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 $(\mathbf{Z}/q)[x]/(x^p - x - 1)$.

ational
c number theory
attice-based cryptography

. Bernstein
ty of Illinois at Chicago &
che Universiteit Eindhoven

Moving to the left
Moving to the right
Big generator
Moving through the night
/es, "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 ad
cycloton

theory
ed cryptography

n
is at Chicago &
siteit Eindhoven

Moving to the left
ving to the right
Big generator
hrough the night
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 o
cyclotomics: mino

graphy

ago &
hoven

the left
e right
nerator
e night
, 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 usua
cyclotomics: minor speedup

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.

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.

Extra advantage often claimed: some "security reductions".

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.

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.

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.

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.

talk slide online:
NTRU should switch to
prime-degree extensions
Galois groups."

blog post:
a concrete suggestion,
ll call NTRU Prime,
nating the structures
nd worrisome in
ideal-lattice-based
on systems."

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

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
"Becaus
in high-
has beer
algorithr
of years
unique e
cryptosc

online:
ould switch to
ree extensions
oups."

:
e suggestion,
RU Prime,
e structures
me in
ce-based
s."

primes p, q
$]/(x^p - x - 1)$.

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 adv
"Because finding s
in high-dimensiona
has been a notorio
algorithmic questio
of years ... we ha
unique evidence th
cryptoschemes are

h to
ions

on,

s

q
$- 1)$.

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 vecto
in high-dimensional lattices
has been a notoriously hard
algorithmic question for hun
of years . . . we have solid an
unique evidence that lattice-
cryptoschemes are secure."

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.”

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."

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.

vantage of the usual
nics: minor speedup.

lvantage often claimed:
ecurity reductions".
his really an advantage?
nd I conjecture that
is *negatively* correlated
ength of reductions.

ntage of cyclotomics:
ore symmetries
cary attack strategy.
serious damage
lattice-based systems,
s 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."

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 Sm
homomo
relatively
sizes": "
key give
therefore
principal
a princip
'small' g
This is c
in comp
and has
previous
see for e

f the usual
r speedup.

ften claimed:
luctions".
n advantage?
cture that
ely correlated
eductions.

yclotomics:
etries
k strategy.
mage
sed systems,
her 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."

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–Verca
homomorphic encr
relatively small key
sizes": "Recoverin
key given the publ
therefore an instan
principal ideal prob
a principal ideal . .
'small' generator o
This is one of the
in computational n
and has formed th
previous cryptogra
see for example [3

l

.

ed:

ge?

ted

s:

ns,

ns.

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 "F
homomorphic encryption wit
relatively small key and ciph
sizes": "Recovering the priv
key given the public key is
therefore an instance of the
principal ideal problem: . . .
a principal ideal . . . comput
'small' generator of the idea
This is one of the core probl
in computational number th
and has formed the basis of
previous cryptographic propo
see for example [3]."

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]."

lattice advertisement:
e finding short vectors
dimensional lattices
n a notoriously hard
mic question for hundreds
. . . we have solid and
evidence that lattice-based
hemes are secure."

ngerous exaggeration!
re many obvious gaps
lattice-based systems
classic lattice problems:
systems use ideals.
nt 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–V
"There a
approach
In conclu
private k
key is ar
and well
algorithm
particula
solutions
the only
does not
equivaler

vertisement:
short vectors
al lattices
ously hard
on for hundreds
ve solid and
hat lattice-based
secure."

aggeration!
bvious gaps
sed systems
tice problems:
use ideals.
y 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–Vercautere
"There are current
approaches to the
In conclusion dete
private key given
key is an instance
and well studied p
algorithmic numbe
particular there ar
solutions for this p
the only sub-expon
does not find a so
equivalent to our

t:
ors

dreds
nd
-based

n!
os
ns
ems:

ps.

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, continu
"There are currently two
approaches to the problem.
In conclusion determining th
private key given only the pu
key is an instance of a classi
and well studied problem in
algorithmic number theory.
particular there are no efficie
solutions for this problem, a
the only sub-exponential me
does not find a solution whi
equivalent to our private key

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.”

nart–Vercauteren "Fully
orphic encryption with
y small key and ciphertext
'Recovering the private
n the public key is
e an instance of the small
 ideal problem: . . . Given
oal ideal . . . compute a
generator of the ideal.
one of the core problems
utational number theory
 formed the basis of
 cryptographic proposals,
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, t
focus on
e.g., ma
for many
make ta
for many

Highligh
Low-dim

Far fewe
consider
of the al
to much

uteren "Fully
ryption with
y and ciphertext
ng the private
ic key is
nce of the small
blem: . . . Given
. compute a
of the ideal.
core problems
number theory
e basis of
phic proposals,
]."

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 classic
focus on small dim
e.g., make table of
for many quadratic
make table of clas
for many cubic fie

Highlights multipli
Low-dim lattice iss

Far fewer papers
consider scalability
of the algorithmic
to much larger din

ully
th
ertext
ate

small
Given
e a
l.
ems
eory

osals,

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 nur
for many quadratic fields,
make table of class numbers
for many cubic fields.

Highlights multiplicative issu
Low-dim lattice issues are ea

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

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.

ercauteren, continued:
are currently two
hes to the problem. . . .
usion determining the
key given only the public
instance of a classical
studied problem in
mic number theory. In
ar there are no efficient
s for this problem, and
sub-exponential method
t find a solution which is
nt 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 sho

Take deg
i.e. field

(Weaker
with **Q**

n, continued:
tly two
problem. . . .
rmining the
only the public
of a classical
roblem in
er theory. In
e no efficient
problem, and
nential method
lution which is
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-generat

Take degree-n nu
i.e. field $K \subseteq \mathbf{C}$ w

(Weaker specificat
with $\mathbf{Q} \subseteq K$ and l

ed:

. . .
ne
ublic
ical

In
ent
nd
thod
ch is
y."

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.

<u>The short-generator problem</u>

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

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

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 $\operatorname{len}_{\mathbf{Q}} K = n$.

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

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 $\operatorname{len}_{\mathbf{Q}} K = n$.

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

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

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 $\text{len}_{\mathbf{Q}} K = n$.)

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

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 $\text{len}_{\mathbf{Q}} K = n$.)

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

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 $\text{len}_{\mathbf{Q}} K = n$.)

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. $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.

the classical studies
small dimensions:
ke table of class numbers
y quadratic fields,
ble of class numbers
y cubic fields.

ts multiplicative issues.
lattice issues are easy.

er papers
scalability
lgorithmic ideas
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 $\text{len}_{\mathbf{Q}} K = n$.)

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 $\mathcal{C}$
$\mathcal{O} \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}^n$

Nonzero
factor u
powers o

al studies
nensions:
f class numbers
c fields,
s numbers
lds.

icative issues.
sues are easy.

/
 ideas
nensions.

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 $\mathcal{O} = \overline{\mathbf{Z}} \cap K$
$\mathcal{O} \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-mo

Nonzero ideals of
factor uniquely as
powers of prime id

mbers

s

ues.

asy.

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

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

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

e.g. $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.

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

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

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. $n = 660$; $\zeta = \exp(2\pi i/661)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\rightarrow \mathbf{Q}[x]/(x^n + \cdots + 1)$.
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.
$\mathcal{O} \hookrightarrow\!\!\!\rightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

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

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. $n = 660$; $\zeta = \exp(2\pi i/661)$;
$K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n+\cdots+1)$.
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.
$\mathcal{O} \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}^n$ as $\mathbf{Z}$-modules.

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

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

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

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. $n = 660$; $\zeta = \exp(2\pi i/661)$; $K = \mathbf{Q}(\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n+\cdots+1)$.
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.
$\mathcal{O} \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}^n$ 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. $\zeta = \exp(2\pi i/661)$, $K = \mathbf{Q}(\zeta)$
$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \cdots$.

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

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

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

rt-generator problem

gree-n number field K.

$K \subseteq \mathbf{C}$ with $\operatorname{len}_{\mathbf{Q}} K = n$.

specification: field K

$\subseteq K$ and $\operatorname{len}_{\mathbf{Q}} K = n$.)

2; $K = \mathbf{Q}(i) =$

$\hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^2+1)$.

256; $\zeta = \exp(\pi i/n)$;

$\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n+1)$.

660; $\zeta = \exp(2\pi i/661)$;

$\zeta) \hookrightarrow\!\!\!\!\rightarrow \mathbf{Q}[x]/(x^n+\cdots+1)$.

$= \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.

Define $\mathcal{O} = \overline{\mathbf{Z}} \cap K$; subring of K.

$\mathcal{O} \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}^n$ 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. $\zeta = \exp(2\pi i/661)$, $K = \mathbf{Q}(\zeta)$

$\Rightarrow \mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \cdots$.

e.g. $K = \mathbf{Q}(\sqrt{5}) \Rightarrow \mathcal{O} =$
$\mathbf{Z}[(1+\sqrt{5})/2] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^2-x-1)$.

The sho

Find "sh

given th

e.g. $\zeta =$

$\mathcal{O} = \mathbf{Z}[\zeta$

The $\mathbf{Z}$-s

$201 - 23$

$935 - 10$

$979 - 11$

$718 - 82$

is an ide

Can you

such tha

or problem

mber field K.

ith $\mathrm{len}_{\mathbf{Q}} K = n$.

tion: field K

en$_{\mathbf{Q}}$ $K = n$.)

$\mathbf{Q}(i) =$

$x^2 + 1)$.

$\exp(\pi i/n)$;

$]/(x^n + 1)$.

$\exp(2\pi i/661)$;

$]/(x^n + \cdots + 1)$.

$3, \sqrt{5}, \ldots, \sqrt{29})$.

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 $\mathcal{O}$.

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-generat

Find "short" nonz

given the principal

e.g. $\zeta = \exp(\pi i/4$

$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/$

The $\mathbf{Z}$-submodule

$201 - 233\zeta - 430$

$935 - 1063\zeta - 198$

$979 - 1119\zeta - 209$

$718 - 829\zeta - 153$

is an ideal I of $\mathcal{O}$.

Can you find a sho

such that $I = g\mathcal{O}$

n

K.

$= n$.

K

n.)

);

).

661);

$\cdot + 1)$.

$, \sqrt{29})$.

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

e.g. $K = \mathbf{Q}(\sqrt{5}) \Rightarrow \mathcal{O} = \mathbf{Z}[(1+\sqrt{5})/2] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^2-x-1)$.

The short-generator problem

Find "short" nonzero $g \in \mathcal{O}$

given the principal ideal $g\mathcal{O}$

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}($

$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^4+1)$.

The $\mathbf{Z}$-submodule of $\mathcal{O}$ gen

$201 - 233\zeta - 430\zeta^2 - 712\zeta$

$935 - 1063\zeta - 1986\zeta^2 - 32$

$979 - 1119\zeta - 2092\zeta^2 - 34$

$718 - 829\zeta - 1537\zeta^2 - 254$

is an ideal I of $\mathcal{O}$.

Can you find a short $g \in \mathcal{O}$

such that $I = g\mathcal{O}$?

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

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

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^4+1)$.
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 $I = g\mathcal{O}$?

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

ideals of $\mathcal{O}$
niquely as products of
of prime ideals of $\mathcal{O}$.

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

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 $I = g\mathcal{O}$?

<u>The latt</u>

Use LLL
short ele
$\mathbf{Z}A + \mathbf{Z}$
$A = (20$
$B = (93$
$C = (97$
$D = (71$

; subring of K.
dules.

$\mathcal{O}$

products of
leals of $\mathcal{O}$.

$\mathbf{Q}[x]/(x^2+1)$
$[x]/(x^2+1)$.
56), $K=\mathbf{Q}(\zeta)$
$[x]/(x^{256}+1)$.
661), $K=\mathbf{Q}(\zeta)$
$\ddots$
$\Rightarrow \mathcal{O}=$
$\mathbf{Z}[x]/(x^2-x-1)$.

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

e.g. $\zeta = \exp(\pi i/4)$; $K=\mathbf{Q}(\zeta)$;
$\mathcal{O}=\mathbf{Z}[\zeta] \hookrightarrow\!\!\!\rightarrow \mathbf{Z}[x]/(x^4+1)$.
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 perspe

Use LLL to quickl
short elements of
$\mathbf{Z}A+\mathbf{Z}B+\mathbf{Z}C+$
$A=(201,-233,-$
$B=(935,-1063,$
$C=(979,-1119,$
$D=(718,-829,-$

of K.

of

$+ 1)$

$1)$.

$\mathbf{Q}(\zeta)$

$+ 1)$.

$\mathbf{Q}(\zeta)$

$x - 1)$.

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

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^4 + 1)$.
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
$A = (201, -233, -430, -71$
$B = (935, -1063, -1986, -$
$C = (979, -1119, -2092, -$
$D = (718, -829, -1537, -2$

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

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\twoheadrightarrow \mathbf{Z}[x]/(x^4+1)$.
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 $I = g\mathcal{O}$?

<u>The lattice perspective</u>

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

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\to \mathbf{Z}[x]/(x^4+1)$.
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
$A = (201, -233, -430, -712)$,
$B = (935, -1063, -1986, -3299)$,
$C = (979, -1119, -2092, -3470)$,
$D = (718, -829, -1537, -2546)$.

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

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

e.g. $\zeta = \exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\mathcal{O} = \mathbf{Z}[\zeta] \hookrightarrow\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^4+1)$.
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
$A = (201, -233, -430, -712)$,
$B = (935, -1063, -1986, -3299)$,
$C = (979, -1119, -2092, -3470)$,
$D = (718, -829, -1537, -2546)$.

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.

rt-generator problem:
hort" nonzero $g \in \mathcal{O}$
e principal ideal $g\mathcal{O}$.

$\exp(\pi i/4)$; $K = \mathbf{Q}(\zeta)$;
$\zeta] \hookrightarrow\!\!\!\!\!\rightarrow \mathbf{Z}[x]/(x^4+1)$.
ubmodule of $\mathcal{O}$ gen by
$33\zeta - 430\zeta^2 - 712\zeta^3$,
$063\zeta - 1986\zeta^2 - 3299\zeta^3$,
$119\zeta - 2092\zeta^2 - 3470\zeta^3$,
$29\zeta - 1537\zeta^2 - 2546\zeta^3$
eal I of $\mathcal{O}$.
find a short $g \in \mathcal{O}$
nt $I = g\mathcal{O}$?

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
$A = (201, -233, -430, -712)$,
$B = (935, -1063, -1986, -3299)$,
$C = (979, -1119, -2092, -3470)$,
$D = (718, -829, -1537, -2546)$.

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 muc

LLL alm
Big gap
and size
that LLL

or problem:
ero $g \in \mathcal{O}$
ideal $g\mathcal{O}$.

); $K = \mathbf{Q}(\zeta)$;
$/(x^4 + 1)$.
of $\mathcal{O}$ gen by
$\zeta^2 - 712\zeta^3$,
$86\zeta^2 - 3299\zeta^3$,
$92\zeta^2 - 3470\zeta^3$,
$7\zeta^2 - 2546\zeta^3$

ort $g \in \mathcal{O}$
?

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
$A = (201, -233, -430, -712)$,
$B = (935, -1063, -1986, -3299)$,
$C = (979, -1119, -2092, -3470)$,
$D = (718, -829, -1537, -2546)$.

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
Big gap between s
and size of “short”
that LLL typically

n:

)

.

ζ);

by

^{3},

$99\zeta^3$,

$70\zeta^3$,

$6\zeta^3$

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.

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.

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.

Increased BKZ block size: reduced gap but slower.

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.

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.

ice perspective

to quickly find
ements of lattice
$B + \mathbf{Z}C + \mathbf{Z}D$ where
$01, -233, -430, -712)$,
$35, -1063, -1986, -3299)$,
$79, -1119, -2092, -3470)$,
$28, -829, -1537, -2546)$.

$1, 4, 1)$ as
$3B - 7C + 16D$.
s my original g.

d, e.g., $(-4, -1, 3, 1)$.
ing by root of unity
) 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.

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.

Exploitir

Use LLL
generate
What ha

Pure lat
Work m

ctive

y find
lattice
$\mathbf{Z}D$ where
$-430, -712)$,
$-1986, -3299)$,
$-2092, -3470)$,
$-1537, -2546)$.

$+ 16D$.
nal g.

$4, -1, 3, 1)$.
ot of unity
s 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.

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 factoriz

Use LLL, BKZ, etc
generate rather sh
What happens if c

Pure lattice appro
Work much harder

e

2),

3299),

3470),

546).

).

s.

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 $\alpha \in g$
What happens if $\alpha\mathcal{O} \neq g\mathcal{O}$

Pure lattice approach: Disca
Work much harder, find sho

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.

<u>Exploiting factorization</u>

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.

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.

<u>Exploiting factorization</u>

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.

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.

Alternative: Gain information
from factorization of ideals.

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

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.

Alternative: Gain information
from factorization of ideals.

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

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.

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 $\alpha_3\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^2$

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.

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 $g\mathcal{O} = \alpha_1^{-1}\alpha_2^{-2}\alpha_3^4\mathcal{O}$.

h larger n:

ost never finds g.
between size of g
of "short" vectors
L typically finds in I.

d BKZ block size:
gap but slower.

lattice algorithms:
easonable assumptions,
arhoven–de Weger
n time $\approx 1.23^n$.
gress compared to, e.g.,
guyen–Vidick ($\approx 1.33^n$)
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.

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
factor α
of some

Solve sy
to find g
as produ

:

finds g.
ize of g
' vectors
finds in I.

ock size:
lower.

orithms:
assumptions,
e Weger
.23^n.
pared to, e.g.,
ck ($\approx$1.33^n)
al 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.

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:
factor $\alpha\mathcal{O}$ into pro
of some primes an

Solve system of ec
to find generator f
as product of pow

.

ns,

.g.,

$3^n)$

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
factor $\alpha\mathcal{O}$ into products of p
of some primes and $g\mathcal{O}$.

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

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.

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.

"Can the system be solved?"

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

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.

"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!"

ng factorization

, BKZ, etc. to
e rather short $\alpha \in g\mathcal{O}$.
appens if $\alpha\mathcal{O} \neq g\mathcal{O}$?

tice approach: Discard α.
uch harder, find shorter α.

ive: Gain information
ctorization of ideals.

$_1\mathcal{O} = g\mathcal{O} \cdot P^2 \cdot Q^2$
$\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^3$
$\mathcal{O} = g\mathcal{O} \cdot P \cdot Q^2$ then
$\alpha_3^{-1}\mathcal{O}$ and $Q = \alpha_2\alpha_3^{-1}\mathcal{O}$
$= \alpha_1^{-1}\alpha_2^{-2}\alpha_3^4\mathcal{O}$.

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.

"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!"

— Restr
e.g., all

ation

c. to
ort $\alpha \in g\mathcal{O}$.
$\alpha\mathcal{O} \neq g\mathcal{O}$?

ach: Discard α.
r, find shorter α.

information
of ideals.

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

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.

"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 "f
e.g., all primes of

$\mathcal{O}$.

?

ard α.

rter α.

on

n

$_3^{-1}\mathcal{O}$

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.

"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 $\leq y$.

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 $\le y$.

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 $\leq y$.

"But what if $\alpha\mathcal{O}$ doesn't factor into those 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 $\le y$.

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

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

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 $\le y$.

"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 $\log y \approx \sqrt{(1/2)\log x \log\log x}$.

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

stem of equations
generator for $g\mathcal{O}$
ict of powers of the α's.

e system be solved?"

mes increasingly
ble to expect as the
of equations approaches
ses the number of primes.

rimes} is infinite!"

— 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 $\log y \approx \sqrt{(1/2) \log x \log\log x}$.

Variatio
Generate
factor α
After en
solve sys
obtain g

For many α's,
oducts of powers
d $g\mathcal{O}$.

quations
for $g\mathcal{O}$
ers of the α's.

be solved?"

asingly
ect as the
ons approaches
mber of primes.

nfinite!"

— Restrict to a "factor base": e.g., all primes of norm $\le y$.

"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
Generate rather sh
factor $\alpha\mathcal{O}$ into sm
After enough α's,
solve system of eq
obtain generator fo

α's,
powers

α's.

”

ches
rimes.

— 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 pr

— 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.

— Restrict to a "factor base": e.g., all primes of norm $\le y$.

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

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

“Do all primes have generators?”

— 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. 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.

rict to a "factor base":
primes of norm $\le y$.

hat if $\alpha\mathcal{O}$ doesn't
to those primes?"

throw it away.
en it *does* factor.

issue from
alculus" DL methods,
LS, QS, NFS, etc.
he norm of $(\alpha/g)\mathcal{O}$
lom" integer in $[1, x]$;
hness chance $\approx 1/y$
$\approx \sqrt{(1/2)\log x \log\log x}$.

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.
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.

{principa
kernel of
{nonzero
C is a fi
the "clas

Fundame
in algebr

factor base":
norm $\le y$.

doesn't
primes?"

away.
factor.

n
DL methods,
NFS, etc.
f $(\alpha/g)\mathcal{O}$
ger in $[1, x]$;
nce $\approx 1/y$
$\log x \log\log x$.

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.
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
kernel of a semigro
{nonzero ideals} –
C is a finite abelia
the "class group o

Fundamental objec
in algebraic numbe

e":

ls,

];

og 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.
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.

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.

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.

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

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.

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.

n: Ignore $g\mathcal{O}$.
e rather short $\alpha \in \mathcal{O}$,
$\mathcal{O}$ into small primes.
ough α's,
stem of equations;
enerator for each prime.
is precomputation,
ne $\alpha\mathcal{O} \subseteq g\mathcal{O}$;
enerator for $g\mathcal{O}$.

primes have generators?"

dard heuristics:
y (most?) number fields,
for big cyclotomics, no!
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.

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 o

Smart–V
regardin
Buchma
complex
$\sqrt{\log(\Delta}$

$g\mathcal{O}$.
hort $\alpha \in \mathcal{O}$,
nall primes.

uations;
or each prime.
putation,
$g\mathcal{O}$;
or $g\mathcal{O}$.

ve generators?"

stics:
number fields,
clotomics, no!
all 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.

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 an

Smart–Vercautere
regarding similar a
Buchmann: "This
complexity exp(O(
$\sqrt{\log(\Delta) \cdot \log\log($

),

.

ime.

ors?”

ields,
no!
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.

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.

<u>A note on time analysis</u>

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

$\{$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.

<u>A note on time analysis</u>

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

$\{$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 $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)})$."

— [citation needed]

$\{$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 $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)})$."

— [citation needed]

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

$\{$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 $\exp(O(N \log N) \cdot \sqrt{\log(\Delta) \cdot \log\log(\Delta)}))$."

— [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.

al nonzero ideals} is
f a semigroup map
o ideals} $\twoheadrightarrow C$ where
nite abelian group,
ss group of K".

ental object of study
raic number theory.

g many small $\alpha\mathcal{O}$
dard textbook method
uting class group
erators of ideals.

mpute unit group $\mathcal{O}^*$
s of generators.

A note on time analysis

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

— [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 gene

Smart–V
this met
a genera
with larg
large, th
generato
θ may ta

Indeed,
product
Must be
but extr

ideals} is
oup map
$\twoheadrightarrow C$ where
n group,
f K".

ct of study
er theory.

nall $\alpha\mathcal{O}$
book method
s group
ideals.

t group $\mathcal{O}^*$
ators.

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–Vercauterer
this method is like
a generator of larg
with large coefficie
large, that writing
generator down as
θ may take expone

Indeed, generator
product of powers
Must be gu for so
but extremely unli

e

y

od

*

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: "Howev this method is likely to prod a generator of large height, with large coefficients. Indee large, that writing the obtai generator down as a polynor θ may take exponential time

Indeed, generator found for product of powers of various Must be gu for some $u \in \mathcal{O}$ but extremely unlikely to be

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.

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.

How do we find g from gu?

on time analysis

/ercauteren statement
g similar algorithm by
nn: “This method has
ity $\exp(O(N \log N) \cdot$
$) \cdot \log\log(\Delta))$.”

tion needed]

/ mean Θ? And $+$?
$/ \log N))$ factor
t-vector enumeration?
KZ works just fine.
ole algorithm will be
nential unless norms are
orse 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.

How do we find g from gu?

There ar
ring ma

alysis

n statement
lgorithm by
method has
$(N \log N) \cdot$
$\overline{(\Delta)})$."

d]

? And $+$?
actor
umeration?
just fine.
hm will be
less norms are
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.

How do we find g from gu?

There are exactly
ring maps $\varphi_1, \ldots,$

nt
by
has
-

n?

s are
al.

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 $\varphi_1, \ldots, \varphi_n : K \rightarrow$

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 $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

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 $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

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

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 $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

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

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

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

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 $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

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.

erator

Vercauteren: "However
hod is likely to produce
tor of large height, i.e.,
ge coefficients. Indeed so
at writing the obtained
or down as a polynomial in
ake exponential time."

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

we find g from gu?

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.

Comput
as sum
for the

n: "However
ely to produce
ge height, i.e.,
ents. Indeed so
the obtained
a polynomial in
ential time."

found for $g\mathcal{O}$ is
of various α's.
me $u \in \mathcal{O}^*$,
kely to be g.

from gu?

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

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 $\operatorname{Log} gu$
as sum of multiple
for the original α's

ver
uce
i.e.,
ed so
ned
mial in
e."

$g\mathcal{O}$ is
s α's.
*,
g.

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

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

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

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

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

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

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

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 $\operatorname{Log} g u$ 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.

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

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 $\operatorname{Log} g u$ 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.

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

re exactly n distinct
ps $\varphi_1, \ldots, \varphi_n : K \to \mathbf{C}$.

og : $K^* \to \mathbf{R}^n$ by
$\log|\varphi_1|, \ldots, \log|\varphi_n|)$.

is a lattice
$r_1 + r_2 - 1$ where
$\{i : \varphi_i(K) \subseteq \mathbf{R}\}$,
$\{i : \varphi_i(K) \not\subseteq \mathbf{R}\}$.

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

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 subfie

Say we k
for a pro

n distinct

$\varphi_n : K \to \mathbf{C}$.

$\to \mathbf{R}^n$ by

$. . , \log|\varphi_n|)$.

e

1 where

$) \subseteq \mathbf{R}\}$,

$) \not\subseteq \mathbf{R}\}$.

56), $K = \mathbf{Q}(\zeta)$:

ring maps

^{511}.

rank 127.

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.

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

<u>A subfield-logarith</u>

Say we know Log r

for a proper subfie

$\mathbf{C}$.

).

$\mathbf{Q}(\zeta)$:

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 $\operatorname{Log} g u$.

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 $F \subset K$.

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

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

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

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

<u>A subfield-logarithm attack</u>

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

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.

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 $F \subset K$.

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

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.

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 $F \subset K$.

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.

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.

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

A subfield-logarithm attack

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

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.

e Log $g\,u$
of multiples of Log α
original α's.

ments of Log $\mathcal{O}^*$
Log $g\,u$.

n close-vector problem
ed-distance decoding").
ding" heuristic:
fast as SVP.

ds Log u.
econstruct g
root of unity.
of unity} is small.

A subfield-logarithm attack

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

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

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

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

Start by
Log norm
for *each*

Various
dependin

es of Log α
s.

Log $\mathcal{O}^*$

tor problem
ce decoding").
ristic:
VP.

g
ity.
is small.

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 $\operatorname{Log} \operatorname{norm}_{K:F} u$.

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 $\operatorname{Log} gu$. Lower-dimension lattice problem, if unit rank of F is positive.

Start by recursivel
$\operatorname{Log} \operatorname{norm}_{K:F} g$ via
for *each* $F \subset K$.

Various constraint
depending on subf

α

m

g").

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 computi
$\operatorname{Log} \operatorname{norm}_{K:F} g$ via norm of
for *each* $F \subset K$.

Various constraints on Log
depending on subfield struct

A subfield-logarithm attack

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

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

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

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

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

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

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 $\operatorname{Log} \operatorname{norm}_{K:F} u$.

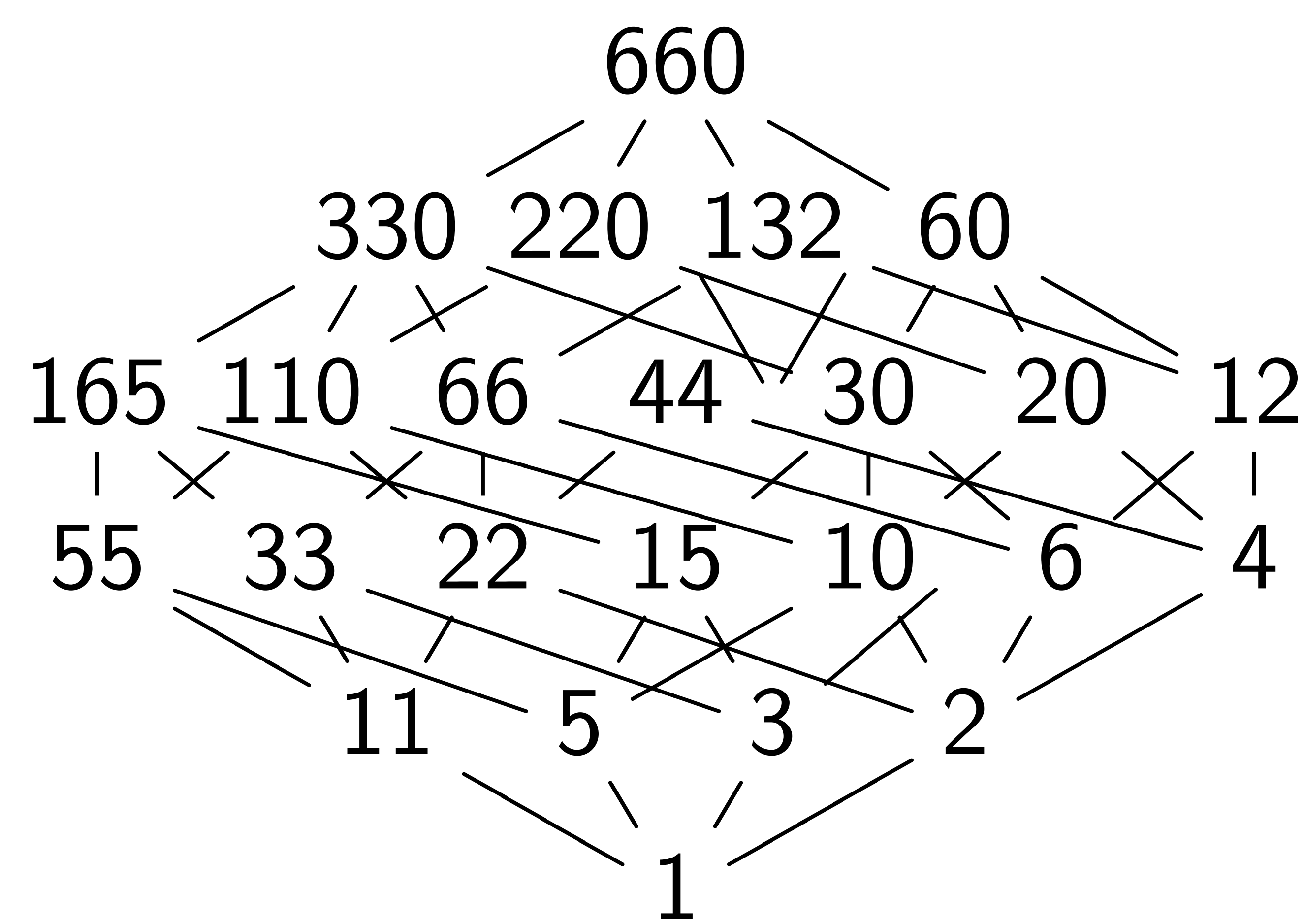
This linearly constrains $\operatorname{Log} u$ to a shifted sublattice of $\operatorname{Log} \mathcal{O}^*$. Number of independent constraints: unit rank for F.

Find elements close to $\operatorname{Log} gu$. 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.

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



ld-logarithm attack

know $\text{Log}\,\text{norm}_{K:F}\, g$
oper subfield $F \subset K$.

know $\text{Log}\,\text{norm}_{K:F}\, g u$,
now $\text{Log}\,\text{norm}_{K:F}\, u$.

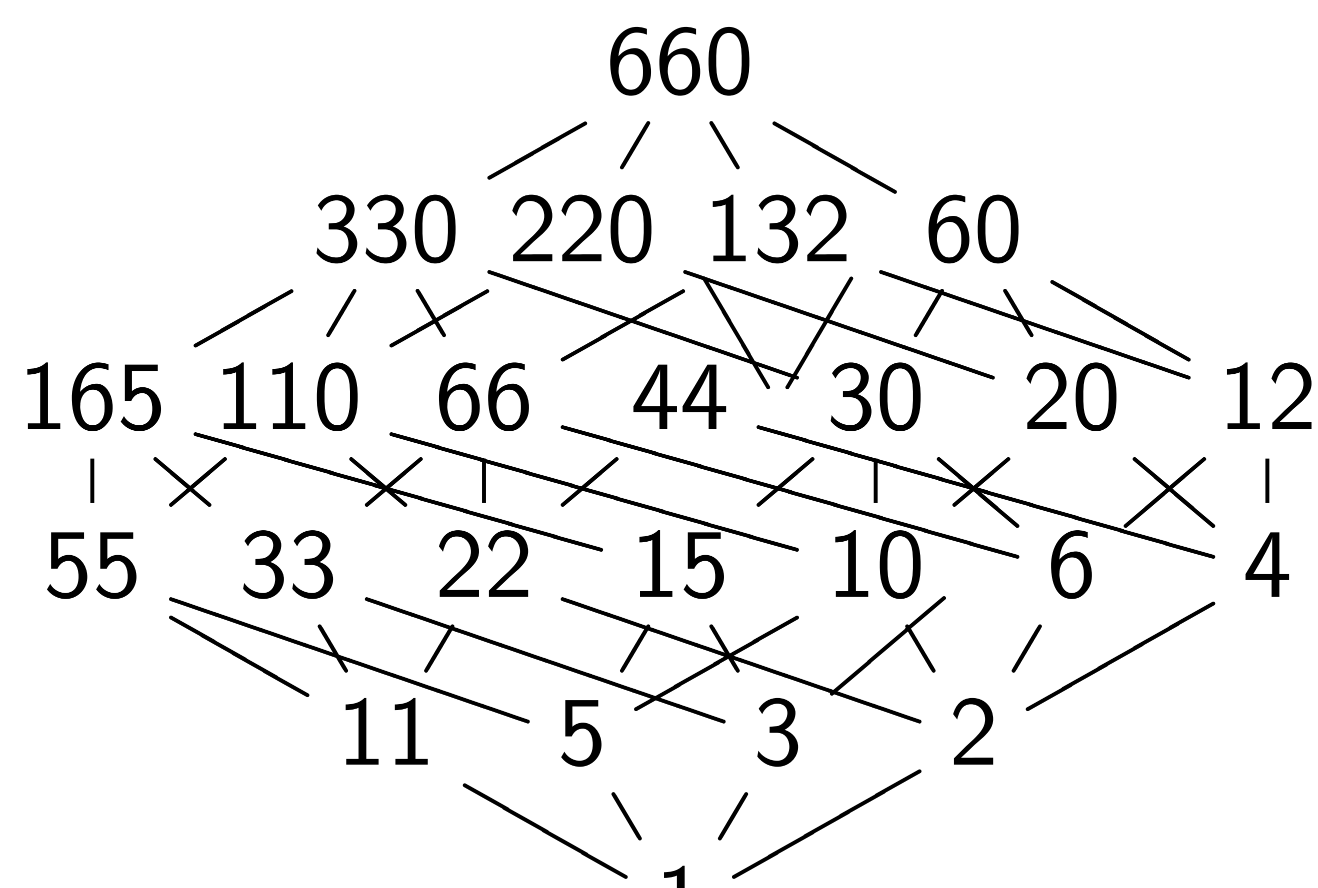
early constrains $\text{Log}\, u$
fted sublattice of $\text{Log}\, \mathcal{O}^*$.
of independent
nts: unit rank for F.

ments close to $\text{Log}\, g u$.
imension lattice problem,
ank of F is positive.

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 ex
Compos
$K = \mathbf{Q}(\surd$
CVP bec

m attack

norm$_{K:F}\, g$
eld $F \subset K$.

norm$_{K:F}\, g u$,
orm$_{K:F}\, u$.

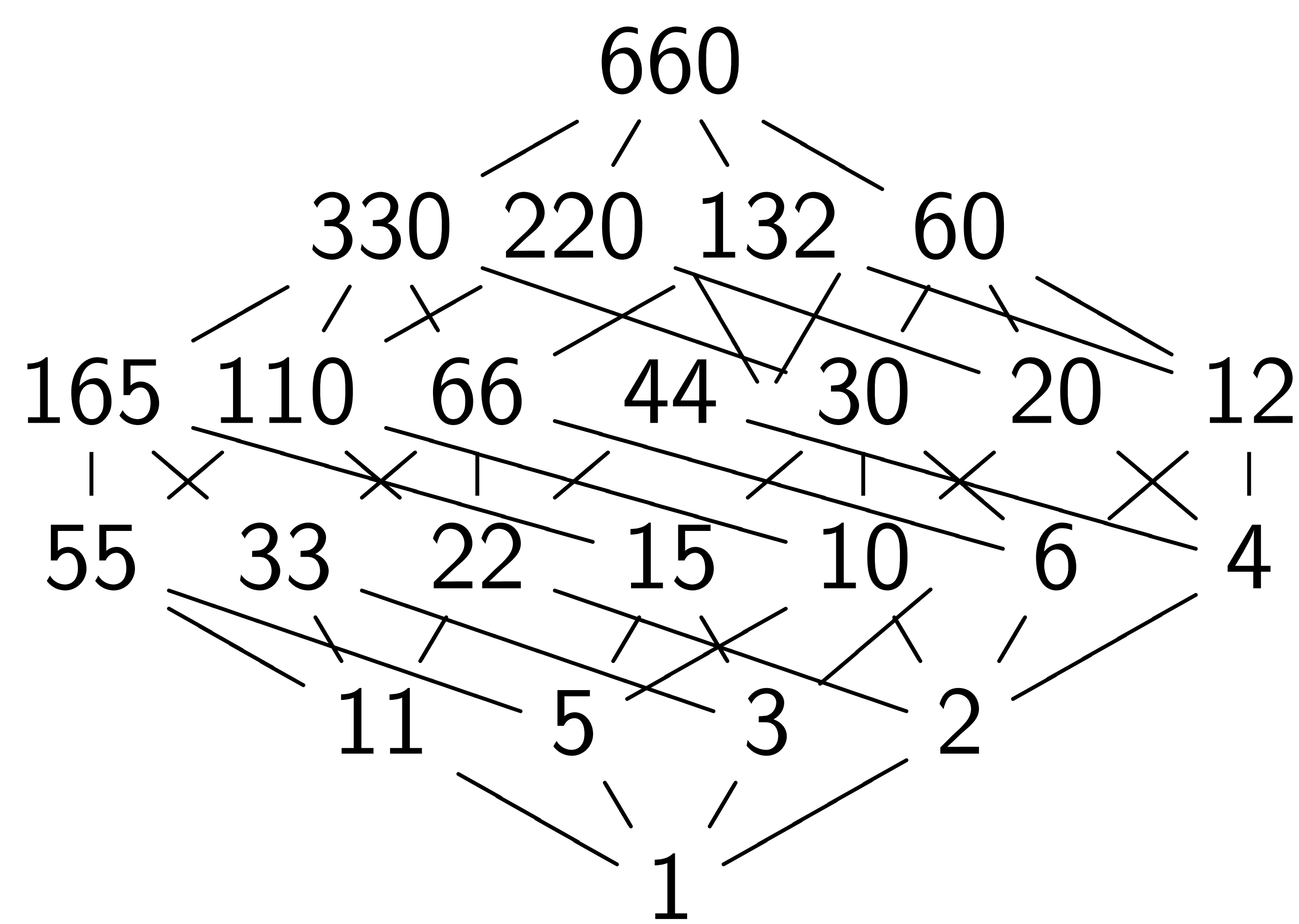
rains Log u
tice of Log $\mathcal{O}^*$.
ndent
rank for F.

se to Log $g u$.
attice problem,
s positive.

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

Various constraints on 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 quad
$K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{\ }$
CVP becomes triv

.

.

$g\,u$,

u

$g\,\mathcal{O}^*$.

.

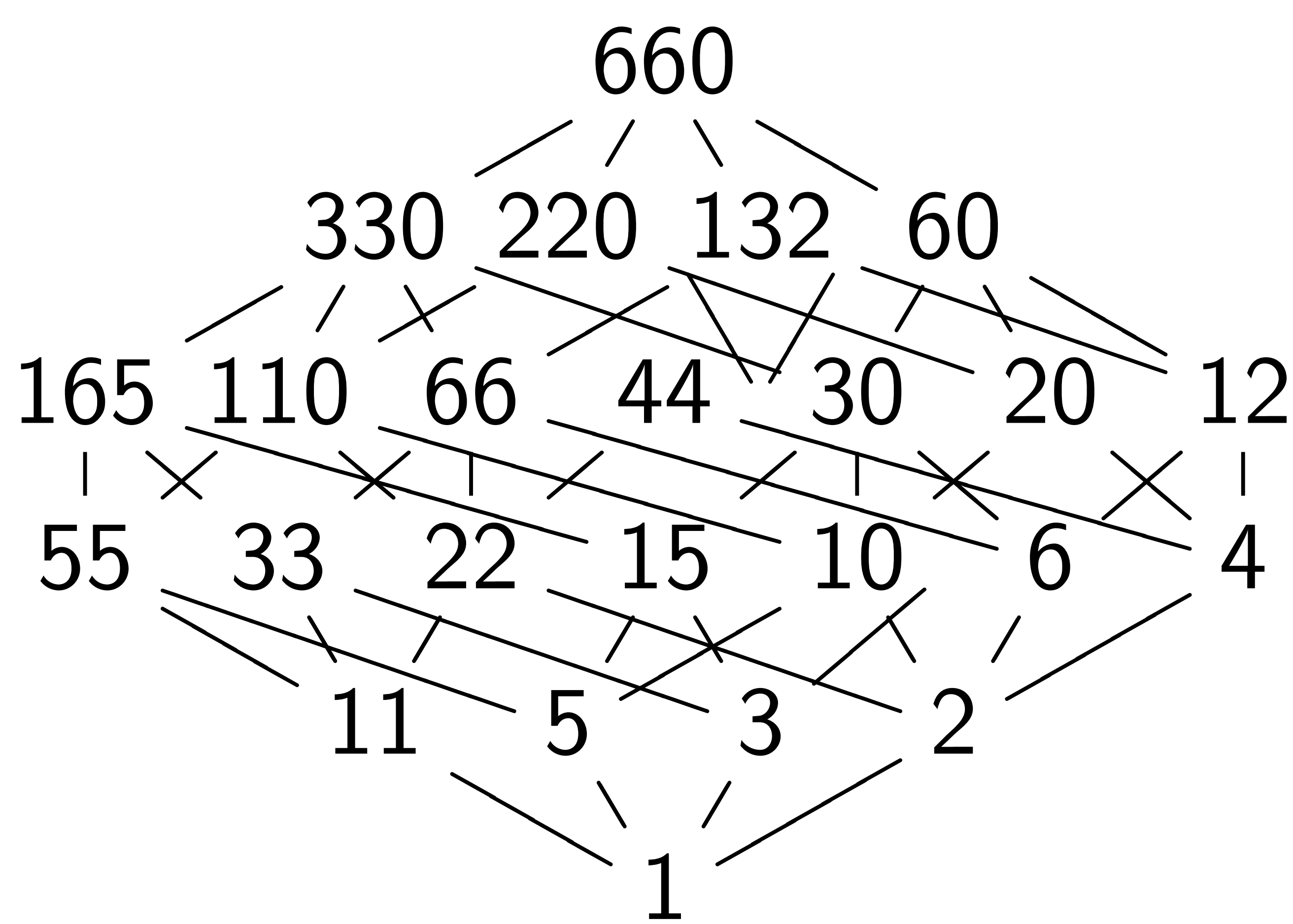
$g\,u$.

blem,

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.

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

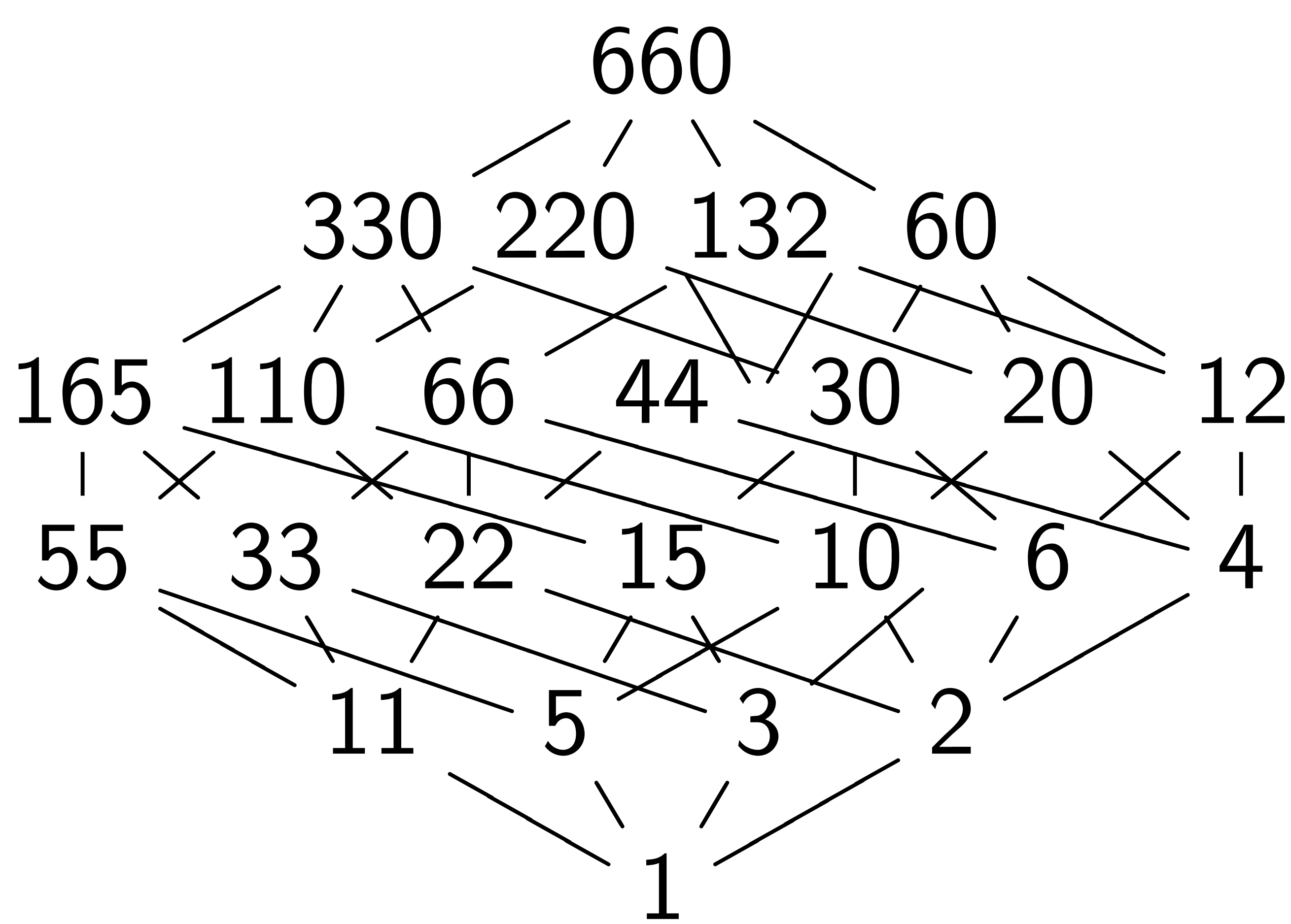


Most extreme case:
Composite of quadratics, su
$K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29}$
CVP becomes trivial!

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.

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

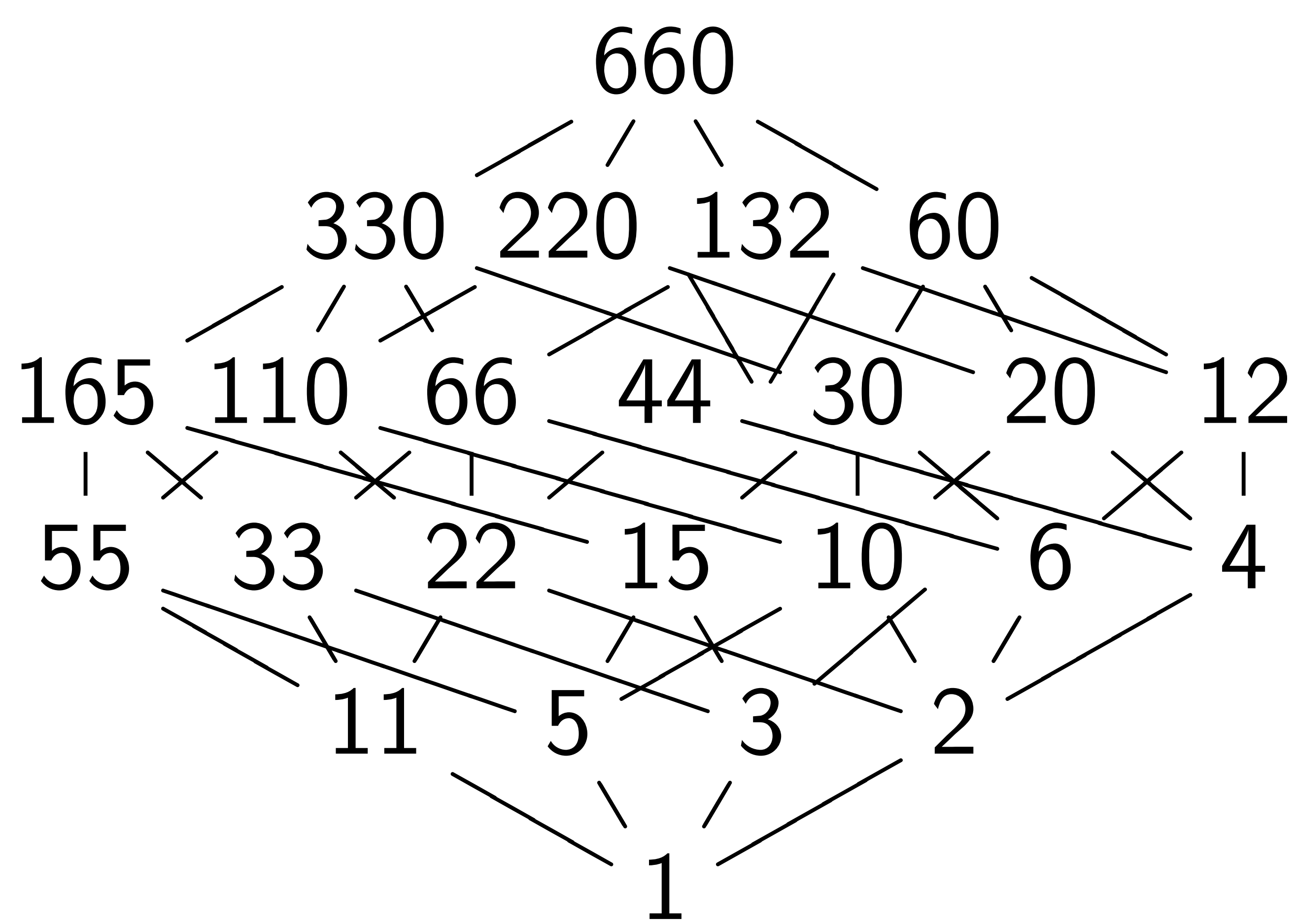


Most extreme case:
Composite of quadratics, such as $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$.
CVP becomes trivial!

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.

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



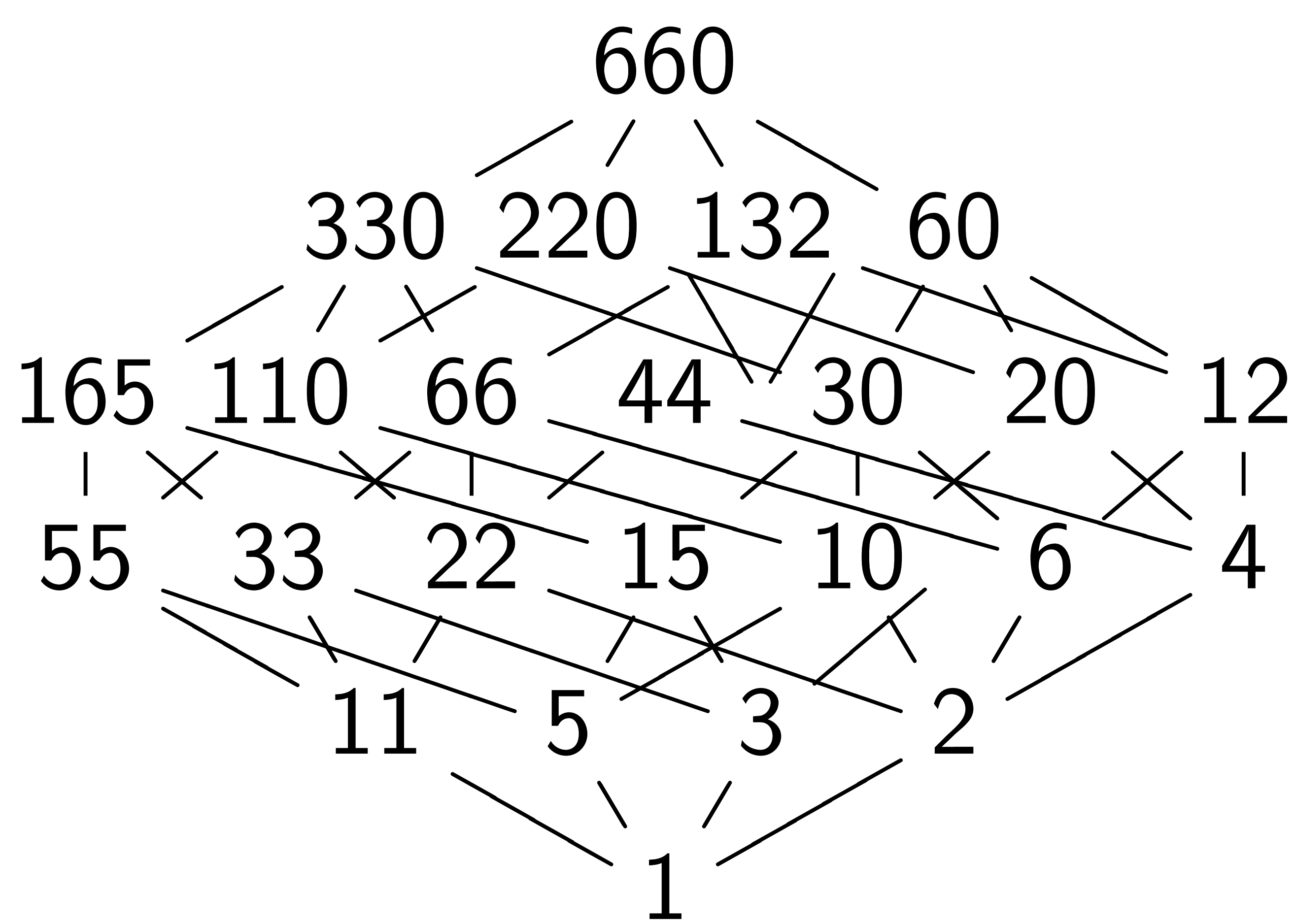
Most extreme case: Composite of quadratics, such as $K = \mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \ldots, \sqrt{29})$. CVP becomes trivial!

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

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.

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!

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.