Computational algebraic number theory tackles lattice-based cryptography

Daniel J. Bernstein
University of Illinois at Chicago &
Technische Universiteit Eindhoven

Moving to the left
Moving to the right
Big generator
Moving through the night
—Yes, "Big Generator", 1987

2013.07 talk slide online:
"I think NTRU should switch to random prime-degree extensions with big Galois groups."

2014.02 blog post:
"Here's a concrete suggestion, which I'll call NTRU Prime, for eliminating the structures that I find worrisome in existing ideal-lattice-based encryption systems."

NTRU Prime uses primes p, q with field $(\mathbf{Z}/q)[x]/(x^p - x - 1)$.

Clear advantage of the usual cyclotomics: minor speedup.

Clear advantage of the usual cyclotomics: minor speedup.

Extra advantage often claimed: some “security reductions”.

Clear advantage of the usual cyclotomics: minor speedup.

Extra advantage often claimed: some "security reductions".
But is this really an advantage?
Lange and I conjecture that security is *negatively* correlated with strength of reductions.

Clear advantage of the usual cyclotomics: minor speedup.

Extra advantage often claimed: some "security reductions". But is this really an advantage? Lange and I conjecture that security is *negatively* correlated with strength of reductions.

Disadvantage of cyclotomics: many more symmetries feed a scary attack strategy. Already serious damage to some lattice-based systems, concerns about other systems.

Typical lattice advertisement:

"Because finding short vectors in high-dimensional lattices has been a notoriously hard algorithmic question for hundreds of years . . . we have solid and unique evidence that lattice-based cryptoschemes are secure."

Typical lattice advertisement:
"Because finding short vectors in high-dimensional lattices has been a notoriously hard algorithmic question for hundreds of years . . . we have solid and unique evidence that lattice-based cryptoschemes are secure."

No. Dangerous exaggeration!
There are many obvious gaps between lattice-based systems and the classic lattice problems: e.g., the systems use ideals.
Important to study these gaps.

2009 Smart–Vercauteren "Fully homomorphic encryption with relatively small key and ciphertext sizes": "Recovering the private key given the public key is therefore an instance of the small principal ideal problem: . . . Given a principal ideal . . . compute a 'small' generator of the ideal. This is one of the core problems in computational number theory and has formed the basis of previous cryptographic proposals, see for example [3]."

Smart–Vercauteren, continued:

"There are currently two approaches to the problem. . . . In conclusion determining the private key given only the public key is an instance of a classical and well studied problem in algorithmic number theory. In particular there are no efficient solutions for this problem, and the only sub-exponential method does not find a solution which is equivalent to our private key."

In fact, the classical studies
focus on small dimensions:
e.g., make table of class numbers
for many quadratic fields,
make table of class numbers
for many cubic fields.

Highlights multiplicative issues.
Low-dim lattice issues are easy.

Far fewer papers
consider scalability
of the algorithmic ideas
to much larger dimensions.

The short-generator problem

Take degree-n number field K.
i.e. field $K \subseteq \mathbf{C}$ with $\text{len}_{\mathbf{Q}} K = n$.

(Weaker specification: field K with $\mathbf{Q} \subseteq K$ and $\text{len}_{\mathbf{Q}} K = n$.)

The short-generator problem

Take degree-n number field K.
i.e. field $K \subseteq \mathbf{C}$ with $\text{len}_{\mathbf{Q}} K = n$.

(Weaker specification: field K with $\mathbf{Q} \subseteq K$ and $\text{len}_{\mathbf{Q}} K = n$.)

e.g. $n = 2$; $K = \mathbf{Q}(i) = \mathbf{Q} \oplus \mathbf{Q}i \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^2 + 1)$.

The short-generator problem

Take degree-n number field K.
i.e. field $K \subseteq \mathbf{C}$ with $\text{len}_{\mathbf{Q}} K = n$.

(Weaker specification: field K with $\mathbf{Q} \subseteq K$ and $\text{len}_{\mathbf{Q}} K = n$.)

e.g. $n = 2$; $K = \mathbf{Q}(i) = \mathbf{Q} \oplus \mathbf{Q}i \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^2 + 1)$.
e.g. $n = 256$; $\zeta = \exp(\pi i/n)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n + 1)$.

The short-generator problem

Take degree-n number field K.
i.e. field $K \subseteq \mathbf{C}$ with $\operatorname{len}_{\mathbf{Q}} K = n$.

(Weaker specification: field K with $\mathbf{Q} \subseteq K$ and $\operatorname{len}_{\mathbf{Q}} K = n$.)

e.g. $n = 2$; $K = \mathbf{Q}(i) = \mathbf{Q} \oplus \mathbf{Q}i \hookrightarrow\!\!\!\rightarrow \mathbf{Q}[x]/(x^2+1)$.
e.g. $n = 256$; $\zeta = \exp(\pi i/n)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\rightarrow \mathbf{Q}[x]/(x^n+1)$.
e.g. $n = 660$; $\zeta = \exp(2\pi i/661)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\rightarrow \mathbf{Q}[x]/(x^n+\cdots+1)$.

The short-generator problem

Take degree-n number field K.
i.e. field $K \subseteq \mathbf{C}$ with $\text{len}_{\mathbf{Q}} K = n$.

(Weaker specification: field K with $\mathbf{Q} \subseteq K$ and $\text{len}_{\mathbf{Q}} K = n$.)

e.g. $n = 2$; $K = \mathbf{Q}(i) = \mathbf{Q} \oplus \mathbf{Q}i \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^2 + 1)$.
e.g. $n = 256$; $\zeta = \exp(\pi i/n)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n + 1)$.
e.g. $n = 660$; $\zeta = \exp(2\pi i/661)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n + \cdots + 1)$.
e.g. $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.

Define $\mathcal{O} = \overline{\mathbf{Z}} \cap K$; subring of K.
$\mathcal{O} \hookrightarrow\!\!\!\!\!\twoheadrightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

Nonzero ideals of $\mathcal{O}$
factor uniquely as products of
powers of prime ideals of $\mathcal{O}$.

Define $\mathcal{O} = \overline{\mathbf{Z}} \cap K$; subring of K.
$\mathcal{O} \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

Nonzero ideals of $\mathcal{O}$
factor uniquely as products of
powers of prime ideals of $\mathcal{O}$.

e.g. $K = \mathbf{Q}(i) \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Q}[x]/(x^2+1)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[i] \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^2+1)$.

Define $\mathcal{O} = \overline{\mathbf{Z}} \cap K$; subring of K.
$\mathcal{O} \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

Nonzero ideals of $\mathcal{O}$
factor uniquely as products of
powers of prime ideals of $\mathcal{O}$.

e.g. $K = \mathbf{Q}(i) \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Q}[x]/(x^2+1)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[i] \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^2+1)$.
e.g. $\zeta = \exp(\pi i/256)$, $K = \mathbf{Q}(\zeta)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^{256}+1)$.

Define $\mathcal{O} = \overline{\mathbf{Z}} \cap K$; subring of K.
$\mathcal{O} \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

Nonzero ideals of $\mathcal{O}$
factor uniquely as products of
powers of prime ideals of $\mathcal{O}$.

e.g. $K = \mathbf{Q}(i) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^2+1)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[i] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^2+1)$.
e.g. $\zeta = \exp(\pi i/256)$, $K = \mathbf{Q}(\zeta)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^{256}+1)$.
e.g. $\zeta = \exp(2\pi i/661)$, $K = \mathbf{Q}(\zeta)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \cdots$.

Define $\mathcal{O} = \overline{\mathbf{Z}} \cap K$; subring of K.
$\mathcal{O} \hookrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

Nonzero ideals of $\mathcal{O}$
factor uniquely as products of
powers of prime ideals of $\mathcal{O}$.

e.g. $K = \mathbf{Q}(i) \hookrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathbf{Q}[x]/(x^2+1)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[i] \hookrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^2+1)$.
e.g. $\zeta = \exp(\pi i/256)$, $K = \mathbf{Q}(\zeta)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^{256}+1)$.
e.g. $\zeta = \exp(2\pi i/661)$, $K = \mathbf{Q}(\zeta)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\!\!\twoheadrightarrow \cdots$.
e.g. $K = \mathbf{Q}(\sqrt{5}) \Rightarrow \mathcal{O} =$
$\mathbf{Z}[(1+\sqrt{5})/2] \hookrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^2-x-1)$.

The short-generator problem:
Find "short" nonzero $g \in \mathcal{O}$ given the principal ideal $g\mathcal{O}$.

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow \mathbf{Z}[x]/(x^4 + 1)$.
The $\mathbf{Z}$-submodule of $\mathcal{O}$ gen by
$201 - 233\zeta - 430\zeta^2 - 712\zeta^3$,
$935 - 1063\zeta - 1986\zeta^2 - 3299\zeta^3$,
$979 - 1119\zeta - 2092\zeta^2 - 3470\zeta^3$,
$718 - 829\zeta - 1537\zeta^2 - 2546\zeta^3$
is an ideal I of $\mathcal{O}$.
Can you find a short $g \in \mathcal{O}$ such that $I = g\mathcal{O}$?

The lattice perspective

Use LLL to quickly find short elements of lattice $\mathbf{Z}A + \mathbf{Z}B + \mathbf{Z}C + \mathbf{Z}D$ where

$A = (201, -233, -430, -712)$,

$B = (935, -1063, -1986, -3299)$,

$C = (979, -1119, -2092, -3470)$,

$D = (718, -829, -1537, -2546)$.

The lattice perspective

Use LLL to quickly find short elements of lattice $\mathbf{Z}A + \mathbf{Z}B + \mathbf{Z}C + \mathbf{Z}D$ where
$A = (201, -233, -430, -712)$,
$B = (935, -1063, -1986, -3299)$,
$C = (979, -1119, -2092, -3470)$,
$D = (718, -829, -1537, -2546)$.

Find $(3, 1, 4, 1)$ as
$-37A + 3B - 7C + 16D$.
This was my original g.

The lattice perspective

Use LLL to quickly find short elements of lattice $\mathbf{Z}A + \mathbf{Z}B + \mathbf{Z}C + \mathbf{Z}D$ where
$A = (201, -233, -430, -712)$,
$B = (935, -1063, -1986, -3299)$,
$C = (979, -1119, -2092, -3470)$,
$D = (718, -829, -1537, -2546)$.

Find $(3, 1, 4, 1)$ as
$-37A + 3B - 7C + 16D$.
This was my original g.

Also find, e.g., $(-4, -1, 3, 1)$.
Multiplying by root of unity (here ζ^2) preserves shortness.

For much larger n:

LLL almost never finds g.
Big gap between size of g
and size of "short" vectors
that LLL typically finds in I.

For much larger n:

LLL almost never finds g.
Big gap between size of g
and size of “short” vectors
that LLL typically finds in I.

Increased BKZ block size:
reduced gap but slower.

For much larger n:

LLL almost never finds g.
Big gap between size of g
and size of "short" vectors
that LLL typically finds in I.

Increased BKZ block size:
reduced gap but slower.

Fancier lattice algorithms:
Under reasonable assumptions,
2015 Laarhoven–de Weger
finds g in time $\approx 1.23^n$.
Big progress compared to, e.g.,
2008 Nguyen–Vidick ($\approx 1.33^n$)
but still exponential time.

Exploiting factorization

Use LLL, BKZ, etc. to
generate rather short $\alpha \in g\mathcal{O}$.
What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

Pure lattice approach: Discard α.
Work much harder, find shorter α.

Exploiting factorization

Use LLL, BKZ, etc. to generate rather short $\alpha \in g\mathcal{O}$. What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

Pure lattice approach: Discard α. Work much harder, find shorter α.

Alternative: Gain information from factorization of ideals.

Exploiting factorization

Use LLL, BKZ, etc. to
generate rather short $\alpha \in g\mathcal{O}$.
What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

Pure lattice approach: Discard α.
Work much harder, find shorter α.

Alternative: Gain information
from factorization of ideals.

e.g. If $\alpha_1\mathcal{O} = g\mathcal{O} \cdot P^2 \cdot Q^2$

Exploiting factorization

Use LLL, BKZ, etc. to
generate rather short $\alpha \in g\mathcal{O}$.
What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

Pure lattice approach: Discard α.
Work much harder, find shorter α.

Alternative: Gain information
from factorization of ideals.

e.g. If $\alpha_1\mathcal{O} = g\mathcal{O} \cdot P^2 \cdot Q^2$
and $\alpha_2\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^3$

Exploiting factorization

Use LLL, BKZ, etc. to generate rather short $\alpha \in g\mathcal{O}$. What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

Pure lattice approach: Discard α. Work much harder, find shorter α.

Alternative: Gain information from factorization of ideals.

e.g. If $\alpha_1\mathcal{O} = g\mathcal{O} \cdot P^2 \cdot Q^2$
and $\alpha_2\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^3$
and $\alpha_3\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^2$

Exploiting factorization

Use LLL, BKZ, etc. to
generate rather short $\alpha \in g\mathcal{O}$.
What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

Pure lattice approach: Discard α.
Work much harder, find shorter α.

Alternative: Gain information
from factorization of ideals.

e.g. If $\alpha_1\mathcal{O} = g\mathcal{O} \cdot P^2 \cdot Q^2$
and $\alpha_2\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^3$
and $\alpha_3\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^2$ then
$P = \alpha_1\alpha_3^{-1}\mathcal{O}$ and $Q = \alpha_2\alpha_3^{-1}\mathcal{O}$
and $g\mathcal{O} = \alpha_1^{-1}\alpha_2^{-2}\alpha_3^4\mathcal{O}$.

General strategy: For many α's, factor $\alpha\mathcal{O}$ into products of powers of some primes and $g\mathcal{O}$.

Solve system of equations
to find generator for $g\mathcal{O}$
as product of powers of the α's.

General strategy: For many α's, factor $\alpha\mathcal{O}$ into products of powers of some primes and $g\mathcal{O}$.

Solve system of equations
to find generator for $g\mathcal{O}$
as product of powers of the α's.

"Can the system be solved?"

— Becomes increasingly reasonable to expect as the number of equations approaches and passes the number of primes.

General strategy: For many α's, factor $\alpha\mathcal{O}$ into products of powers of some primes and $g\mathcal{O}$.

Solve system of equations
to find generator for $g\mathcal{O}$
as product of powers of the α's.

"Can the system be solved?"

— Becomes increasingly reasonable to expect as the number of equations approaches and passes the number of primes.

"But {primes} is infinite!"

— Restrict to a “factor base”:
e.g., all primes of norm $\leq y$.

— Restrict to a “factor base”:
e.g., all primes of norm $\leq y$.

“But what if $\alpha\mathcal{O}$ doesn’t
factor into those primes?”

— Restrict to a “factor base”:
e.g., all primes of norm $\leq y$.

“But what if $\alpha\mathcal{O}$ doesn't factor into those primes?”

— Then throw it away.
But often it *does* factor.

— Restrict to a “factor base”:
e.g., all primes of norm $\leq y$.

“But what if $\alpha\mathcal{O}$ doesn't factor into those primes?”

— Then throw it away.
But often it *does* factor.

Familiar issue from “index calculus” DL methods, CFRAC, LS, QS, NFS, etc.
Model the norm of $(\alpha/g)\mathcal{O}$ as “random” integer in $[1, x]$;
y-smoothness chance $\approx 1/y$ if $\log y \approx \sqrt{(1/2) \log x \log \log x}$.

Variation: Ignore $g\mathcal{O}$.
Generate rather short $\alpha \in \mathcal{O}$,
factor $\alpha\mathcal{O}$ into small primes.
After enough α's,
solve system of equations;
obtain generator for each prime.

Variation: Ignore $g\mathcal{O}$.

Generate rather short $\alpha \in \mathcal{O}$,

factor $\alpha\mathcal{O}$ into small primes.

After enough α's,

solve system of equations;

obtain generator for each prime.

After this precomputation,

factor *one* $\alpha\mathcal{O} \subseteq g\mathcal{O}$;

obtain generator for $g\mathcal{O}$.

Variation: Ignore $g\mathcal{O}$.

Generate rather short $\alpha \in \mathcal{O}$,
factor $\alpha\mathcal{O}$ into small primes.
After enough α's,
solve system of equations;
obtain generator for each prime.
After this precomputation,
factor *one* $\alpha\mathcal{O} \subseteq g\mathcal{O}$;
obtain generator for $g\mathcal{O}$.

"Do all primes have generators?"

Variation: Ignore $g\mathcal{O}$.

Generate rather short $\alpha \in \mathcal{O}$,
factor $\alpha\mathcal{O}$ into small primes.
After enough α's,
solve system of equations;
obtain generator for each prime.

After this precomputation,
factor *one* $\alpha\mathcal{O} \subseteq g\mathcal{O}$;
obtain generator for $g\mathcal{O}$.

"Do all primes have generators?"

— Standard heuristics:
For many (most?) number fields,
yes; but for big cyclotomics, no!
Modulo a few small primes, yes.

$\{$principal nonzero ideals$\}$ is kernel of a semigroup map $\{$nonzero ideals$\} \twoheadrightarrow C$ where C is a finite abelian group, the "class group of K".

Fundamental object of study in algebraic number theory.

$\{$principal nonzero ideals$\}$ is kernel of a semigroup map $\{$nonzero ideals$\} \twoheadrightarrow C$ where C is a finite abelian group, the "class group of K".

Fundamental object of study in algebraic number theory.

Factoring many small $\alpha\mathcal{O}$ is a standard textbook method of computing class group and generators of ideals.

$\{$principal nonzero ideals$\}$ is kernel of a semigroup map $\{$nonzero ideals$\} \twoheadrightarrow C$ where C is a finite abelian group, the "class group of K".

Fundamental object of study in algebraic number theory.

Factoring many small $\alpha\mathcal{O}$ is a standard textbook method of computing class group and generators of ideals.

Also compute unit group $\mathcal{O}^*$ via ratios of generators.

A note on time analysis

Smart–Vercauteren statement regarding similar algorithm by Buchmann: "This method has complexity $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)})$."

A note on time analysis

Smart–Vercauteren statement regarding similar algorithm by Buchmann: “This method has complexity $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)})$.”

— [citation needed]

A note on time analysis

Smart–Vercauteren statement regarding similar algorithm by Buchmann: "This method has complexity $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)})$."

— [citation needed]

Did they mean Θ? And $+$?
$\exp(\Theta(N \log N))$ factor
for short-vector enumeration?
Silly: BKZ works just fine.

A note on time analysis

Smart–Vercauteren statement regarding similar algorithm by Buchmann: "This method has complexity $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)})$."

— [citation needed]

Did they mean Θ? And $+$?
$\exp(\Theta(N \log N))$ factor
for short-vector enumeration?
Silly: BKZ works just fine.
The whole algorithm will be subexponential unless norms are much worse than exponential.

Big generator

Smart–Vercauteren: "However this method is likely to produce a generator of large height, i.e., with large coefficients. Indeed so large, that writing the obtained generator down as a polynomial in θ may take exponential time."

Indeed, generator found for $g\mathcal{O}$ is product of powers of various α's. Must be gu for some $u \in \mathcal{O}^*$, but extremely unlikely to be g.

Big generator

Smart–Vercauteren: "However this method is likely to produce a generator of large height, i.e., with large coefficients. Indeed so large, that writing the obtained generator down as a polynomial in θ may take exponential time."

Indeed, generator found for $g\mathcal{O}$ is product of powers of various α's. Must be gu for some $u \in \mathcal{O}^*$, but extremely unlikely to be g.

How do we find g from gu?

There are exactly n distinct ring maps $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

There are exactly n distinct ring maps $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

Define $\mathrm{Log} : K^* \to \mathbf{R}^n$ by $\mathrm{Log} = (\log|\varphi_1|, \ldots, \log|\varphi_n|)$.

There are exactly n distinct ring maps $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

Define $\operatorname{Log} : K^* \to \mathbf{R}^n$ by $\operatorname{Log} = (\log|\varphi_1|, \ldots, \log|\varphi_n|)$.

$\operatorname{Log} \mathcal{O}^*$ is a lattice of rank $r_1 + r_2 - 1$ where

$$r_1 = \#\{i : \varphi_i(K) \subseteq \mathbf{R}\},$$
$$2r_2 = \#\{i : \varphi_i(K) \not\subseteq \mathbf{R}\}.$$

There are exactly n distinct ring maps $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

Define $\mathrm{Log} : K^* \to \mathbf{R}^n$ by $\mathrm{Log} = (\log|\varphi_1|, \ldots, \log|\varphi_n|)$.

$\mathrm{Log}\, \mathcal{O}^*$ is a lattice of rank $r_1 + r_2 - 1$ where

$$r_1 = \#\{i : \varphi_i(K) \subseteq \mathbf{R}\},$$
$$2r_2 = \#\{i : \varphi_i(K) \not\subseteq \mathbf{R}\}.$$

e.g. $\zeta = \exp(\pi i/256)$, $K = \mathbf{Q}(\zeta)$: images of ζ under ring maps are $\zeta, \zeta^3, \zeta^5, \ldots, \zeta^{511}$.
$r_1 = 0$; $r_2 = 128$; rank 127.

Compute Log $g u$
as sum of multiples of Log α
for the original α's.

Compute $\operatorname{Log} g u$
as sum of multiples of $\operatorname{Log} \alpha$
for the original α's.

Find elements of $\operatorname{Log} \mathcal{O}^*$
close to $\operatorname{Log} g u$.

Compute $\operatorname{Log} g u$
as sum of multiples of $\operatorname{Log} \alpha$
for the original α's.

Find elements of $\operatorname{Log} \mathcal{O}^*$
close to $\operatorname{Log} g u$.

This is a close-vector problem
("bounded-distance decoding").
"Embedding" heuristic:
CVP as fast as SVP.

Compute $\operatorname{Log} gu$
as sum of multiples of $\operatorname{Log} \alpha$
for the original α's.

Find elements of $\operatorname{Log} \mathcal{O}^*$
close to $\operatorname{Log} gu$.

This is a close-vector problem
("bounded-distance decoding").
"Embedding" heuristic:
CVP as fast as SVP.

This finds $\operatorname{Log} u$.
Easily reconstruct g
up to a root of unity.
#{roots of unity} is small.

A subfield-logarithm attack

Say we know $\operatorname{Log}\operatorname{norm}_{K:F} g$ for a proper subfield $F \subset K$.

A subfield-logarithm attack

Say we know $\operatorname{Log} \operatorname{norm}_{K:F} g$ for a proper subfield $F \subset K$.

We also know $\operatorname{Log} \operatorname{norm}_{K:F} gu$, so we know $\operatorname{Log} \operatorname{norm}_{K:F} u$.

A subfield-logarithm attack

Say we know $\text{Log}\,\text{norm}_{K:F}\,g$ for a proper subfield $F \subset K$.

We also know $\text{Log}\,\text{norm}_{K:F}\,gu$, so we know $\text{Log}\,\text{norm}_{K:F}\,u$.

This linearly constrains $\text{Log}\,u$ to a shifted sublattice of $\text{Log}\,\mathcal{O}^*$. Number of independent constraints: unit rank for F.

A subfield-logarithm attack

Say we know $\operatorname{Log}\operatorname{norm}_{K:F} g$ for a proper subfield $F \subset K$.

We also know $\operatorname{Log}\operatorname{norm}_{K:F} g u$, so we know $\operatorname{Log}\operatorname{norm}_{K:F} u$.

This linearly constrains $\operatorname{Log} u$ to a shifted sublattice of $\operatorname{Log} \mathcal{O}^*$. Number of independent constraints: unit rank for F.

Find elements close to $\operatorname{Log} g u$. Lower-dimension lattice problem, if unit rank of F is positive.

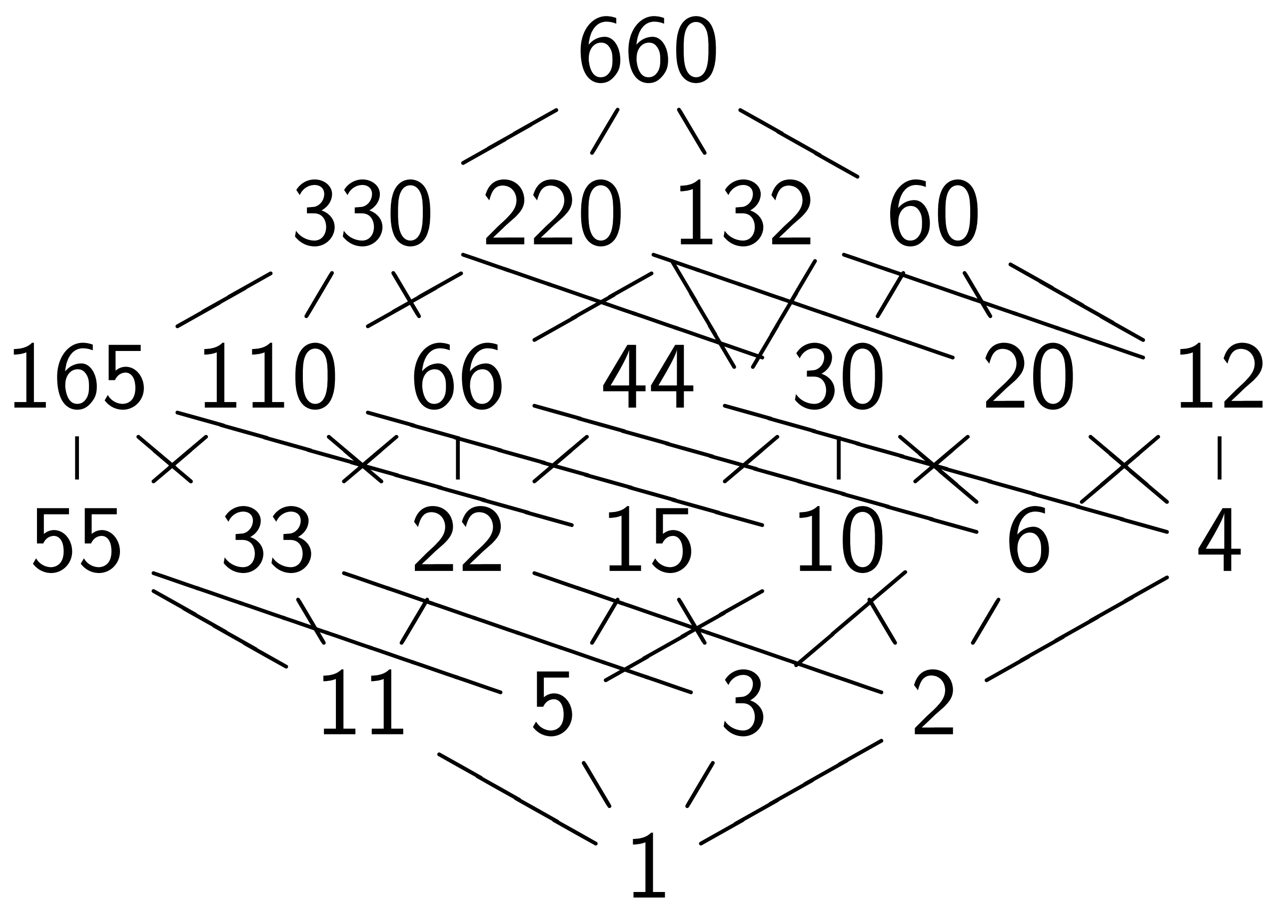
Start by recursively computing $\operatorname{Log}\operatorname{norm}_{K:F} g$ via norm of $g\mathcal{O}$ for *each* $F \subset K$.

Various constraints on $\operatorname{Log} u$, depending on subfield structure.

Start by recursively computing $\text{Log}\,\text{norm}_{K:F}\,g$ via norm of $g\mathcal{O}$ for *each* $F \subset K$.

Various constraints on $\text{Log}\,u$, depending on subfield structure.

e.g. $\zeta = \exp(2\pi i/661)$, $K = \mathbf{Q}(\zeta)$. Degrees of subfields of K:



Most extreme case:

Composite of quadratics, such as $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.

CVP becomes trivial!

Most extreme case:

Composite of quadratics, such as $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.

CVP becomes trivial!

Many intermediate cases.

"Subexponential in *cyclotomic* rings of *highly smooth* index":

It's much more general than that.

Most extreme case:
Composite of quadratics, such as
$K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.
CVP becomes trivial!

Many intermediate cases.
"Subexponential in *cyclotomic* rings of *highly smooth* index":
It's much more general than that.

For cyclotomics this approach is superseded by subsequent Campbell–Groves–Shepherd algorithm, using known (good) basis for cyclotomic units.