Quantum attacks against isogenies

Daniel J. Bernstein

1994 Shor discrete-log algorithm:

Input prime p; $g \in \mathbf{F}_p^*$; $h \in g^{\mathbf{Z}}$.

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

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

Shor computes φ on quantum superposition of many (a, b); deduces $\operatorname{Ker} \varphi$; 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; $g \in \mathbf{F}_p^*$; $h \in g^{\mathbf{Z}}$.

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

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

Shor computes φ on quantum superposition of many (a, b); deduces $\operatorname{Ker}\varphi$; 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 $E/\mathbf{F}_q$.

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

Quantum attacks against isogenies

Daniel J. Bernstein

1994 Shor discrete-log algorithm:

Input prime p; $g \in \mathbf{F}_p^*$; $h \in g^{\mathbf{Z}}$.

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

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

Shor computes φ on quantum superposition of many (a, b); deduces $\operatorname{Ker} \varphi$; 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 $E/\mathbf{F}_q$.

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

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

m attacks
isogenies

. Bernstein

---

or discrete-log algorithm:

ime p; $g \in \mathbf{F}_p^*$; $h \in g^{\mathbf{Z}}$.

$\rho : \mathbf{Z} \times \mathbf{Z} \to \mathbf{F}_p^*$ by
$= g^a h^b$. Fast function.

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

mputes φ on quantum
sition of many (a, b);
Ker φ; 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 $E/\mathbf{F}_q$.

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

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

The hidd

Given N
$f_0 : \mathbf{Z}/N$
$f_1(a) =$

Goal: Fi

n

e-log algorithm:

$\in \mathbf{F}_p^*$; $h \in g^{\mathbf{Z}}$.

$\to \mathbf{F}_p^*$ by

ast function.

as order N

$, 0) + \mathbf{Z}(s, -1)$.

on quantum

any (a, b);

duces 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 $E/\mathbf{F}_q$.

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

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

The hidden-shift p

Given $N \in \mathbf{Z}$, $N >$
$f_0 : \mathbf{Z}/N \hookrightarrow X$; f_1
$f_1(a) = f_0(a + s)$ f

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

ithm:

$g^{\mathbf{Z}}$.

on.

/

, $-1)$.

m

);

$\mathbf{Z}/N$.

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 $E/\mathbf{F}_q$.

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

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

The hidden-shift problem

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

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

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 $E/\mathbf{F}_q$.

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

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

The hidden-shift problem

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

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

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 $E/\mathbf{F}_q$.

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

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

The hidden-shift problem

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

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

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

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 $\varphi : \mathbf{Z}^n \to X$
that induces $\mathbf{Z}^n/L \hookrightarrow X$.

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

The hidden-shift problem

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

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

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

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then φ hides
subgroup $\{(0, 0), (s, 1)\}$ of D_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 $E/\mathbf{F}_q$.

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

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

The hidden-shift problem

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

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

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

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

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

o generalizes
to other finite groups
t computations.
for prime power q;
or elliptic curve $E/\mathbf{F}_q$.

neh–Lipton:
dden" lattice $L \subseteq \mathbf{Z}^n$,
st function $\varphi : \mathbf{Z}^n \to X$
uces $\mathbf{Z}^n/L \hookrightarrow X$.

mmutative generalizations:
hidden subgroup $H \subseteq S_n$,
st function $\varphi : S_n \to X$
uces $S_n/H \hookrightarrow X$?
ogress, some obstacles.

The hidden-shift problem

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

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

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

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

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

1998 Ett
Solve hid
$O(\log N$
huge φ-i

zes

finite groups

tions.

power q;

curve $E/\mathbf{F}_q$.

n:

tice $L \subseteq \mathbf{Z}^n$,

n $\varphi : \mathbf{Z}^n \to X$

$\hookrightarrow X$.

generalizations:

ubgroup $H \subseteq S_n$,

n $\varphi : S_n \to X$

$H \hookrightarrow X$?

me obstacles.

The hidden-shift problem

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

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

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

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

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

1998 Ettinger–Høy

Solve hidden-shift

$O(\log N)$ quantum

huge φ-independe

ps

q.

^{n},

$\hookrightarrow X$

ations:

$\subseteq S_n$,

$\hookrightarrow X$

les.

The hidden-shift problem

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

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

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

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then φ 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 u
$O(\log N)$ quantum φ evalua
huge φ-independent comput

The hidden-shift problem

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

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

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

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then φ 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 φ evaluations,
huge φ-independent computation.

The hidden-shift problem

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

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

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

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then φ 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 φ evaluations,
huge φ-independent computation.

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

The hidden-shift problem

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

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

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

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then φ 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 φ evaluations,
huge φ-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 φ evaluations,
less φ-independent computation.

The hidden-shift problem

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

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

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

Define $\varphi : D_N \to X$ by
$\varphi(a, i) = f_i(a)$. Then φ 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 φ evaluations,
huge φ-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 φ evaluations,
less φ-independent computation.

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

den-shift problem

$\in \mathbf{Z}$, $N > 0$;

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

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

nd $s \in \mathbf{Z}/N$.

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

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

$: D_N \to X$ by

$= f_i(a)$. Then φ hides

$\{(0,0),(s,1)\}$ of D_N.

re the only "Shor-hard"

ubgroups of D_N.

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

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

Attackin

CRS/CS

acts free

on a set

problem

> 0;

: $\mathbf{Z}/N \hookrightarrow X$;

for all $a \in \mathbf{Z}/N$.

$/N$.

$_N = \mathbf{Z}/N \times \mathbf{Z}/2$:

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

X by

hen φ hides

$(s, 1)\}$ of D_N.

y “Shor-hard”

of D_N.

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

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

Attacking isogenie

CRS/CSIDH: Clas
acts freely and tra
on a set X of curv

X;
$\mathbf{Z}/N$.

$\times\ \mathbf{Z}/2$:
$b + d)$.

es
D_N.

rd”

1998 Ettinger–Høyer:
Solve hidden-shift problem using
$O(\log N)$ quantum φ evaluations,
huge φ-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 φ evaluations,
less φ-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 $\mathbf{F}_p$

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

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

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

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

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

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a + s)$ for all $a \in \mathbf{Z}/N$.

1998 Ettinger–Høyer:
Solve hidden-shift problem using $O(\log N)$ quantum φ evaluations, huge φ-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 φ evaluations, less φ-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 $\mathbf{F}_p$.

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

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

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

tinger–Høyer:
dden-shift problem using
) quantum φ evaluations,
independent computation.

004 Ettinger–Høyer–Knill:
few evaluations for
ubgroups of any group.)

perberg:
dden-shift problem using
antum φ evaluations,
dependent computation.

gev, 2011 Kuperberg:
adeoffs, better tradeoffs.

Attacking isogenies

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

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

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

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

How ma

Steps fo
fast algo
small $[P$
e.g., $d =$

yer:

problem using

φ evaluations,

nt computation.

ger–Høyer–Knill:

uations for

of any group.)

problem using

evaluations,

t computation.

Kuperberg:

etter tradeoffs.

Attacking isogenies

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

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

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

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

How many steps i

Steps for CRS/CS

fast algorithms fo

small $[P_1], [P_2], [P$

e.g., $d = 74$ for C

using
tions,
tation.
-Knill:
oup.)
using
s,
tion.
g:
offs.

Attacking isogenies

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

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

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

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

How many steps in an actio

Steps for CRS/CSIDH users
fast algorithms for actions o
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 $\mathbf{F}_p$.

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a + s)$ for all $a \in \mathbf{Z}/N$.
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 $\mathbf{F}_p$.

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a+s)$ for all $a \in \mathbf{Z}/N$.
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.

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

Attacking isogenies

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

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a+s)$ for all $a \in \mathbf{Z}/N$.
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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.

Attacking isogenies

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

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a+s)$ for all $a \in \mathbf{Z}/N$.
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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

Attacking isogenies

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

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a + s)$ for all $a \in \mathbf{Z}/N$.
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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

Attacking isogenies

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

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

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

$E_1 = [I]^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_1(a) = f_0(a + s)$ for all $a \in \mathbf{Z}/N$.
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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

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

g isogenies

IDH: Class group G
ely and transitively
X of curves over $\mathbf{F}_p$.

$G \cong \mathbf{Z}/N$ with $N \approx p^{1/2}$.
e N by Shor's algorithm.
al I with $G = [I]^{\mathbf{Z}}$.

$_0, E_1 \in X$: define
$\hookrightarrow X$ by $a \mapsto [I]^a E_0$;
$\hookrightarrow X$ by $a \mapsto [I]^a E_1$.

$^s E_0$ for some $s \in \mathbf{Z}/N$.
$f_0(a + s)$ for all $a \in \mathbf{Z}/N$.
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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

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

Approac
exp((log
randoml

s

s group G
nsitively
ves over $\mathbf{F}_p$.

with $N \approx p^{1/2}$.
or's algorithm.
$G = [I]^{\mathbf{Z}}$.

define
$a \mapsto [I]^a E_0$;
$a \mapsto [I]^a E_1$.

ome $s \in \mathbf{Z}/N$.
for all $a \in \mathbf{Z}/N$.
hift 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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

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

Approach 2: Incre
$\exp((\log N)^{1/2+o(1}$
randomly for smal

p.

$p^{1/2}$.

thm.

$_0$;

$_1$.

$/N$.

$\mathbf{Z}/N$.

, 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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

Given $a \in \mathbf{Z}^d$, 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 t
$\exp((\log N)^{1/2+o(1)})$. 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 $a \in \mathbf{Z}/N$: Hmmm.

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

Given $a \in \mathbf{Z}^d$, 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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

Given $a \in \mathbf{Z}^d$, 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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

Given $a \in \mathbf{Z}^d$, 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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

Given $a \in \mathbf{Z}^d$, 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.

$[P_1]^5[P_2]^4[P_3]^1$: 10 steps.
$[P_1]^{7038304916}$: 7038304916 steps.
$[P_1]^a$ for huge $a \in \mathbf{Z}/N$: Hmmm.

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

Given $a \in \mathbf{Z}^d$, 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 $gE_0 = E_1$.

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

ny steps in an action?

r CRS/CSIDH users:
orithms for actions of
$_1], [P_2], [P_3], \ldots, [P_d]$.
$= 74$ for CSIDH-512.

$]^4[P_3]^1$: 10 steps.
^{304916}: 7038304916 steps.
huge $a \in \mathbf{Z}/N$: Hmmm.

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

$\in \mathbf{Z}^d$, find close $v \in L$:
$\exp((\log N)^{1/2+o(1)})$
me $\exp((\log N)^{1/2+o(1)})$.

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 $gE_0 = E_1$.

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

Approac
Bernstei
Panny):
in $\{-c,$
somewha

Not muc
Surely g
nearly u

n an action?

IDH users:
actions of
$_3], \ldots, [P_d]$.
SIDH-512.

0 steps.
38304916 steps.
$\mathbf{Z}/N$: Hmmm.

pute lattice $L =$
$[P_1]^{a_1} \cdots [P_d]^{a_d}$).

l close $v \in L$:
$N)^{1/2+o(1)}$)
g $N)^{1/2+o(1)}$).

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 $gE_0 = E_1$.

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

Approach 3 (ment
Bernstein–Lange–
Panny): Uniform
in $\{-c, \ldots, c\}^d$.
somewhat larger t

Not much slowdow
Surely $g = [P_1]^{a_1}$
nearly uniformly d

n?

:

f

].

steps.

mm.

e $L =$

$[P_d]^{a_d})$.

L:

)

$(1))$.

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 $gE_0 = E_1$.

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

Approach 3 (mentioned in 2

Bernstein–Lange–Martindale

Panny): Uniform $(a_1, \ldots, a_d$

in $\{-c, \ldots, c\}^d$. Choose c

somewhat larger than users

Not much slowdown in actio

Surely $g = [P_1]^{a_1} \cdots [P_d]^{a_d}$

nearly uniformly distributed

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 $\mathbf{Z}/N$.

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 $\mathbf{Z}/N$.

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

h 2: Increase d up to
$N)^{1/2+o(1)}$). Search
y for small relations.

ilds–Jao–Soukharev:

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

ably flawed argument that
h 2 beats Approach 1.

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

f assuming only GRH,
ovable-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 $\mathbf{Z}/N$.

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

How fas

e.g. CSI
on G, er
adequat

$\approx 2^{51}$ by
Leonard

ase d up to
$1)$). Search
l relations.

Soukharev:

$N)^{1/2+o(1)}$) to
by Approach 2.

d argument that
Approach 1.

rg (or Regev):
$1/2+o(1)$)
n $gE_0 = E_1$.

g only GRH,
toring 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 $\mathbf{Z}/N$.

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

How fast are the s

e.g. CSIDH-512, u
on G, error rate $<$
adequate?), nonlir

$\approx 2^{51}$ by 2018 Jao
Leonardi–Ruiz-Lop

to
n
.
:
) to
ach 2.
nt that
1.
ev):
.
$_1$.
H,
as.

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 $\mathbf{Z}/N$.

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

How fast are the steps?

e.g. CSIDH-512, user distrib
on G, error rate $<2^{-32}$ (is t
adequate?), nonlinear bit op

$\approx 2^{51}$ 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 $\mathbf{Z}/N$.

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 $\mathbf{Z}/N$.

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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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 $\mathbf{Z}/N$.

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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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 $\mathbf{Z}/N$.

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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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.

h 3 (mentioned in 2018
n–Lange–Martindale–
Uniform $(a_1, \ldots, a_d)$
$\ldots, c\}^d$. Choose c
at larger than users do.

ch slowdown in action.
$= [P_1]^{a_1} \cdots [P_d]^{a_d}$ is
niformly distributed in G.

ckly compute gE_b
ge of g in $\mathbf{Z}/N$.

ore analysis of impact of
dundant representations
uperberg'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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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 ma

2011 Ku
exp((0.9
compare
exp((1.2

ioned in 2018
Martindale–
$(a_1, \ldots, a_d)$
Choose c
han users do.

wn in action.
$\cdots [P_d]^{a_d}$ is
istributed in G.

ute gE_b
$\mathbf{Z}/N$.

s of impact of
epresentations
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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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

2011 Kuperberg es
$\exp((0.98\ldots + o($
compares to 2003
$\exp((1.23\ldots + o($

018
e–
$_d$)

do.

on.
is
in G.

ct of
ions
.

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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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

2011 Kuperberg estimates "
$\exp((0.98\ldots + o(1))(\log_2 N$
compares to 2003 Kuperberg
$\exp((1.23\ldots + o(1))(\log_2 N$

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: $765325228976 \approx 0.7 \cdot 2^{40}$ 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:

$\approx 2^{51}$ by 2018 Jao–LeGrow–Leonardi–Ruiz-Lopez.

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:

$\approx 2^{51}$ by 2018 Jao–LeGrow–Leonardi–Ruiz-Lopez.

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.