_d)$ in $\{-c, \ldots, c\}^d$. Choose $c$ somewhat larger than users do.

Not much slowdown in action.
Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$ is nearly uniformly distributed in $G$.

Can quickly compute $gE_b$ and image of $g$ in $\mathbb{Z}/N$.

Need more analysis of impact of these redundant representations upon Kuperberg's algorithm.

How fast are the steps?
e.g. CSIDH-512, user distribution on $G$, error rate $< 2^{-32}$ (is this adequate?), nonlinear bit ops $\approx 2^{51}$ by 2018 Jao–LeGrow–Leonardi–Ruiz-Lopez.
Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform \((a_1, \ldots, a_d)\) in \([-c, \ldots, c]^d\). Choose \(c\) somewhat larger than users do.

Not much slowdown in action. Surely \(g = [P_1]^{a_1} \cdots [P_d]^{a_d}\) is nearly uniformly distributed in \(G\).

Can quickly compute \(gE_b\) and image of \(g\) in \(\mathbb{Z}/N\).

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.

How fast are the steps?

e.g. CSIDH-512, user distribution on \(G\), error rate \(< 2^{-32}\) (is this adequate?), nonlinear bit ops:

Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform \((a_1, \ldots, a_d)\) in \([-c, \ldots, c]^d\). Choose \(c\) somewhat larger than users do.

Not much slowdown in action. Surely \(g = [P_1]^{a_1} \cdots [P_d]^{a_d}\) is nearly uniformly distributed in \(G\).

Can quickly compute \(gE_b\) and image of \(g\) in \(\mathbb{Z}/N\).

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.

How fast are the steps?

e.g. CSIDH-512, user distribution on \(G\), error rate < \(2^{-32}\) (is this adequate?), nonlinear bit ops: \(\approx 2^{51}\) by 2018 Jao–LeGrow–Leonardi–Ruiz-Lopez.

Many optimizations, detailed analysis: \(765325228976 \approx 0.7 \times 2^{40}\) by 2018 BLMP Algorithm 8.1.
Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform \((a_1, \ldots, a_d)\) in \(\{-c, \ldots, c\}^d\). Choose \(c\) somewhat larger than users do.

Not much slowdown in action.

Surely \(g = [P_1]^{a_1} \cdots [P_d]^{a_d}\) is nearly uniformly distributed in \(G\).

Can quickly compute \(gE_b\) and image of \(g\) in \(\mathbb{Z}/N\).

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.

How fast are the steps?

e.g. CSIDH-512, user distribution on \(G\), error rate \(<2^{-32}\) (is this adequate?), nonlinear bit ops:


Many optimizations, detailed analysis: \(765325228976 \approx 0.7 \cdot 2^{40}\) by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org:
full software and 56-page paper; variations in 512, distrib, \(2^{-32}\).
Approach 3 (mentioned in 2018 Bernstein–Lange–Martindale–Panny): Uniform \((a_1, \ldots, a_d)\) in \(\{-c, \ldots, c\}^d\). Choose \(c\) somewhat larger than users do.

Not much slowdown in action. Surely \(g = [P_1]^{a_1} \cdots [P_d]^{a_d}\) is nearly uniformly distributed in \(G\).

Can quickly compute \(gE_b\) and image of \(g\) in \(\mathbb{Z}/N\).

Need more analysis of impact of these redundant representations upon Kuperberg’s algorithm.

How fast are the steps?
e.g. CSIDH-512, user distribution on \(G\), error rate < \(2^{-32}\) (is this adequate?), nonlinear bit ops:

Many optimizations, detailed analysis: 765325228976 \(\approx 0.7 \cdot 2^{40}\) by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org: full software and 56-page paper; variations in 512, distrib, \(2^{-32}\).

Next big challenge: \(AT\) analysis.
How fast are the steps?

e.g. CSIDH-512, user distribution on $G$, error rate $< 2^{-32}$ (is this adequate?), nonlinear bit ops:


Many optimizations, detailed analysis: $765325228976 \approx 0.7 \cdot 2^{40}$

by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org:

full software and 56-page paper; variations in 512, distrib, $2^{-32}$.

Next big challenge: $AT$ analysis.
How fast are the steps?
e.g. CSIDH-512, user distribution on $G$, error rate $< 2^{-32}$ (is this adequate?), nonlinear bit ops:

Many optimizations, detailed analysis: $765325228976 \approx 0.7 \cdot 2^{40}$
by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org:
full software and 56-page paper;
variations in 512, distrib, $2^{-32}$.

Next big challenge: $AT$ analysis.
How fast are the steps?
e.g. CSIDH-512, user distribution on G, error rate $<2^{-32}$ (is this adequate?), nonlinear bit ops:

Many optimizations, detailed analysis: $765325228976 \approx 0.7 \cdot 2^{40}$ by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org: full software and 56-page paper; variations in 512, distrib, $2^{-32}$.

Next big challenge: $AT$ analysis.

How many actions + other costs?
2011 Kuperberg estimates “time” $\exp((0.98 \ldots + o(1))(\log_2 N))^{1/2}$, compares to 2003 Kuperberg: $\exp((1.23 \ldots + o(1))(\log_2 N))^{1/2}$.

2011 Bernstein–Lange–Martindale–Panny: Uniform $(a_1; \ldots; a_d)$ in $\{-c; \ldots; c\}^d$. Choose $c$ somewhat larger than users do. Not much slowdown in action. Surely $g = [P_1]a_1 \cdots [P_d]a_d$ is nearly uniformly distributed in $G$.
How fast are the steps?
e.g. CSIDH-512, user distribution on \( G \), error rate \(< 2^{-32}\) (is this adequate?), nonlinear bit ops:

Many optimizations, detailed analysis: \( 765325228976 \approx 0.7 \cdot 2^{40} \)
by 2018 BLMP Algorithm 8.1.

**quantum.isogenies.org:**
full software and 56-page paper;
variations in 512, distrib, \( 2^{-32} \).

Next big challenge: \( AT \) analysis.

How many actions + other costs?
2011 Kuperberg estimates “time”
\( \exp((0.98 \ldots + o(1))(\log_2 N)^{1/2}) \);
compares to 2003 Kuperberg:
\( \exp((1.23 \ldots + o(1))(\log_2 N)^{1/2}) \).
How fast are the steps?
e.g. CSIDH-512, user distribution on $G$, error rate $< 2^{-32}$ (is this adequate?), nonlinear bit ops:
Many optimizations, detailed analysis: $765325228976 \approx 0.7 \cdot 2^{40}$ by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org:
full software and 56-page paper; variations in 512, distrib, $2^{-32}$.

Next big challenge: $AT$ analysis.

How many actions + other costs?
2011 Kuperberg estimates “time”
$\exp((0.98 \ldots + o(1))(\log_2 N)^{1/2})$;
compares to 2003 Kuperberg:
$\exp((1.23 \ldots + o(1))(\log_2 N)^{1/2})$.
Open: Do better than $1/2$?
Do better than $0.98 \ldots$?
How fast are the steps?

e.g. CSIDH-512, user distribution on $G$, error rate $<2^{-32}$ (is this adequate?), nonlinear bit ops:


Many optimizations, detailed analysis: $765325228976 \approx 0.7 \cdot 2^{40}$ by 2018 BLMP Algorithm 8.1.

quantum.isogenies.org:
full software and 56-page paper;
variations in 512, distrib, $2^{-32}$.

Next big challenge: $AT$ analysis.

How many actions + other costs?

2011 Kuperberg estimates “time”

$\exp((0.98\ldots + o(1))(\log_2 N)^{1/2})$;

compares to 2003 Kuperberg:

$\exp((1.23\ldots + o(1))(\log_2 N)^{1/2})$.

Open: Do better than $1/2$?

Do better than $0.98\ldots$?

Exact number of actions? Some
work on analysis+optimization:

2003 Kuperberg; 2011 Kuperberg;
2018 Bonnetain–Naya-Plasencia;
2018 Bonnetain–Schrottenloher;
2019 Kuperberg; 2019 Peikert;
2019 Bonnetain–Schrottenloher.