S-unit attacks

Daniel J. Bernstein
University of Illinois at Chicago; Ruhr University Bochum

Includes new joint work with Kirsten Eisenträger, Tanja Lange, Karl Rubin, Alice Silverberg, and Christine van Vredendaal.

Builds upon vast previous literature; see upcoming paper for credits.
Algebraic geometry: the line over $\mathbb{C}$

$$f = x^4 + 6x^3 + 5x^2 = (x + 1)^1(x + 5)^1x^2 \in \mathbb{C}[x]:$$

$$f(10) = f \mod x - 10 = 16500 \quad \text{ord}_{10} f = 0$$
$$f(-1) = f \mod x + 1 = 0 \quad \text{ord}_{-1} f = 1$$
$$f(-5) = f \mod x + 5 = 0 \quad \text{ord}_{-5} f = 1$$
$$f(0) = f \mod x - 0 = 0 \quad \text{ord}_0 f = 2$$

... and consider $\mathbb{C}[1/x] \subset \mathbb{C}(x):$  \text{ord}_\infty f = -4

"ord}_r f" = x - r exponent in f. "ord}_\infty" = -deg.

This $f$ is an "S-unit" if \{\infty, 0, -1, -5\} $\subseteq S$.

Fundamental thm of algebra: $\sum_{\rho \in \mathbb{C} \cup \{\infty\}} \text{ord}_\rho f = 0.$

$f$ is almost determined by the vector $\rho \mapsto \text{ord}_\rho f.$
Intermediate: the line over $\mathbb{F}_7$

\[ f = x^4 + 3x^3 + x^2 + 5x + 2 = (x - 2)^2(x^2 - 3)^1 \in \mathbb{F}_7[x]: \]

\[
\begin{align*}
    f & \mod x - 0 = 2 & \quad \text{ord}_x f &= 0 & \quad |f|_x &= 1 \\
    f & \mod x - 2 = 0 & \quad \text{ord}_{x-2} f &= 2 & \quad |f|_{x-2} &= 1/7^2 \\
    f & \mod x^2 + 1 \neq 0 & \quad \text{ord}_{x^2+1} f &= 0 & \quad |f|_{x^2+1} &= 1 \\
    f & \mod x^2 - 3 = 0 & \quad \text{ord}_{x^2-3} f &= 1 & \quad |f|_{x^2-3} &= 1/7^2 \\
    \text{ord}_\infty f &= -4 & \quad |f|_\infty &= 7^4
\end{align*}
\]

\[ |f|_P = 1/\#(\mathbb{F}_7[x]/P)^{\text{ord}_P f} \text{ for "finite place" } P. \]

"Product formula": \[ \prod_\rho |f|_\rho = 1; \sum_\rho \log |f|_\rho = 0; \]

here $\rho$ ranges over \{monic irreducible in $\mathbb{F}_7[x]\} \cup \{\infty\}.

$f$ is almost determined by the vector $\rho \mapsto \text{ord}_\rho f$. 

Daniel J. Bernstein

S-unit attacks
Number theory: \( \mathbb{Z} \)

\[
f = -50421 = -3^17^5 \in \mathbb{Z}:
\]

\[
\begin{align*}
 f \mod 2 &= 1 \quad \text{ord}_2 f = 0 \quad |f|_2 = 1 \\
 f \mod 3 &= 0 \quad \text{ord}_3 f = 1 \quad |f|_3 = 1/3^1 \\
 f \mod 5 &= 4 \quad \text{ord}_5 f = 0 \quad |f|_5 = 1 \\
 f \mod 7 &= 0 \quad \text{ord}_7 f = 5 \quad |f|_7 = 1/7^5 \\
\end{align*}
\]

\[
|f|_\infty = 50421
\]

\[
|f|_P = 1/\#(\mathbb{Z}/P)^{\text{ord}_P f} \text{ for “finite place” } P.
\]

“Product formula”: \( \prod |f|_\rho = 1; \sum \log |f|_\rho = 0; \)

here \( \rho \) ranges over \( \{\text{prime numbers}\} \cup \{\infty\}. \)

\( f \) is almost determined by the vector \( \rho \mapsto \text{ord}_\rho f. \)
Lattice-based cryptography

2010 LPR proved “very strong hardness guarantees”:

Assume “worst-case problems on ideal lattices are hard for polynomial-time quantum algorithms”

“the ring-LWE distribution is pseudorandom”

security for a “truly practical lattice-based public-key cryptosystem”

Concrete parameters in cryptosystems are chosen assuming much more than polynomial hardness.
What’s the supposedly hard problem?

Parameters: Define $R = \mathbb{Z}[x]/(x^n + 1)$ for some $n \in \{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, \ldots \}$. [Can generalize, but this talk focuses on these rings $R$.]

Problem: Given a nonzero ideal $I \subseteq R$, find a “short” nonzero element $g \in I$.

“Given” $I$: given $v_1, v_2, \ldots, v_n \in R$ such that $I = \mathbb{Z}v_1 + \mathbb{Z}v_2 + \cdots + \mathbb{Z}v_n$.

E.g. $v_1 = x^3 + 817$ $\rightarrow$ $g = 2v_1 + 3v_2 - 5v_3 - 2v_4$
$v_2 = x^2 + 540$ $= 2x^3 + 3x^2 - 5x + 1$
$v_3 = x + 247$
$v_4 = 1009$
Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
817 & 0 & 0 & 1 \\
540 & 0 & 1 & 0 \\
247 & 1 & 0 & 0 \\
1009 & 0 & 0 & 0 \\
\end{array}
\]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{pmatrix}
817 & 0 & 0 & 1 \\
540 & 0 & 1 & 0 \\
247 & 1 & 0 & 0 \\
192 & 0 & 0 & -1 \\
\end{pmatrix}
\]

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( \mathbb{F}_7 \) \( \mod n \): naive algorithm gives shortest basis in poly time.]

Daniel J. Bernstein

S-unit attacks
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{align*}
277 & \quad 0 & \quad -1 & \quad 1 \\
540 & \quad 0 & \quad 1 & \quad 0 \\
247 & \quad 1 & \quad 0 & \quad 0 \\
192 & \quad 0 & \quad 0 & \quad -1 \\
\end{align*}
\]

But this doesn’t reach “short” when \( n \) is large.

(This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( F_7 \) [\( y \)]: naive algorithm gives shortest basis in poly time.)

Daniel J. Bernstein

\( S \)-unit attacks
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{align*}
277 & \quad 0 & \quad -1 & \quad 1 \\
263 & \quad 0 & \quad 2 & \quad -1 \\
247 & \quad 1 & \quad 0 & \quad 0 \\
192 & \quad 0 & \quad 0 & \quad -1
\end{align*}
\]

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( \mathbb{F}_7 \) \( y \): naive algorithm gives shortest basis in poly time.]

Daniel J. Bernstein
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0</td>
<td>−3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>263</td>
<td>0</td>
<td>2</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>247</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td></td>
</tr>
</tbody>
</table>

But this doesn’t reach “short” when $n$ is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of $F[q]$: naive algorithm gives shortest basis in poly time.]

Daniel J. Bernstein

S-unit attacks
Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
14 & 0 & -3 & 2 \\
16 & -1 & 2 & -1 \\
247 & 1 & 0 & 0 \\
192 & 0 & 0 & -1 \\
\end{array}
\]
Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0</td>
<td>−3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>−1</td>
<td>2</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td></td>
</tr>
</tbody>
</table>
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{align*}
14 & & 0 & & -3 & & 2 \\
16 & & -1 & & 2 & & -1 \\
55 & & 1 & & 0 & & 1 \\
137 & & -1 & & 0 & & -2 \\
\end{align*}
\]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
14 & 0 & -3 & 2 \\
-3 & 2 & -1 & 1 \\
55 & 1 & 0 & -3 \\
82 & -2 & 0 & -3 \\
\end{array}
\]

But this doesn’t reach “short” when \(n\) is large.

\[\text{[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \(F\) \(\mathbb{Q}^n\): naive algorithm gives shortest basis in poly time.]}\]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
14 & 0 & -3 & 2 \\
16 & -1 & 2 & -1 \\
55 & 1 & 0 & 1 \\
27 & -3 & 0 & -4 \\
\end{array}
\]

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( \mathbb{F}_7 \): naive algorithm gives shortest basis in poly time.]

Daniel J. Bernstein
Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0</td>
<td>−3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>−1</td>
<td>2</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>−3</td>
<td>0</td>
<td>−4</td>
<td></td>
</tr>
</tbody>
</table>
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{align*}
14 & \quad 0 & -3 & \quad 2 \\
16 & \quad -1 & 2 & \quad -1 \\
1 & \quad 7 & 0 & \quad 9 \\
27 & \quad -3 & 0 & \quad -4
\end{align*}
\]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0</td>
<td>−3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>−1</td>
<td>2</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>−2</td>
<td>−2</td>
<td>−3</td>
<td></td>
</tr>
</tbody>
</table>
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0</td>
<td>−3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>−1</td>
<td>5</td>
<td>−3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>−2</td>
<td>−2</td>
<td>−3</td>
<td></td>
</tr>
</tbody>
</table>

But this doesn’t reach “short” when \( n \) is large.

[this difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( F \).]

Daniel J. Bernstein
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
1 & 7 & 0 & 9 \\
11 & -2 & -2 & -3 \\
\end{array}
\]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
1 & 7 & 0 & 9 \\
9 & -1 & -7 & 0
\end{array}
\]

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( F \) \(^n \): naive algorithm gives shortest basis in poly time.]

Daniel J. Bernstein

S-unit attacks
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
-2 & 5 & 1 & 4 \\
9 & -1 & -7 & 0 \\
\end{array}
\]

But this doesn’t reach “short” when \( n \) is large.

(This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( \mathbb{F}_7 \).)

Daniel J. Bernstein

S-unit attacks
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
-2 & 5 & 1 & 4 \\
6 & -3 & -6 & -5 \\
\end{array}
\]

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( \mathbb{F}_7 \): naive algorithm gives shortest basis in poly time.]
Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
-2 & 5 & 1 & 4 \\
4 & 2 & -5 & -1 \\
\end{array}
\]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>−1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>−1</td>
<td>5</td>
<td>−3</td>
<td></td>
</tr>
<tr>
<td>−5</td>
<td>3</td>
<td>2</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>−5</td>
<td>−1</td>
<td></td>
</tr>
</tbody>
</table>

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( F^n \): naive algorithm gives shortest basis in poly time.]
Doesn’t look so hard . . .

Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

$$\begin{array}{cccccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
-5 & 3 & 2 & -1 \\
-1 & 5 & -3 & -2 \\
\end{array}$$

But this doesn’t reach “short” when \(n\) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \(F_n\): naive algorithm gives shortest basis in poly time.]
Naive lattice-basis reduction: Reduce largest row by subtracting closest multiple of another row.

\[
\begin{array}{cccc}
3 & 2 & -1 & 5 \\
2 & -1 & 5 & -3 \\
-5 & 3 & 2 & -1 \\
-1 & 5 & -3 & -2 \\
\end{array}
\]

But this doesn’t reach “short” when \( n \) is large.

[This difficulty is only for number theory, not geometry. Analogous short-vector problem for sublattice of \( \mathbb{F}_7[y]^n \): naive algorithm gives shortest basis in poly time.]
Fig. 5: Quality of Quantum Ideal-SVP vs. LLL and BKZ.

1.035
1.030
1.025
1.020
1.015
1.010
1.005
1.000

LLL

BKZ80
BKZ120
BKZ160
BKZ300

Naive algorithm, $p = 2m + 1$.
Naive algorithm, $p = m \ln m$.
HeuristicCVP, $p = 2m + 1$.
HeuristicCVP, $p = m \ln m$.
Lower bound, $p = 2m + 1$.
Halved lower bound (Remark 5) $p = 2m + 1$. 

Daniel J. Bernstein

S-unit attacks
How well the algorithms do

Given nonzero ideal \( I \subseteq R = \mathbb{Z}[x]/(x^n + 1) \), algorithm finds nonzero \( g = g_0 + \cdots + g_{n-1}x^{n-1} \in I \) with \( (g_0^2 + \cdots + g_{n-1}^2)^{1/2} = \eta \cdot (\#(R/I))^{1/n} \).

Algorithms using only additive structure of \( I \):

- LLL (fast): \( \eta^{1/n} \approx 1.022 \).
- BKZ-80 (not hard): \( \eta^{1/n} \approx 1.010 \).
- BKZ-160 (public attack): \( \eta^{1/n} \approx 1.007 \).
- BKZ-300 (large-scale attack): \( \eta^{1/n} \approx 1.005 \).

Algorithms also using multiplicative structure of \( R \):

blue/red curves; \( \eta \in 2^{n^{1/2+o(1)}} \) but worse \( \eta \) than LLL below “rank 1000”. Thin curves: “lower bound”. 
Major research directions

Many papers analyzing+optimizing BKZ-\(\beta\): e.g.,

- Last century: \(\exp(\Theta(\beta \log \beta))\) ops.
- 2001: \(\exp((0.415 \ldots + o(1))\beta)\) ops.
- 2015: \(\exp((0.292 \ldots + o(1))\beta)\) ops.
- 2015: \(\exp((0.265 \ldots + o(1))\beta)\) quantum ops.
- 2021: \(\exp((0.257 \ldots + o(1))\beta)\) quantum ops.
- Many more speedups hidden inside the \(o(1)\).

This talk focuses on multiplicative attacks:

- Part 2 of talk: How multiplicative attacks work.
- Part 3 of talk: Better multiplicative attacks.
Part 2
How multiplicative attacks work
Infinite places of $K = \mathbb{Q}[x]/(x^n + 1)$

Define $\zeta_m = \exp(2\pi i / m) \in \mathbb{C}$ for nonzero $m \in \mathbb{Z}$.

For any $c \in 1 + 2\mathbb{Z}$ have $(\zeta_{2^n}^c)^n + 1 = 0$ so there is a unique ring morphism $\nu_c : K \to \mathbb{C}$ taking $x$ to $\zeta_{2^n}^c$.

All $x^n + 1$ roots in $\mathbb{C}$: $\zeta_{2^n}^{1}, \ldots, \zeta_{2^n}^{n-1}, \zeta_{2^n}^{-(n-1)}, \ldots, \zeta_{2^n}^{1}$.

All $\nu : K \to \mathbb{C}$: $\nu_1, \ldots, \nu_{n-1}, \nu_{-(n-1)}, \ldots, \nu_{-1}$.

Define $|g|_c = |\nu_c(g)|^2 = \nu_c(g)\nu_{-c}(g)$.

The maps $g \mapsto |g|_c$ are the infinite places of $K$.

All places: $g \mapsto |g|_1, g \mapsto |g|_3, \ldots, g \mapsto |g|_{n-1}$.

Same as: $g \mapsto |g|_{-1}, g \mapsto |g|_{-3}, \ldots, g \mapsto |g|_{-n-1}$.

$$\sum_{c \in \{1, 3, \ldots, n-1\}} |g_0 + \cdots + g_{n-1}x^{n-1}|_c = \frac{n}{2}(g_0^2 + \cdots + g_{n-1}^2).$$
Finite places of $K = \mathbb{Q}[x]/(x^n + 1)$

Nonzero ideals of $R$ factor into prime ideals.

For each nonzero prime ideal $P$ of $R$, define $|g|_P = \#(R/P)^{-\text{ord}_P g}$. “Norm of $P$” is $\#(R/P)$.

The maps $g \mapsto |g|_P$ are the finite places of $K$.

For each prime number $p$: Factor $x^n + 1$ in $\mathbb{F}_p[x]$ to see the prime ideals of $R$ containing $p$.

e.g. $p = 2$: Prime ideal $2R + (x + 1)R = (x + 1)R$.

e.g. “unramified degree-1 primes”: $p \in 1 + 2n\mathbb{Z} \Rightarrow$ exactly $n$ $n$th roots $r_1, \ldots, r_n$ of $-1$ in $\mathbb{F}_p$.

$x^n + 1 = (x - r_1)(x - r_2)\ldots(x - r_n)$ in $\mathbb{F}_p[x]$.

Prime ideals $pR + (x - r_1)R, \ldots, pR + (x - r_n)R$. 
Example: $n = 4; \ R = \mathbb{Z}[x]/(x^4 + 1)$

$$g = g_0 + g_1 x + g_2 x^2 + g_3 x^3, \quad \zeta_8 = \exp(2\pi i/8):$$

$$\nu_{-1}(g) = g_0 + g_1 \zeta_8^{-1} + g_2 \zeta_8^{-2} + g_3 \zeta_8^{-3};$$

$$\nu_1(g) = g_0 + g_1 \zeta_8 + g_2 \zeta_8^2 + g_3 \zeta_8^3; \quad |g|_1 = |\nu_1(g)|^2.$$  

$$\nu_{-3}(g) = g_0 + g_1 \zeta_8^{-3} + g_2 \zeta_8^{-6} + g_3 \zeta_8^{-9};$$

$$\nu_3(g) = g_0 + g_1 \zeta_8^3 + g_2 \zeta_8^6 + g_3 \zeta_8^9; \quad |g|_3 = |\nu_3(g)|^2.$$  

$$P_{17, 2} = 17R + (x - 2)R: \quad |g|_{17, 2} = 17^{-\text{ord}_{P_{17, 2}} g}.$$  

$$P_{17, 8} = 17R + (x - 8)R: \quad |g|_{17, 8} = 17^{-\text{ord}_{P_{17, 8}} g}.$$  

$$P_{17, -8} = 17R + (x + 8)R: \quad |g|_{17, -8} = 17^{-\text{ord}_{P_{17, -8}} g}.$$  

$$P_{17, -2} = 17R + (x + 2)R: \quad |g|_{17, -2} = 17^{-\text{ord}_{P_{17, -2}} g}.$$  

$$P_{41, 3} = 41R + (x - 3)R: \quad |g|_{41, 3} = 41^{-\text{ord}_{P_{41, 3}} g}.$$  

etc.
**S-units of** $K = \mathbb{Q}[x]/(x^n + 1)$

Assume $\infty \subseteq S \subseteq \{\text{places of } K\}$.
Useful special case: $S$ has all primes $\leq$ something.
[Warning: Often people rename $S - \infty$ as $S$.]

$g \in K^*$ is an **$S$-unit**

$\iff gR = \prod_{P \in S} P^{e_P}$ for some $e_P$
$\iff |g|_\rho = 1$ for all $\rho \in \{\text{places of } K\} - S$
$\iff$ the vector $\rho \mapsto \log |g|_\rho$ is 0 outside $S$.

**$S$-unit lattice:** set of such vectors $\rho \mapsto \log |g|_\rho$.

E.g. Temporarily allowing $n = 1$, $K = \mathbb{Q}$:

$\{\{\infty, 2, 3\}$-units in $\mathbb{Q}\} = \pm 2^\mathbb{Z}3^\mathbb{Z}$. ("3-smooth").

Lattice: $(\log 2, -\log 2, 0)\mathbb{Z} + (\log 3, 0, -\log 3)\mathbb{Z}$. 
0. Choose a finite set $S$ of places.
1. Input a nonzero ideal $I$ of $R$.
2. Find an $S$-generator of $I$:
   some $g$ with $gR = I \prod_{P \in S} P^{e_P}$.
   This has a poly-time quantum algorithm, 
   and surprisingly fast non-quantum algorithms.
3. Find an $S$-unit $u$ “close to $g/I$”.
   This is an $S$-unit-lattice close-vector problem.
4. Output $g/u$.

Critical for Step 3 speed: constructing short vectors 
in the $S$-unit lattice. We’ll see several constructions!
Special case: unit attacks

0. Define $S = \infty$.
   \[ \{\infty\text{-units of } K\} = \{\text{units of } R\} = R^*. \]

1. Input a nonzero ideal $I$ of $R$.
2. Find a generator of $I$: some $g$ with $gR = I$.
3. Find a unit $u$ “close to $g$”.
4. Output $g/u$.

Questions coming up later in this talk:

- How small is $g/u$ compared to $I$?
- What happens if $I$ isn’t principal?
- Is this special case as good as the general case?
“Cyclotomic units” in $R = \mathbb{Z}[x]/(x^n + 1)$

$\pm 1, \pm x, \pm x^2, \ldots, \pm x^{n-1} = \mp 1/x$ are units.

$(1 - x^3)/(1 - x) = 1 + x + x^2 \in R$. Unit since

$(1 - x)/(1 - x^3) = (1 - x^{2n^2+1})/(1 - x^3) \in R$.

For $c \in 1 + 2\mathbb{Z}$: $R$ has automorphism $\sigma_c : x \mapsto x^c$.

$\sigma_c(1 + x + x^2) = 1 + x^c + x^{2c}$ is a unit.

Useful to symmetrize: define $u_c = 1 + x^c + x^{-c}$.

$x^Z \prod_c u_c^Z$ has finite index in $R^*$. Index is called $h^+$. Assume $h^+ = 1$. Proven, assuming GRH, for $n \in \{2, 4, 8, \ldots, 256\}$; heuristics say always true.

[Note to number theorists: This talk is only for powers of 2.]
Unit lattice for $n = 8$

$|u_1|_1 = |1 + \zeta_{16} + \zeta_{16}^{-1}|^2 \approx \exp 2.093.$
$|u_1|_3 = |1 + \zeta_{16}^3 + \zeta_{16}^{-3}|^2 \approx \exp 1.137.$
$|u_1|_5 = |1 + \zeta_{16}^5 + \zeta_{16}^{-5}|^2 \approx \exp -2.899.$
$|u_1|_7 = |1 + \zeta_{16}^7 + \zeta_{16}^{-7}|^2 \approx \exp -0.330.$

Define $\log\infty f = (\log |f|_1, \log |f|_3, \log |f|_5, \log |f|_7)$.

$\log\infty u_1 \approx (2.093, 1.137, -2.899, -0.330).$
$\log\infty u_3 \approx (1.137, -0.330, 2.093, -2.899).$
$\log\infty u_5 \approx (-2.899, 2.093, -0.330, 1.137).$
$\log\infty u_7 \approx (-0.330, -2.899, 1.137, 2.093).$

$\log\infty \mathcal{R}^*$ is lattice of dim $n/2 - 1 = 3$ in hyperplane

$\{(l_1, l_3, l_5, l_7) \in \mathbb{R}^4 : l_1 + l_3 + l_5 + l_7 = 0\}.$

Short lattice basis: $\log\infty u_1, \log\infty u_3, \log\infty u_5.$
Reducing mod units

Start with \[ g = g_0 + g_1 x + \cdots + g_{n-1} x^{n-1}. \]
Compute \[ \text{Log}_\infty g = (\log |g|_1, \log |g|_3, \ldots, \log |g|_{n-1}). \]

Try to reduce \( \text{Log}_\infty g \) modulo unit lattice:
adjust \( \text{Log}_\infty g \) by subtracting closest vector from some precomputed combinations of basis vectors;
repeat several times; keep smallest \( g^2_0 + \cdots + g^2_{n-1} \).

Replacing \( g \) with \( gu \) replaces \( |g|_c \) with \( |g|_c |u|_c \).
Easy to track \( \sum_c |g|_c = (n/2)(g^2_0 + \cdots + g^2_{n-1}) \).

Note that unit hyperplane is orthogonal to norm:
\[ #(R/I) = #(R/g) = \prod_c |g|_c = \exp \sum_c \log |g|_c. \]
Experiments for small $n$

Geometric average of $\eta^{1/n}$ over 100000 experiments:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Model</th>
<th>Attack</th>
<th>Tweak</th>
<th>Shortest</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.01516</td>
<td>1.01518</td>
<td>1.01518</td>
<td>1.01518</td>
</tr>
<tr>
<td>8</td>
<td>1.01968</td>
<td>1.01972</td>
<td>1.01696</td>
<td>1.01696</td>
</tr>
<tr>
<td>16</td>
<td>1.01861</td>
<td>1.01860</td>
<td>1.01628</td>
<td>1.01627</td>
</tr>
</tbody>
</table>

“Shortest”: Take $I$, find a shortest nonzero vector $g$, output $\eta = (g_0^2 + \cdots + g_{n-1}^2)^{1/2}/\#(R/I)^{1/n}$.

[Assuming BKZ-$n$ software produces shortest nonzero vector.]

“Attack”: Same $I$, find a generator, reduce mod unit lattice $\rightarrow g$, output $(g_0^2 + \cdots + g_{n-1}^2)^{1/2}/\#(R/I)^{1/n}$.

“Model”: Take a hyperplane point, reduce mod unit lattice $\rightarrow \log_{\infty} g$, output $(g_0^2 + \cdots + g_{n-1}^2)^{1/2}$. 

Daniel J. Bernstein  
S-unit attacks
Wasn’t this attack supposed to be useless?

Geometric average of 100000 runs of model for 32, 64, 128, 256, 512, 1024: 1.01570, 1.01332, 1.01118, 1.00950, 1.00804, (10000:) 1.00667.

Why did 2019 DPW say $>1.022$ for $n$ below 1000?
Wasn’t this attack supposed to be useless?

Geometric average of 100000 runs of model for 32, 64, 128, 256, 512, 1024: 1.01570, 1.01332, 1.01118, 1.00950, 1.00804, (10000:) 1.00667.

Why did 2019 DPW say $>1.022$ for $n$ below 1000?

Aha: 2019 DPW applies unit attack to principal $IJ$.

Multiplying $J$ into $I$

$\Rightarrow$ multiplying $\#(R/J)$ into $\#(R/I)$

$\Rightarrow$ multiplying $\#(R/J)^{1/n}$ into $\#(R/I)^{1/n}$

$\Rightarrow$ expanding $\eta$ by $\#(R/J)^{1/n}$

$\Rightarrow$ expanding $\eta^{1/n}$ by $\#(R/J)^{1/n^2}$.
Finding a close principal multiple $IJ$

Prime $p \in 1 + 2n\mathbb{Z}$ is contained in $n$ prime ideals $P_c$. “Augmented Stickelberger”: known rank-$n$ lattice $\Lambda \subseteq \mathbb{Z}^n$ with $e \in \Lambda \Rightarrow \prod_c P_c^{e_c}$ principal; e.g., $P_c P_{-c}$.

Poly-time quantum algorithm + minor assumption $\Rightarrow$ some vector $v$ such that $I \prod_c P_c^{v_c}$ is principal.

Search some $e \in \Lambda$, trying to minimize $\sum_c |v_c - e_c|$. Use principal $P_c P_{-c}$ to force $e_c \leq v_c$.

Define $J = \prod_c P_c^{v_c - e_c}$. Then $IJ$ is principal.

Replace $I$ with $IJ$, and apply unit attack.

Contribution to $\eta^{1/n}$: $\#(R/J)^{1/n^2} = (p^{1/n^2})\sum_c |v_c - e_c|$. 

Daniel J. Bernstein  

S-unit attacks  

25
Constructing the 2019 DPW graph

Reverse-engineered procedure to build the graph:

- Experiments for $\sum_c |v_c - e_c|$ (for red curve; blue: limit search; thin: “lower bound”).
- Experiments for reducing mod unit lattice.
- Insert $n^{1/2}$ factor because of notation choices.
- Combine appropriately to obtain $n^{1/2} \eta$.
- Multiply by $n^{-1/2}$ to obtain $\eta$. Graph $\eta^{1/n}$.
Constructing the 2019 DPW graph

Reverse-engineered procedure to build the graph:

• Experiments for $\sum_c |\nu_c - e_c|$ (for red curve; blue: limit search; thin: “lower bound”).
• Experiments for reducing mod unit lattice.
• Insert $n^{1/2}$ factor because of notation choices.
• Combine appropriately to obtain $n^{1/2} \eta$.
• Multiply by $n^{-1/2}$ to obtain $\eta$. Graph $\eta^{1/n}$.
• Typo: Omit the “−” in the previous line.
Constructing the 2019 DPW graph

Reverse-engineered procedure to build the graph:

- Experiments for $\sum_c |v_c - e_c|$ (for red curve; blue: limit search; thin: “lower bound”).
- Experiments for reducing mod unit lattice.
- Insert $n^{1/2}$ factor because of notation choices.
- Combine appropriately to obtain $n^{1/2} \eta$.
- Multiply by $n^{-1/2}$ to obtain $\eta$. Graph $\eta^{1/n}$.
- Typo: Omit the “−” in the previous line.

Big impact of typo: e.g., $n^{1/n} \approx 1.012$ for $n = 512$. Attack is much more effective than graph shows.
Part 3
Better multiplicative attacks
Prime factors of some random integers

\[
\begin{align*}
2 & \cdot 3 & \cdot 59 & \cdot 73 & \cdot 14051 & \cdot 57977 & \cdot 1492315939 \\
136652609 & \cdot 229896280545203 & \\
2^2 & \cdot 43973 & \cdot 2825227 & \cdot 63219409867 & \\
3 & \cdot 7 & \cdot 13 & \cdot 115076653977648103973 & \\
2 & \cdot 5 & \cdot 41 & \cdot 4259 & \cdot 17991127274751277 & \\
11 & \cdot 17 & \cdot 167407 & \cdot 3365381 & \cdot 298195039 & \\
2^3 & \cdot 3^4 & \cdot 29 & \cdot 92401 & \cdot 150959 & \cdot 119850869 & \\
43 & \cdot 730602942695300753131 & \\
2 & \cdot 79 & \cdot 379 & \cdot 577 & \cdot 5009 & \cdot 382979 & \cdot 473971 & \\
3 & \cdot 5 & \cdot 2094395102393195492309 & \\
2^2 & \cdot 7 & \cdot 337 & \cdot 3329369069086258201 & \\
23 & \cdot 4363 & \cdot 14153 & \cdot 22120162700921 & \\
\end{align*}
\]
Traditional method to find $S$-units

Take random small element $u \in R$:
\[ u = x^{31} - x^{41} + x^{59} + x^{26} - x^{53}. \]
1. Does $\#(R/u)$ factor into primes $\leq y$?
2. Is $u$ an $S$-unit for $S = \infty \cup \{P : \#(R/P) \leq y\}$?

Small primes $\Rightarrow$ fast non-quantum factorization.
[Helpful speedups: $\#(R/P) \in 1 + 2n\mathbb{Z}$. Batch factorization.]

Standard heuristics $\Rightarrow y^{2+o(1)}$ choices of $u$
include $y^{1+o(1)}$ $S$-units, spanning all $S$-units, for
- appropriate $n^{1/2+o(1)}$ choice for $\log y$,
- appropriate $n^{1/2+o(1)}$ choice for $\sum_i u_i^2$.

Total time $\exp(n^{1/2+o(1)})$. [Extension NFS: $1/3 + o(1)$?]
Automorphisms and subrings

Apply each $\sigma_c$ to quickly amplify each $u$ found into, typically, $n$ independent $S$-units.

What if $u$ is invariant under (say) two $\sigma_c$? Great! Start with $u$ from proper subrings. Makes $\#(R/u)$ much more likely to factor into small primes.

Examples of useful subrings of $R = \mathbb{Z}[x]/(x^n + 1)$:

- $\mathbb{Z}[x^2]/(x^n + 1) = \{u \in R : \sigma_{n+1}(u) = u\}$.
- $R^+ = \{u \in R : \sigma_{-1}(u) = u\}$.

Also use subrings to speed up $\#(R/u)$ computation for any $u \in R$: $v = u\sigma_{n+1}(u)$, $w = v\sigma_{n/2+1}(v)$, $\ldots$ $n^{1+o(1)}$ times faster than “fast” resultant methods.
More cyclotomic fun: Gauss sums

For each prime number \( p \in 1 + 2n\mathbb{Z} \), and each group morphism \( \chi : F_p^* \rightarrow \zeta_{2n}^\mathbb{Z} \), define

\[
\text{Gauss} \Sigma_p(\chi) = \sum_{a \in F_p^*} \chi(a) \zeta_p^a.
\]

Exercise: \( |\text{Gauss} \Sigma_p(\chi)|^2 = p \) if \( \chi \neq 1 \).
So \( \text{Gauss} \Sigma_p(\chi) \) is an \( S \)-unit for \( S = \infty \cup p \).

e.g. \( n = 16, \zeta_{2n} = \zeta_{32}, p = 97 \in 1 + 2n\mathbb{Z} \):
There is a morphism \( \chi : F_{97}^* \rightarrow \zeta_{32}^\mathbb{Z} \) with \( \chi(5) = \zeta_{32} \).
\[
\text{Gauss} \Sigma_p(\chi) = \zeta_{32}^0 \zeta_{97}^1 + \zeta_{32}^1 \zeta_{97}^5 + \zeta_{32}^2 \zeta_{97}^{25} + \cdots.
\]
\[
\text{Gauss} \Sigma_p(\chi^2) = \zeta_{32}^0 \zeta_{97}^1 + \zeta_{32}^1 \zeta_{97}^5 + \zeta_{32}^2 \zeta_{97}^{25} + \cdots.
\]
Many \( S \)-units for \( S = \infty \cup p \)

Magic fact: \( \text{Gauss}\Sigma_p(\chi)^3 / \text{Gauss}\Sigma_p(\chi^3) \in \mathbb{Z}[\zeta_{2n}] \).

Pull back via \( \iota_1 \) to an element of \( R = \mathbb{Z}[x] / (x^n + 1) \).

Factor element into prime ideals for, e.g., \( n = 16 \):
\[
P_{11} P_{13} P_{15} P_{-15} P_{-13} P_{11} P_{-9} P_{7} P_{5} P_{3} P_{1} \quad \text{where}
\]
\[
P_{\pm 1}, P_{\pm 3}, \ldots, P_{\pm 15} \text{ are the prime ideals containing } p.
\]

Similarly \( \text{Gauss}\Sigma_p(\chi)^5 / \text{Gauss}\Sigma_p(\chi^5) \) etc. \( \Rightarrow \) More principal products of powers of \( P_{\pm 1}, P_{\pm 3}, \ldots, P_{\pm 15} \).

\( \Lambda \) is generated by exponent vectors for (1) these \( S \)-units and (2) \( P_c P_{-c} \) (principal since \( h^+ = 1 \)).

[Note to number theorists: labeling here is \( P_c = \sigma_c^{-1}(P_1) \).]
Explaining the magic: Jacobi sums

Define Jacobi$\Sigma_p(\chi_1, \chi_2) = \sum_{a \in \mathbb{F}_p^* - \{1\}} \chi_1(a)\chi_2(1-a)$.

Exercise: If $\chi_1 \chi_2 \neq 1$ then $\text{Jacobi}\Sigma_p(\chi_1, \chi_2) = \text{Gauss}\Sigma_p(\chi_1) \text{Gauss}\Sigma_p(\chi_2)/\text{Gauss}\Sigma_p(\chi_1 \chi_2)$.

So $|\text{Jacobi}\Sigma_p(\chi_1, \chi_2)|^2 = p$ if $1 \not\in \{\chi_1, \chi_2, \chi_1 \chi_2\}$.

e.g. $n = 16, \zeta_{2n} = \zeta_{32}, p = 97, \chi(5) = \zeta_{32}$:

$\text{Jacobi}\Sigma_p(\chi, \chi) = \zeta_{32}^{1+20} + \zeta_{32}^{2+28} + \zeta_{32}^{3+66} + \cdots$,

$\text{Jacobi}\Sigma_p(\chi^2, \chi) = \zeta_{32}^{2+20} + \zeta_{32}^{4+28} + \zeta_{32}^{6+66} + \cdots$.

since $1 - 5^1 = 5^{20}, 1 - 5^2 = 5^{28}$, etc. in $\mathbb{F}_{97}$.
Λ', improving Λ by a factor 2

JacobiΣ_p(χ^i, χ) for i = 1, i = 2, etc.:
GaussΣ_p(χ)^2 / GaussΣ_p(χ^2),
GaussΣ_p(χ^2)GaussΣ_p(χ) / GaussΣ_p(χ^3),
GaussΣ_p(χ^3)GaussΣ_p(χ) / GaussΣ_p(χ^4),
GaussΣ_p(χ^4)GaussΣ_p(χ) / GaussΣ_p(χ^5), etc.

Multiply:
GaussΣ_p(χ)^2 / GaussΣ_p(χ^2) (wasn’t used in Λ),
GaussΣ_p(χ)^3 / GaussΣ_p(χ^3) (was used in Λ),
GaussΣ_p(χ)^4 / GaussΣ_p(χ^4) (wasn’t used in Λ),
GaussΣ_p(χ)^5 / GaussΣ_p(χ^5) (was used in Λ), etc.

Define Λ’ using all Jacobi sums: all base-field combinations of Gauss sums. 
\#(\mathbb{Z}^n/\Lambda) = 2\#(\mathbb{Z}^n/\Lambda').
Λ”, improving Λ by a factor $2^{n/2}$

Fact: More products $\prod_c P_{c}^{e_{c}}$ are principal if $n \geq 4$. Typical case: $P_{c}$ generates the “class group”; then $\Lambda'$ has index $2^{n/2-1}$ inside lattice of “class relations”.

Class group $= \{ \text{ideals} \neq 0 \} / \{ \text{principal ideals} \neq 0 \}$.

Start from all known $S$-units: group generated by cyclotomic units, Jacobi sums, generators of $P_{c}P_{-c}$. Successively extend set by adjoining square roots.

How to find square products of powers of current generators? Map the group in many ways to $F_{2}$: use known exponents of $P_{c}$; use random quadratic characters (squareness mod random prime ideals $Q$). Then fast linear algebra over $F_{2}$ finds squares.

Daniel J. Bernstein  
S-unit attacks
Example: $n = 8$

Take $p = 17$, $\chi(3) = \zeta_{16}$, $u_c = 1 + x^c + x^{-c}$.

Find generator $g_7 = x^6 - x^5 + x^3 - x^2 - 1$ of $P_7 P_{-7}$.

Compute $\Sigma_i = \text{Jacobi}\Sigma_p(\chi^i, \chi)$ pulled back to $R$.

<table>
<thead>
<tr>
<th>S-unit</th>
<th>ideal factorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma_1 = 2x^7 + 2x^6 - x^4 + 2x^2 - 2x$</td>
<td>$P_{-7} P_{-5} P_{-3} P_{-1}$</td>
</tr>
<tr>
<td>$\Sigma_2 = x^7 - 2x^6 - 3x^5 + x^4 - x^3 - x$</td>
<td>$P_7 P_{-5} P_{-3} P_{-1}$</td>
</tr>
<tr>
<td>$\Sigma_2/\Sigma_1$</td>
<td>$P_7/P_{-7}$</td>
</tr>
<tr>
<td>$g_7$</td>
<td>$P_7 P_{-7}$</td>
</tr>
<tr>
<td>$g_7 \Sigma_2/\Sigma_1$</td>
<td>$P_7^2$</td>
</tr>
<tr>
<td>$(u_5 g_7 \Sigma_2/\Sigma_1)^{1/2} = x^7 - x^4 + x^3$</td>
<td>$P_7$</td>
</tr>
</tbody>
</table>

Scaling up to $n = 256$: All sqrts in 10 minutes.
End of the story for $n = 4$, $n = 8$, $n = 16$

For $n = 16$: $\#(\mathbb{Z}^{16}/\Lambda) = 256$. “Lower bound” $2 \Rightarrow$ expand $\#(R/I)^{1/n^2}$ by $p^2/n^2 = 97^2/n^2 \approx 1.03639$, on top of $\approx 1.01861$ for unit-lattice model.
End of the story for $n = 4, n = 8, n = 16$

For $n = 16$: $\#(\mathbb{Z}^{16}/\Lambda) = 256$. "Lower bound" $2 \Rightarrow$ expand $\#(R/I)^{1/n^2}$ by $p^{2/n^2} = 97^{2/n^2} \approx 1.03639$, on top of $\approx 1.01861$ for unit-lattice model.

Instead construct more $S$-units: $\#(\mathbb{Z}^{16}/\Lambda^{'}) = 1$. The input ideal was principal in the first place! Find generator of $I$. Reduce mod units.

"Tweak": Multiply by $x + 1$, reduce, repeat for $I, (x + 1)I, (x + 1)^2I, (x + 1)^3I, (x + 1)^4I, \ldots$. Often $(x + 1)^e g$ is closer to unit lattice than $g$. Take smallest generator found across all $(x + 1)^e I$. When to stop? Compare $2^e \#(R/I)$ to best $g$.

[Faster: reduce in log space mod units and $x + 1$.]
Recap: Constructing small $S$-units

\[ u_1 = 1 + x + x^{-1} \]

\[ x + 1 \]

\[ \sigma_c \]

\[ S\text{-units} \]

\[ P_1 P_{-1} \text{ gen} \]

\[ \text{square roots} \]

\[ \text{random} \]

\[ \text{in } R \]

\[ \text{in } R^+ \]

\[ \text{Jacobi}\Sigma \]

\[ \text{Gauss}\Sigma \text{ ratios} \]
Impact for larger values of $n$

For $n = 32$: \( \#(\mathbb{Z}^{32}/\Lambda) = 1114112 \).

“Lower bound” 5 ⇒ expand by \( \approx 1.02603 \), on top of \( \approx 1.01570 \) for unit-lattice model.
Impact for larger values of $n$

For $n = 32$: $\#(\mathbb{Z}^{32}/\Lambda) = 1114112$.

“Lower bound” $5 \Rightarrow$ expand by $\approx 1.02603$, on top of $\approx 1.01570$ for unit-lattice model.

Instead construct more $S$-units: $\#(\mathbb{Z}^{32}/\Lambda'') = 17$.

“Class number” $= \#$(class group) $= 17$.


Chance $16/17$: $I$ non-principal. $IP$ principal for some prime ideal $P$ with $\#(R/P) = 193$.

Expansion factor $193^{1/n^2} \approx 1.00515$.

[Note to number theorists: upcoming labels use $P_{p,c} = \sigma_c(P_{p,1})$, with $P_{p,1} = pR + (x + a)R$ for smallest $a$ in $\{0, 1, \ldots, p - 1\}$.]
Broader $n = 32$ search example, part 1

32 prime ideals $P_{193,c}$ have $\#(R/P_{193,c}) = 193$.
32 prime ideals $P_{257,c}$ have $\#(R/P_{257,c}) = 257$.
32 prime ideals $P_{449,c}$ have $\#(R/P_{449,c}) = 449$.
Note $449^{1/n^2} \approx 1.00598$ vs. $193^{1/n^2} \approx 1.00515$.

Precompute $S$-units, including

 generators $\gamma_{193}, \gamma_{257}, \gamma_{449}, \gamma_{577}, \gamma_{641}, \gamma_{769}, \ldots$ of $P_{193,31}^{-1}, P_{257,-19}^{-1}, P_{193,1}^{-1}, P_{449,-19}^{-1}, P_{193,1}^{-1}, P_{577,15}^{-1}, P_{641,19}^{-1}, P_{193,1}^{-1}, P_{769,5}^{-1}, P_{193,1}^{-1}, \ldots$
Broader $n = 32$ search example, part 1

32 prime ideals $P_{193,c}$ have $\#(R/P_{193,c}) = 193$.
32 prime ideals $P_{257,c}$ have $\#(R/P_{257,c}) = 257$.
32 prime ideals $P_{449,c}$ have $\#(R/P_{449,c}) = 449$.
Note $449^{1/n^2} \approx 1.00598$ vs. $193^{1/n^2} \approx 1.00515$.

Precompute $S$-units, including generators $\gamma_{193}, \gamma_{257}, \gamma_{449}, \gamma_{577}, \gamma_{641}, \gamma_{769}, \ldots$ of $P_{193,31}P_{193,1}^{-1}, P_{257,-19}P_{193,1}^{-1}, P_{449,-19}P_{193,1}^{-1}, P_{577,15}P_{193,1}^{-1}, P_{641,19}P_{193,1}^{-1}, P_{769,5}P_{193,1}^{-1}, \ldots$

Random example of a target: $I = 3141592653589793238462643383280129R + (x + 13443234652173688219737012017423)R$.
Initial $S$-generator computation: $gR = IP_{193,13}$. 
Broader $n = 32$ search example, part 2

Multiply by precomputed $S$-units for more $S$-gens of $I$. (Don’t repeat the quantum computations!)

$gR = IP_{193,13}$. Attack: 1.02549; tweak: 1.01901.

$g\sigma_{13}(\gamma_{193})R = IP_{193,19}$. 1.01709; 1.01709.

$g\sigma_{13}(\gamma_{257})R = IP_{257,9}$. 1.02179; 1.02103.

$g\sigma_{13}(\gamma_{193})\sigma_{19}(\gamma_{257})R = IP_{257,23}$. 1.02517; 1.01588.

$g\sigma_{13}(\gamma_{449})R = IP_{449,9}$. 1.02100; 1.02100.

$g\sigma_{13}(\gamma_{193})\sigma_{19}(\gamma_{449})R = IP_{449,23}$. 1.02584; 1.01830.

$g\sigma_{13}(\gamma_{577})R = IP_{577,3}$. 1.02634; 1.02456.

$g\sigma_{13}(\gamma_{193})\sigma_{19}(\gamma_{577})R = IP_{577,29}$. 1.02682; 1.02224.

$g\sigma_{13}(\gamma_{641})R = IP_{641,-9}$. 1.01810; 1.01810.

$g\sigma_{13}(\gamma_{193})\sigma_{19}(\gamma_{641})R = IP_{641,-23}$. 1.00990; 1.00990.
End of the story for $n = 32$

Geometric average of $\eta^{1/n}$ over 10000 experiments:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Attack10</th>
<th>Attack12</th>
<th>Attack14</th>
<th>Shortest</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1.01660</td>
<td>1.01622</td>
<td>1.01599</td>
<td>1.01576</td>
</tr>
</tbody>
</table>

“Attack10”: Tweaked unit attack starting from 12 gens of ideals $IP_{p,c}$ with $p < 2^{10}$.

“Attack12”: Tweaked unit attack starting from same $I$ pool, 32 gens of ideals $IP_{p,c}$ with $p < 2^{12}$.

“Attack14”: Tweaked unit attack starting from same $I$ pool, 124 gens of ideals $IP_{p,c}$ with $p < 2^{14}$.

(If $I$ is principal, take gen of $I$. Could also try $IJ$.)
Generalizing to any $n$

Find $S$-unit lattice: generators of $\prod_{P \in S} P^{e_P}$. Typically see small $P_{\ell,1} \in S$ generating class group; for each $Q \in S$, find generator of some $Q \prod_c P^{e_c}_{\ell,c}$.

Find $S$-generator of $I$: $gR = I \prod_{P \in S} P^{v_P}$.

No more quantum steps required after this.

Try $J = R$, $J = Q$, $J = QQ'$, etc. For each $J$, immediately see generator of some $IJ \prod_c P^{e_c}_{\ell,c}$.

Fast reduction mod $\Lambda'' \Rightarrow$ gen of small multiple of $I$.

(For $n = 32$, jumped to $J$ with $IJ$ principal.) Fast reduction mod unit lattice and $x + 1 \Rightarrow$ short.

Much shorter vectors than pure unit attack.
Using more primes for $n = 64$

$\#(\mathbb{Z}^{64}/\Lambda'') = 17 \cdot 21121 = 359057$.

Again precompute $S$-units.

Given $I$, compute $S$-generator: $gR = I \prod_c P_{257,i}^{v_c}$. 

Basic attack: Reduce exponent vector mod $\Lambda''$, finding generator of small $I \prod_c P_{257,i}^{v_c-e_c}$.

“Small”: 1000 experiments in $\sum_c |v_c - e_c|$ model $\Rightarrow$
25.2% 5, 64.8% 4, 9.6% 3, 0.3% 2, 0.1% 1.

$257^4/n^2 \approx 1.00543; 257^{1/n^2} \approx 1.00136$.

Further options: $I \prod_c P_{641,i}^{v_c}$. Many more options: $IP_{641,b} \prod_c P_{257,i}^{v_c}; IP_{769,a} P_{641,b} \prod_c P_{257,i}^{v_c};$ etc.

Paying 2 primes gains many tries at closeness.
A meet-in-the-middle search for $n = 64$

Efficiently index each ideal class by $e \in \mathbb{Z}/359057$:

$I$ has class $e \iff IP_{257,1}^{-e}$ principal.

$\sigma_{-1}, \sigma_{3}$ act as mults by $-1, 29301$ on $\mathbb{Z}/359057$.

Precompute classes of $P_{257,1}, P_{641,1}, P_{769,1}, P_{1153,1}$ (via small $S$-units): 1, 25489, 99282, 201437.

Start with $S$-generator of $I \Rightarrow$ class of $I$.

Tabulate $64^2$ classes of $IP_{1153,a}P_{769,b}$.

Tabulate $64^2$ classes of $P_{641,c}^{-1}P_{257,d}^{-1}$.

Rough estimate: $64^4/359057 \approx 47$ collisions.

Collision $\Rightarrow IP_{1153,a}P_{769,b}P_{641,c}P_{257,d}$ principal.

Reconstruct $IP_{1153,a}P_{769,b}P_{641,c}P_{257,d}$ generator.

Reduce each generator mod units, and apply tweak.
A numerical example for \( n = 64 \)

Took ideal \( I \subset R \) containing the random prime
31415926535897932384626433832795028841971710593.

Examples of short \( g \in I \) found by meet-in-the-middle
search of principal \( IJ_1J_2 \) with odd \( \#(R/J_j) < 2^{22} \):

<table>
<thead>
<tr>
<th>Ideal generated by ( g )</th>
<th>( \eta^{1/n} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1 + x)^8 IP_{641,...}P_{769,...}P_{78977,...})</td>
<td>1.01399</td>
</tr>
<tr>
<td>((1 + x)^5 IP_{398977,...})</td>
<td>1.01389</td>
</tr>
<tr>
<td>(IP_{641,...}P_{1340033,...})</td>
<td>1.01385</td>
</tr>
<tr>
<td>((1 + x)^4 IP_{257,...}P_{1153,...}P_{11777,...}P_{39041,...})</td>
<td>1.01350</td>
</tr>
<tr>
<td>((1 + x)^3 IP_{35969,...}P_{2350081,...})</td>
<td>1.01288</td>
</tr>
</tbody>
</table>

For comparison, shortest nonzero vector in \( I \):
\((1 + x)IP_{6525293171851009,...}\) | 1.01243 |
Conjectured scalability: \(\exp(n^{1/2+o(1)})\)

Simple algorithm variant, skipping many speedups:

Take traditional \(\log y \in n^{1/2+o(1)}\).
Take \(S = \infty \cup \{P : \#(R/P) \leq y\}\).
Precompute \(\{S\text{-unit } u \in R: \sum_i u_i^2 \leq n^{1/2+o(1)}\}\).

Compute \(S\)-generator \(g\) of \(I\).
Replace \(g\) with \(gu/v\) having log vector closest to \(I\); repeat until stable \(\Rightarrow\) small \(S\)-generator of \(I\).
Multiply by \(P_cP_{-c}\) gens \(\Rightarrow\) short element of \(I\).
Repeat \(y^{O(1)}\) times, avoiding cycles; take shortest.

Heuristics \(\Rightarrow\) \(\eta \leq n^{1/2+o(1)}\), time \(\exp(n^{1/2+o(1)})\).
“Vector within \(\epsilon\) of shortest in subexponential time.”