Finding Short Generators of Ideals, and Implications for Cryptography

Chris Peikert
University of Michigan

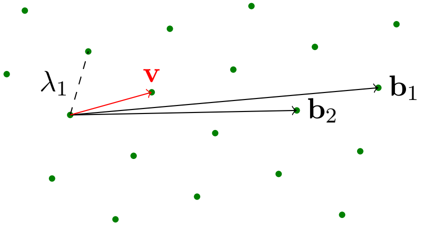
ANTS XII
29 August 2016

Based on work with Ronald Cramer, Léo Ducas, and Oded Regev

Lattice-Based Cryptography

- High-dimensional lattices in $\mathbb{R}^n$ appear to offer (quantumly) hard problems that are useful for cryptography.

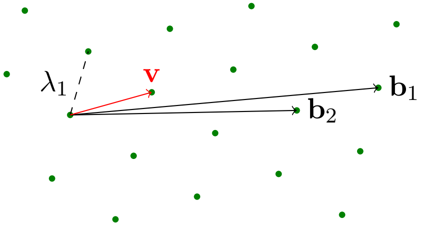
  E.g., the approximate Shortest Vector Problem:



Lattice-Based Cryptography

- High-dimensional lattices in $\mathbb{R}^n$ appear to offer (quantumly) hard problems that are useful for cryptography.

  E.g., the approximate Shortest Vector Problem:



- Cryptography requires average-case hardness: systems must be infeasible to break for random keys & outputs (w/ very high prob).

A Brief History of Lattice Cryptography

1978– 'Ad-hoc' constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, ...

A Brief History of Lattice Cryptography

1978– 'Ad-hoc' constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, ...

✗ Many broken or severely weakened due to 'Achilles heels:' random instances from the system are easier than intended.

A Brief History of Lattice Cryptography

1978– 'Ad-hoc' constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, ...

✗ Many broken or severely weakened due to 'Achilles heels:' random instances from the system are easier than intended.

1996– Worst-case to average-case reductions for lattice problems.

[Ajtai'96,AjtaiDwork'97,(Micciancio)Regev'03-'05,...]

A Brief History of Lattice Cryptography

1978– 'Ad-hoc' constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, . . .

✗ Many broken or severely weakened due to 'Achilles heels:' random instances from the system are easier than intended.

1996– Worst-case to average-case reductions for lattice problems.

[Ajtai'96,AjtaiDwork'97,(Micciancio)Regev'03-'05,. . .]

✔ Random instances are provably at least as hard as all instances of some lattice problems, via poly-time reduction.

A Brief History of Lattice Cryptography

1978– 'Ad-hoc' constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, ...

✗ Many broken or severely weakened due to 'Achilles heels:' random instances from the system are easier than intended.

1996– Worst-case to average-case reductions for lattice problems.

[Ajtai'96,AjtaiDwork'97,(Micciancio)Regev'03-'05,...]

✔ Random instances are provably at least as hard as all instances of some lattice problems, via poly-time reduction.

✗ Not so inefficient (though this is changing).

A Brief History of Lattice Cryptography

1978– ‘Ad-hoc’ constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, . . .

✗ Many broken or severely weakened due to ‘Achilles heels:’ random instances from the system are easier than intended.

1996– Worst-case to average-case reductions for lattice problems.

[Ajtai’96,AjtaiDwork’97,(Micciancio)Regev’03-’05,. . .]

✔ Random instances are provably at least as hard as all instances of some lattice problems, via poly-time reduction.

✗ Not so inefficient (though this is changing).

1996 NTRU efficient ring-based encryption: ad-hoc design, but unbroken for suitable parameters. [HoffsteinPipherSilverman’98,. . .]

A Brief History of Lattice Cryptography

1978– ‘Ad-hoc’ constructions: Merkle-Hellman, GGH/NTRU signatures, SV/Soliloquy, multilinear maps, . . .

✗ Many broken or severely weakened due to ‘Achilles heels:’ random instances from the system are easier than intended.

1996– Worst-case to average-case reductions for lattice problems.

[Ajtai’96,AjtaiDwork’97,(Micciancio)Regev’03-’05,. . .]

✔ Random instances are provably at least as hard as all instances of some lattice problems, via poly-time reduction.

✗ Not so inefficient (though this is changing).

1996 NTRU efficient ring-based encryption: ad-hoc design, but unbroken for suitable parameters. [HoffsteinPipherSilverman’98,. . .]

2002– Ring-based crypto with worst-case hardness from ideal lattices.

[Micciancio’02,LyubashevskyPeikertRegev’10,. . .]

Ideal Lattice Cryptography

1. Some ad-hoc ideal-based cryptosystems (e.g., [SV'10,GGH'13,CGS'14]) share this $\textsc{KeyGen}$:

   sk = 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.
   pk = 'Bad' $\mathbb{Z}$-basis (e.g., HNF) of the principal ideal $\mathcal{I} = gR$.

Ideal Lattice Cryptography

1. Some ad-hoc ideal-based cryptosystems (e.g., [SV'10,GGH'13,CGS'14]) share this $\textsc{KeyGen}$:

   sk = 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.
   pk = 'Bad' $\mathbb{Z}$-basis (e.g., HNF) of the principal ideal $\mathcal{I} = gR$.

2. Many systems use Learning With Errors over Rings [LyuPeiReg'10]:

$$a_1 \leftarrow R/qR \quad , \quad b_1 = s \cdot a_1 + e_1 \in R/qR$$
$$a_2 \leftarrow R/qR \quad , \quad b_2 = s \cdot a_2 + e_2 \in R/qR$$
$$\vdots$$

errors $e_i \in R$ are 'small' relative to q

Ideal Lattice Cryptography

1. Some ad-hoc ideal-based cryptosystems (e.g., [SV'10,GGH'13,CGS'14]) share this $\textsc{KeyGen}$:

   sk = 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.
   pk = 'Bad' $\mathbb{Z}$-basis (e.g., HNF) of the principal ideal $\mathcal{I} = gR$.

2. Many systems use Learning With Errors over Rings [LyuPeiReg'10]:

$$\begin{array}{lcl} a_1 \leftarrow R/qR & , & b_1 = s \cdot a_1 + e_1 \in R/qR \\ a_2 \leftarrow R/qR & , & b_2 = s \cdot a_2 + e_2 \in R/qR \\ & \vdots & \end{array}$$

errors $e_i \in R$ are 'small' relative to q

For appropriate rings and error distributions,

worst-case approx-SVP on any ideal lattice in R $\leq_{\text{quant}}$ search R-LWE $\leq$ decision R-LWE

Ideal Lattice Cryptography

1. Some ad-hoc ideal-based cryptosystems (e.g., [SV'10,GGH'13,CGS'14]) share this $\textsc{KeyGen}$:

   sk $=$ 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.
   pk $=$ 'Bad' $\mathbb{Z}$-basis (e.g., HNF) of the principal ideal $\mathcal{I} = gR$.

2. Many systems use Learning With Errors over Rings [LyuPeiReg'10]:

$$\begin{array}{lcl} a_1 \leftarrow R/qR & , & b_1 = s \cdot a_1 + e_1 \in R/qR \\ a_2 \leftarrow R/qR & , & b_2 = s \cdot a_2 + e_2 \in R/qR \\ & \vdots & \end{array}$$

errors $e_i \in R$ are 'small' relative to q

For appropriate rings and error distributions,

worst-case approx-SVP on any ideal lattice in R $\leq_{\text{quant}}$ search R-LWE $\leq$ decision R-LWE

(Note: no explicit ideals in Ring-LWE problem, only in reductions.)

Agenda

1. Finding short generators (when they exist) of principal ideals

2. Bounds for generators of arbitrary principal ideals

3. Implications for cryptography and open problems

Part 1:

Finding Short Generators
(when they exist)

Key Recovery

- Recall ad-hoc KeyGen:

  sk = 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.

  pk = 'Bad' $\mathbb{Z}$-basis (e.g., the HNF) of the principal ideal $\mathcal{I} = gR$.

Key Recovery

- Recall ad-hoc $\textsc{KeyGen}$:

  sk $=$ 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.

  pk $=$ 'Bad' $\mathbb{Z}$-basis (e.g., the HNF) of the principal ideal $\mathcal{I} = gR$.

  (Decryption works given any sufficiently short $v \in \mathcal{I}$, e.g., $g \cdot X^i$.)

Key Recovery

- Recall ad-hoc $\textsc{KeyGen}$:

  sk = 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.

  pk = 'Bad' $\mathbb{Z}$-basis (e.g., the HNF) of the principal ideal $\mathcal{I} = gR$.

  (Decryption works given any sufficiently short $v \in \mathcal{I}$, e.g., $g \cdot X^i$.)

Secret-key recovery in two steps:

Key Recovery

- Recall ad-hoc $\textsc{KeyGen}$:

  sk $=$ 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.

  pk $=$ 'Bad' $\mathbb{Z}$-basis (e.g., the HNF) of the principal ideal $\mathcal{I} = gR$.

  (Decryption works given any sufficiently short $v \in \mathcal{I}$, e.g., $g \cdot X^i$.)

Secret-key recovery in two steps:

Principal Ideal Problem

1. Given a $\mathbb{Z}$-basis $\mathbf{B}$ of a principal ideal $\mathcal{I}$, find some generator h of $\mathcal{I}$.

Key Recovery

- Recall ad-hoc $\textsc{KeyGen}$:

  sk $=$ 'Short' g in some known ring R, often $R = \mathbb{Z}[\zeta_{2^k}]$.

  pk $=$ 'Bad' $\mathbb{Z}$-basis (e.g., the HNF) of the principal ideal $\mathcal{I} = gR$.

  (Decryption works given any sufficiently short $v \in \mathcal{I}$, e.g., $g \cdot X^i$.)

Secret-key recovery in two steps:

Principal Ideal Problem

1. Given a $\mathbb{Z}$-basis $\mathbf{B}$ of a principal ideal $\mathcal{I}$, find some generator h of $\mathcal{I}$.

Short Generator Problem

2. Given an arbitrary generator h of $\mathcal{I}$, find a sufficiently short generator.

How to Perform the Steps

1. Principal Ideal Problem (PIP) has a:
   - ⋆ classical subexponential $2^{\tilde{O}(n^{2/3})}$-time algorithm [BF'14,B'14]
   - ⋆ quantum polynomial-time algorithm [EHKS'14,CGS'14,BS'14]

How to Perform the Steps

1. Principal Ideal Problem (PIP) has a:
   - ⋆ classical subexponential $2^{\tilde{O}(n^{2/3})}$-time algorithm [BF'14,B'14]
   - ⋆ quantum polynomial-time algorithm [EHKS'14,CGS'14,BS'14]

2. Short Generator Problem (SGP):
   - ⋆ Can be seen as a Closest Vector Problem in the *log-unit* lattice of R...

How to Perform the Steps

1. Principal Ideal Problem (PIP) has a:
   - classical subexponential $2^{\tilde{O}(n^{2/3})}$-time algorithm [BF'14,B'14]
   - quantum polynomial-time algorithm [EHKS'14,CGS'14,BS'14]
2. Short Generator Problem (SGP):
   - Can be seen as a Closest Vector Problem in the *log-unit* lattice of R...
   - ...but is actually Bounded Distance Decoding for $\textsc{KeyGen}$'s instances.

How to Perform the Steps

1. Principal Ideal Problem (PIP) has a:
   - ⋆ classical subexponential $2^{\tilde{O}(n^{2/3})}$-time algorithm [BF'14,B'14]
   - ⋆ quantum polynomial-time algorithm [EHKS'14,CGS'14,BS'14]
2. Short Generator Problem (SGP):
   - ⋆ Can be seen as a Closest Vector Problem in the *log-unit* lattice of R...
   - ⋆ ...but is actually Bounded Distance Decoding for KeyGen's instances.
   - ⋆ Was claimed to be easy in two-power cyclotomic rings [CamGroShe'14] and experimentally confirmed in relevant dimensions [Shank'15]

How to Perform the Steps

1. Principal Ideal Problem (PIP) has a:
   - ★ classical subexponential $2^{\tilde{O}(n^{2/3})}$-time algorithm [BF'14,B'14]
   - ★ quantum polynomial-time algorithm [EHKS'14,CGS'14,BS'14]
2. Short Generator Problem (SGP):
   - ★ Can be seen as a Closest Vector Problem in the *log-unit* lattice of R...
   - ★ ...but is actually Bounded Distance Decoding for KeyGen's instances.
   - ★ Was claimed to be easy in two-power cyclotomic rings [CamGroShe'14] and experimentally confirmed in relevant dimensions [Shank'15]

Theorem 1 [CramerDucasPeikertRegev Eurocrypt'16]

- SGP can be solved in classical polynomial time* on KeyGen's random instances for any prime-power cyclotomic ring $R = \mathbb{Z}[\zeta_{p^k}]$.

How to Perform the Steps

1. Principal Ideal Problem (PIP) has a:
   - ★ classical subexponential $2^{\tilde{O}(n^{2/3})}$-time algorithm [BF'14,B'14]
   - ★ quantum polynomial-time algorithm [EHKS'14,CGS'14,BS'14]
2. Short Generator Problem (SGP):
   - ★ Can be seen as a Closest Vector Problem in the *log-unit* lattice of R...
   - ★ ...but is actually Bounded Distance Decoding for KeyGen's instances.
   - ★ Was claimed to be easy in two-power cyclotomic rings [CamGroShe'14] and experimentally confirmed in relevant dimensions [Shank'15]

Theorem 1 [CramerDucasPeikertRegev Eurocrypt'16]

- SGP can be solved in classical polynomial time* on KeyGen's random instances for any prime-power cyclotomic ring $R = \mathbb{Z}[\zeta_{p^k}]$.

(*assuming $h^+ \leq$ poly(dim))

(Logarithmic) Embedding

Let $K \cong \mathbb{Q}[X]/f(X)$ be a number field of degree n, and let $\sigma_i \colon K \to \mathbb{C}$ be its n complex embeddings. The *canonical embedding* is the ring homom.

$$\begin{aligned} \sigma \colon K &\to \mathbb{C}^n \\ x &\mapsto (\sigma_1(x)\,,\,\ldots\,,\,\sigma_n(x)). \end{aligned}$$

(Logarithmic) Embedding

Let $K \cong \mathbb{Q}[X]/f(X)$ be a number field of degree n, and let $\sigma_i\colon K \to \mathbb{C}$ be its n complex embeddings. The *canonical embedding* is the ring homom.

$$\begin{aligned}\sigma\colon K &\to \mathbb{C}^n \\ x &\mapsto (\sigma_1(x), \ldots, \sigma_n(x)).\end{aligned}$$

The *logarithmic embedding* is

$$\begin{aligned}\mathrm{Log}\colon K^\times &\to \mathbb{R}^n \\ x &\mapsto (\log|\sigma_1(x)|, \ldots, \log|\sigma_n(x)|).\end{aligned}$$

It is a group homomorphism from $(K^\times, \times)$ to $(\mathbb{R}^n, +)$.

(Logarithmic) Embedding

Let $K \cong \mathbb{Q}[X]/f(X)$ be a number field of degree n, and let $\sigma_i \colon K \to \mathbb{C}$ be its n complex embeddings. The *canonical embedding* is the ring homom.

$$\begin{aligned} \sigma \colon K &\to \mathbb{C}^n \\ x &\mapsto (\sigma_1(x), \ldots, \sigma_n(x)). \end{aligned}$$

The *logarithmic embedding* is

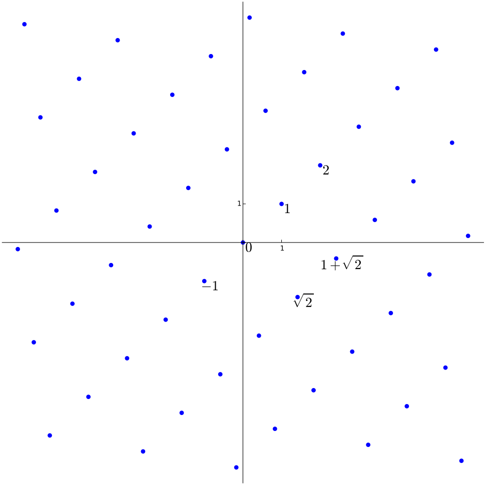
$$\begin{aligned} \mathrm{Log} \colon K^\times &\to \mathbb{R}^n \\ x &\mapsto (\log|\sigma_1(x)|, \ldots, \log|\sigma_n(x)|). \end{aligned}$$

It is a group homomorphism from $(K^\times, \times)$ to $(\mathbb{R}^n, +)$.

Example: Two-Power Cyclotomics

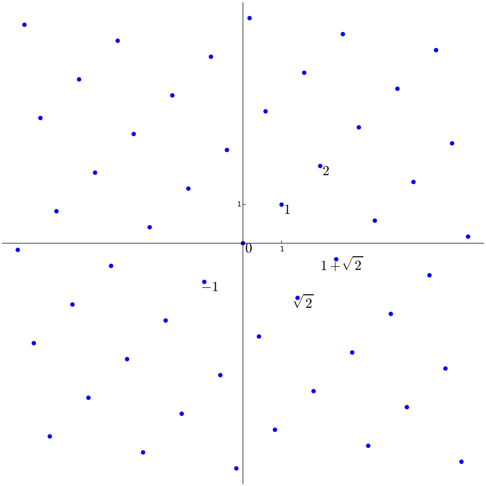
- $K \cong \mathbb{Q}[X]/(X^n + 1)$ for $n = 2^k$.
- $\sigma_i(X) = \omega^{2i-1}$, where $\omega = \exp(\pi\sqrt{-1}/n) \in \mathbb{C}$.
- $\mathrm{Log}(X^j) = \mathbf{0}$ for all j.

Example: Embedding $\sigma(\mathbb{Z}[\sqrt{2}]) \to \mathbb{R}^2$



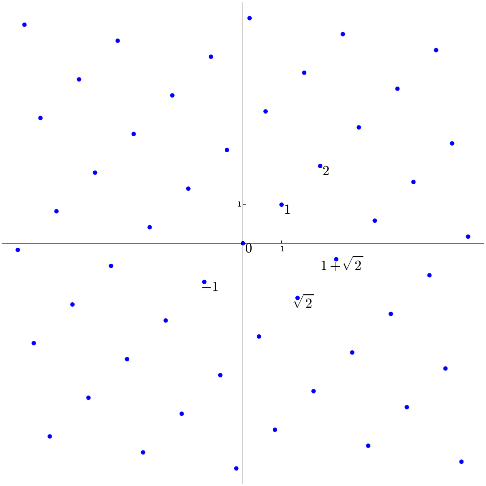
- x-axis: $\sigma_1(a + b\sqrt{2}) = a + b\sqrt{2}$
- y-axis: $\sigma_2(a + b\sqrt{2}) = a - b\sqrt{2}$

Example: Embedding $\sigma(\mathbb{Z}[\sqrt{2}]) \to \mathbb{R}^2$



- x-axis: $\sigma_1(a+b\sqrt{2}) = a+b\sqrt{2}$
- y-axis: $\sigma_2(a+b\sqrt{2}) = a-b\sqrt{2}$
- component-wise multiplication

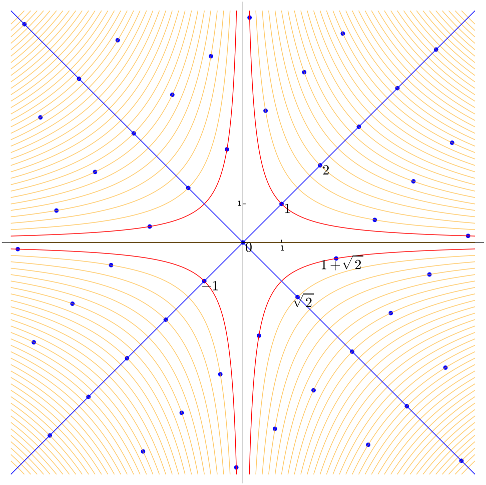
Example: Embedding $\sigma(\mathbb{Z}[\sqrt{2}]) \to \mathbb{R}^2$



- x-axis: $\sigma_1(a + b\sqrt{2}) = a + b\sqrt{2}$
- y-axis: $\sigma_2(a + b\sqrt{2}) = a - b\sqrt{2}$
- component-wise multiplication

- Symmetries induced by
  - multiplication by $-1, \sqrt{2}$
  - conjugation $\sqrt{2} \mapsto -\sqrt{2}$

Example: Embedding $\sigma(\mathbb{Z}[\sqrt{2}]) \to \mathbb{R}^2$



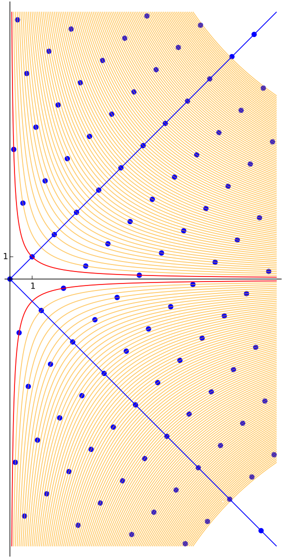
- x-axis: $\sigma_1(a + b\sqrt{2}) = a + b\sqrt{2}$
- y-axis: $\sigma_2(a + b\sqrt{2}) = a - b\sqrt{2}$
- component-wise multiplication

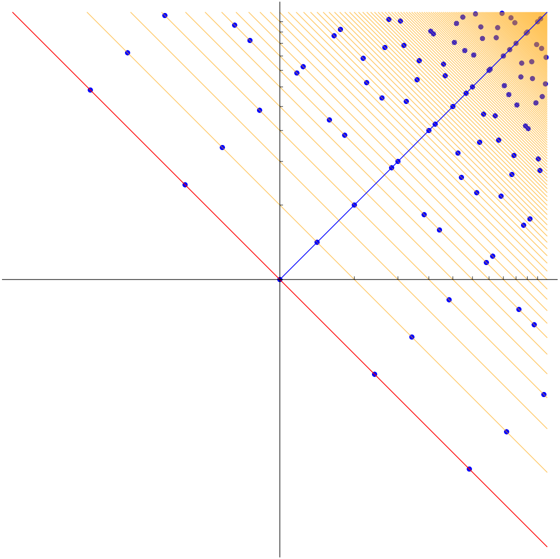
- Symmetries induced by
  - ★ multiplication by $-1, \sqrt{2}$
  - ★ conjugation $\sqrt{2} \mapsto -\sqrt{2}$

- Orthogonal lattice axes
- Units (algebraic norm 1)
- "Isonorms"

Example: Logarithmic Embedding $\operatorname{Log}\mathbb{Z}[\sqrt{2}]$

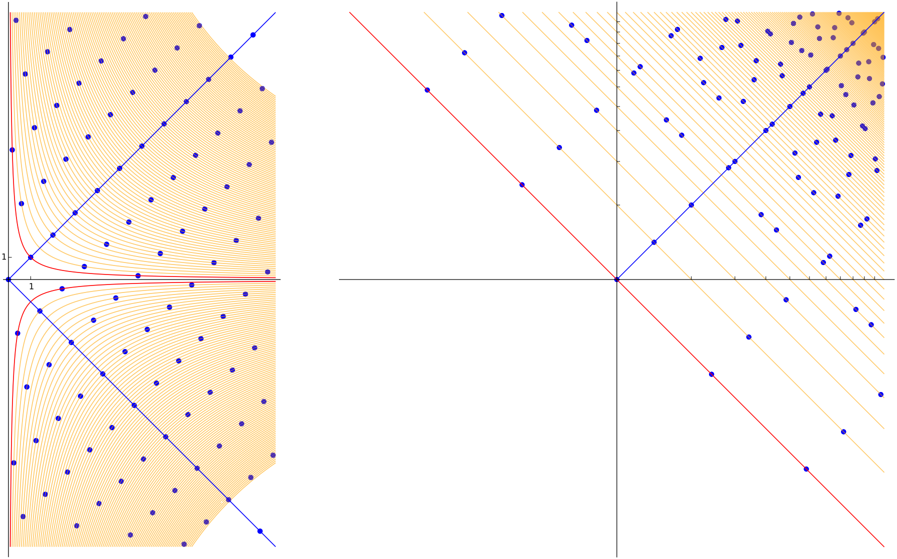
$\Lambda = \{\bullet\} \cap$ ╲ is a rank-1 lattice $\Lambda \subset \mathbb{R}^2$, orthogonal to $\mathbf{1}$





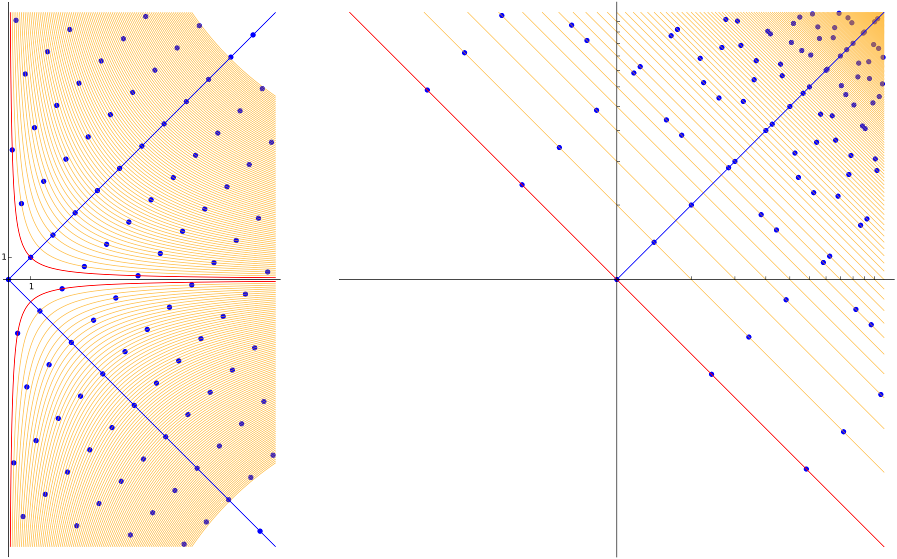
Example: Logarithmic Embedding $\operatorname{Log} \mathbb{Z}[\sqrt{2}]$

$\{\bullet\} \cap \diagdown$ are finite $\#$ of shifted copies (cosets) of Λ



Example: Logarithmic Embedding $\operatorname{Log} \mathbb{Z}[\sqrt{2}]$

Some $\{\bullet\} \cap \diagdown$ may be empty (e.g., no elements of norm 3)



Unit Group and the Log-Unit Lattice

Let $R^\times$ denote the mult. group of units of R, and $\Lambda = \operatorname{Log} R^\times \subset \mathbb{R}^n$.

Unit Group and the Log-Unit Lattice

Let $R^\times$ denote the mult. group of units of R, and $\Lambda = \operatorname{Log} R^\times \subset \mathbb{R}^n$.

Dirichlet Unit Theorem

- The kernel of Log is the cyclic subgroup of roots of unity in $R^\times$, and
- $\Lambda \subset \mathbb{R}^n$ is a lattice of rank $r + c - 1$, orthogonal to $\mathbf{1}$

(where K has r real embeddings and $2c$ complex embeddings)

Unit Group and the Log-Unit Lattice

Let $R^\times$ denote the mult. group of units of R, and $\Lambda = \operatorname{Log} R^\times \subset \mathbb{R}^n$.

Dirichlet Unit Theorem

- The kernel of Log is the cyclic subgroup of roots of unity in $R^\times$, and
- $\Lambda \subset \mathbb{R}^n$ is a lattice of rank $r + c - 1$, orthogonal to $\mathbf{1}$

(where K has r real embeddings and $2c$ complex embeddings)

Shortest Generators from Bounded Distance Decoding

Elements $g, h \in R$ generate the same ideal if and only if $g = h \cdot u$ for some unit $u \in R^\times$, i.e.,

$$\operatorname{Log} g = \operatorname{Log} h + \operatorname{Log} u \in \operatorname{Log} h + \Lambda.$$

Unit Group and the Log-Unit Lattice

Let $R^\times$ denote the mult. group of units of R, and $\Lambda = \operatorname{Log} R^\times \subset \mathbb{R}^n$.

Dirichlet Unit Theorem

- The kernel of Log is the cyclic subgroup of roots of unity in $R^\times$, and
- $\Lambda \subset \mathbb{R}^n$ is a lattice of rank $r + c - 1$, orthogonal to $\mathbf{1}$

(where K has r real embeddings and $2c$ complex embeddings)

Shortest Generators from Bounded Distance Decoding

Elements $g, h \in R$ generate the same ideal if and only if $g = h \cdot u$ for some unit $u \in R^\times$, i.e.,

$$\operatorname{Log} g = \operatorname{Log} h + \operatorname{Log} u \in \operatorname{Log} h + \Lambda.$$

- By $\textsc{KeyGen}$, we know that $\operatorname{Log} h + \Lambda$ has a 'short' $\mathbf{g} = \operatorname{Log} g$.

Unit Group and the Log-Unit Lattice

Let $R^\times$ denote the mult. group of units of R, and $\Lambda = \operatorname{Log} R^\times \subset \mathbb{R}^n$.

Dirichlet Unit Theorem

- The kernel of Log is the cyclic subgroup of roots of unity in $R^\times$, and
- $\Lambda \subset \mathbb{R}^n$ is a lattice of rank $r + c - 1$, orthogonal to $\mathbf{1}$

(where K has r real embeddings and $2c$ complex embeddings)

Shortest Generators from Bounded Distance Decoding

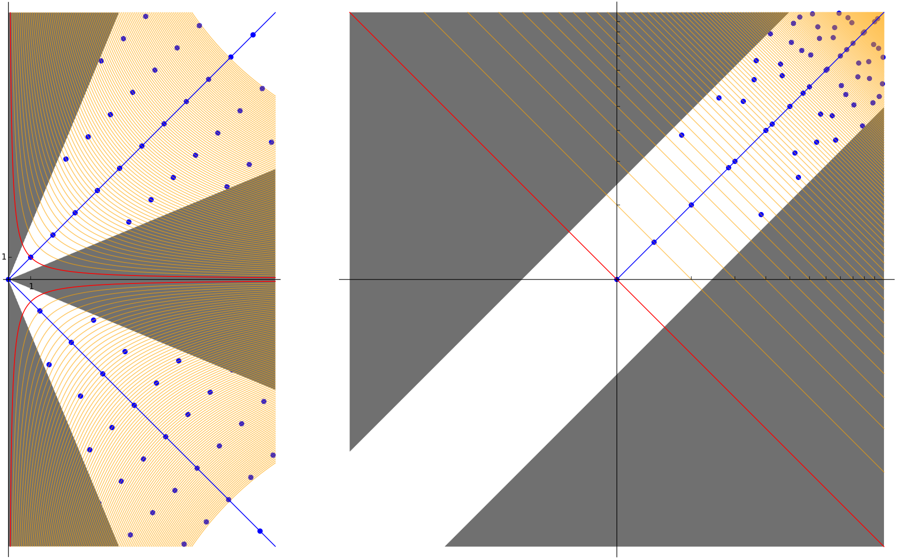
Elements $g, h \in R$ generate the same ideal if and only if $g = h \cdot u$ for some unit $u \in R^\times$, i.e.,

$$\operatorname{Log} g = \operatorname{Log} h + \operatorname{Log} u \in \operatorname{Log} h + \Lambda.$$

- By $\textsc{KeyGen}$, we know that $\operatorname{Log} h + \Lambda$ has a 'short' $\mathbf{g} = \operatorname{Log} g$.
- Our goal is to 'decode' such $\mathbf{g}$, yielding g (up to roots of unity).

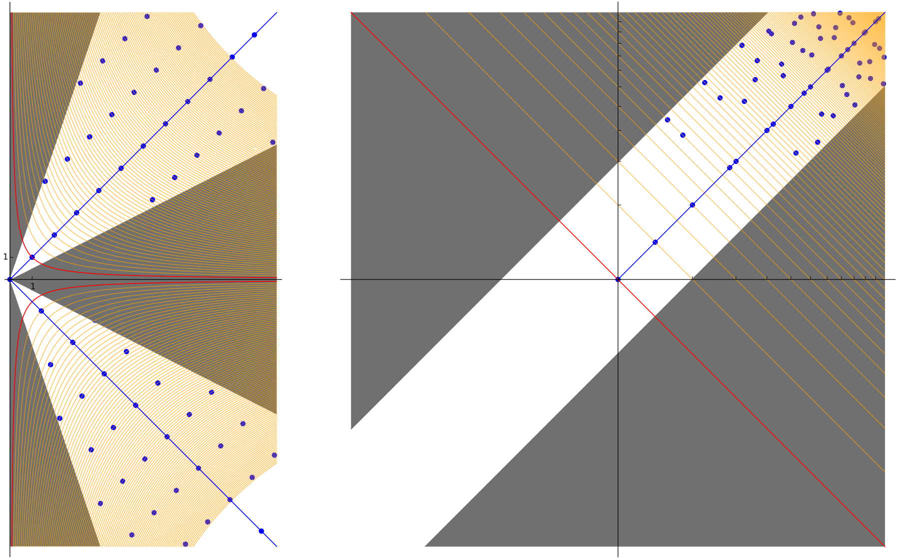
Decoding $\Lambda = \operatorname{Log} \mathbb{Z}[\sqrt{2}]^{\times}$

Decoding cosets $\mathbf{h} + \Lambda$ into various fundamental domains of Λ.



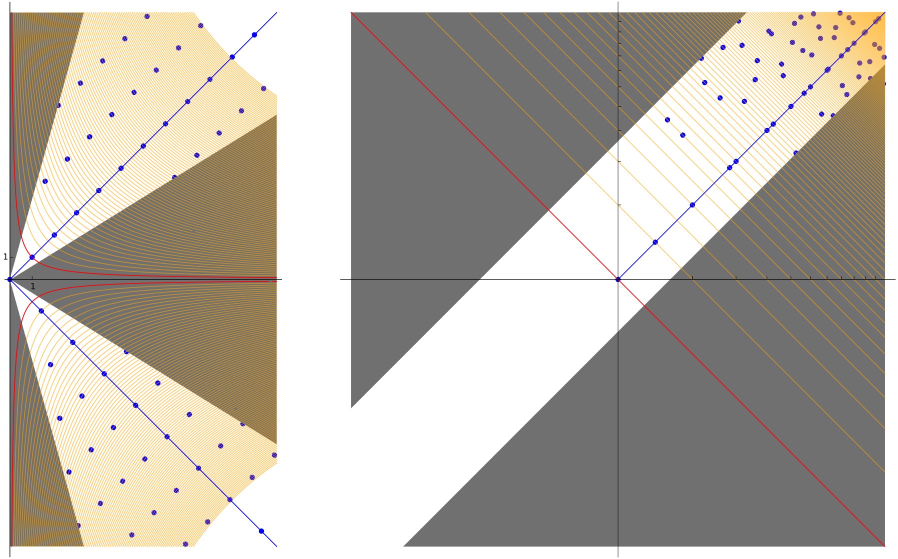
Decoding $\Lambda = \operatorname{Log} \mathbb{Z}[\sqrt{2}]^{\times}$

Decoding cosets $\mathbf{h} + \Lambda$ into various fundamental domains of Λ.



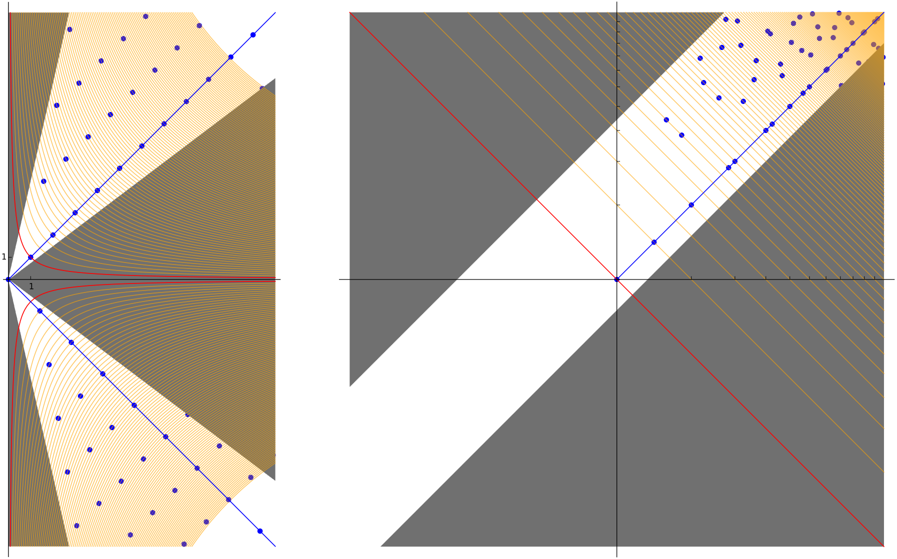
Decoding $\Lambda = \operatorname{Log} \mathbb{Z}[\sqrt{2}]^{\times}$

Decoding cosets $\mathbf{h} + \Lambda$ into various fundamental domains of Λ.



Decoding $\Lambda = \operatorname{Log} \mathbb{Z}[\sqrt{2}]^{\times}$

Decoding cosets $\mathbf{h} + \Lambda$ into various fundamental domains of Λ.

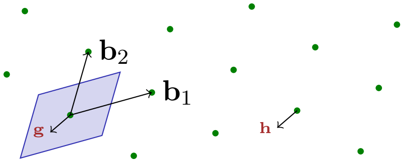


Round-Off Decoding

The simplest lattice-decoding algorithm:

$\textsc{Round}(\mathbf{B}, \mathbf{h})$ for a basis $\mathbf{B}$ of Λ and $\mathbf{h} \in \mathbb{R}^n$

- Return $\mathbf{B} \cdot \text{frac}(\mathbf{B}^{-1} \cdot \mathbf{h})$, where frac: $\mathbb{R}^n \to [-\frac{1}{2}, \frac{1}{2})^n$.

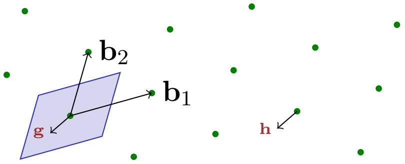


Round-Off Decoding

The simplest lattice-decoding algorithm:

$\textsc{Round}(\mathbf{B}, \mathbf{h})$ for a basis $\mathbf{B}$ of Λ and $\mathbf{h} \in \mathbb{R}^n$

- Return $\mathbf{B} \cdot \mathrm{frac}(\mathbf{B}^{-1} \cdot \mathbf{h})$, where $\mathrm{frac}\colon \mathbb{R}^n \to [-\frac{1}{2}, \frac{1}{2})^n$.



Behavior is characterized by the 'offset' and the *dual basis* $\mathbf{B}^\vee = \mathbf{B}^{-t}$.

Trivial Fact

Suppose $\mathbf{h} = \mathbf{g} + \mathbf{u}$ for some $\mathbf{u} \in \Lambda$. If $\langle \mathbf{b}_j^\vee, \mathbf{g} \rangle \in [-\frac{1}{2}, \frac{1}{2})$ for all j, then

$$\textsc{Round}(\mathbf{B}, \mathbf{h}) = \mathbf{g}.$$

Recovering a Short Generator: Proof Outline

1. Obtain a "good" basis $\mathbf{B}$ of the log-unit lattice $\Lambda = \mathrm{Log}\, R^{\times}$.
   - ⋆ For $K = \mathbb{Q}(\zeta_m)$, $m = p^k$, a standard (almost[1]-)basis of Λ is given by

$$\mathbf{b}_j = \mathrm{Log}\, \frac{1 - \zeta^j}{1 - \zeta}, \quad 1 < j < m/2,\ \gcd(j, m) = 1.$$

[1] it generates a sublattice of finite index h^+, which is conjectured to be small.

Recovering a Short Generator: Proof Outline

1. Obtain a "good" basis $\mathbf{B}$ of the log-unit lattice $\Lambda = \operatorname{Log} R^{\times}$.
   - ⋆ For $K = \mathbb{Q}(\zeta_m)$, $m = p^k$, a standard (almost[1]-)basis of Λ is given by

$$\mathbf{b}_j = \operatorname{Log} \frac{1-\zeta^j}{1-\zeta}, \quad 1 < j < m/2,\ \gcd(j, m) = 1.$$

2. Prove that $\mathbf{B}$ is "good," i.e., all $\|\mathbf{b}_j^\vee\|$ are small.

[1] it generates a sublattice of finite index h^+, which is conjectured to be small.

Recovering a Short Generator: Proof Outline

1. Obtain a "good" basis $\mathbf{B}$ of the log-unit lattice $\Lambda = \operatorname{Log} R^{\times}$.
   - ⋆ For $K = \mathbb{Q}(\zeta_m)$, $m = p^k$, a standard (almost[1]-)basis of Λ is given by

$$\mathbf{b}_j = \operatorname{Log} \frac{1 - \zeta^j}{1 - \zeta}, \quad 1 < j < m/2, \ \gcd(j, m) = 1.$$

2. Prove that $\mathbf{B}$ is "good," i.e., all $\|\mathbf{b}_j^{\vee}\|$ are small.
3. Prove that $\mathbf{g} = \operatorname{Log} g$ from KeyGen is sufficiently small, so that $\langle \mathbf{b}_j^{\vee}, \mathbf{g} \rangle \in [-\frac{1}{2}, \frac{1}{2})$ and round-off decoding yields $\mathbf{g}$.

[1] it generates a sublattice of finite index h^+, which is conjectured to be small.

Recovering a Short Generator: Proof Outline

1. Obtain a "good" basis $\mathbf{B}$ of the log-unit lattice $\Lambda = \operatorname{Log} R^{\times}$.
   - ★ For $K = \mathbb{Q}(\zeta_m)$, $m = p^k$, a standard (almost[1]-)basis of Λ is given by

$$\mathbf{b}_j = \operatorname{Log} \frac{1-\zeta^j}{1-\zeta}, \quad 1 < j < m/2,\ \gcd(j,m) = 1.$$

2. Prove that $\mathbf{B}$ is "good," i.e., all $\|\mathbf{b}_j^\vee\|$ are small.
3. Prove that $\mathbf{g} = \operatorname{Log} g$ from KeyGen is sufficiently small, so that $\langle \mathbf{b}_j^\vee, \mathbf{g}\rangle \in [-\frac{1}{2}, \frac{1}{2})$ and round-off decoding yields $\mathbf{g}$.

Technical Steps

- Bound $\|\mathbf{b}_j^\vee\| = \tilde{O}(1/\sqrt{m})$ using Gauss sums and Dirichlet L-series.
- Bound $|\langle \mathbf{b}_j^\vee, \mathbf{g}\rangle| \ll \frac{1}{2}$ via subexponential random variables.

[1] it generates a sublattice of finite index h^+, which is conjectured to be small.

Part 2:

Bounds for Generators of Arbitrary Principal Ideals

Average Case Versus Worst Case

- Cryptanalysis of KeyGen exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

Average Case Versus Worst Case

- Cryptanalysis of $\textsc{KeyGen}$ exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

  How (a)typical are such principal ideals?

Average Case Versus Worst Case

- Cryptanalysis of KeyGen exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

  How (a)typical are such principal ideals?

- Recall that breaking Ring-LWE implies (quantumly) solving approx-SVP, usually for small poly factors, on any ideal in the ring.

Average Case Versus Worst Case

- Cryptanalysis of KeyGen exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

  How (a)typical are such principal ideals?

- Recall that breaking Ring-LWE implies (quantumly) solving approx-SVP, usually for small poly factors, on any ideal in the ring.

  Do log-unit attacks apply to this problem? To Ring-LWE itself?

Average Case Versus Worst Case

- Cryptanalysis of KeyGen exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

  How (a)typical are such principal ideals?

- Recall that breaking Ring-LWE implies (quantumly) solving approx-SVP, usually for small poly factors, on any ideal in the ring.

  Do log-unit attacks apply to this problem? To Ring-LWE itself?

In cyclotomic rings $R = \mathbb{Z}[\zeta_m]$ of prime-power conductor m:

Average Case Versus Worst Case

- Cryptanalysis of $\textsc{KeyGen}$ exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

  How (a)typical are such principal ideals?

- Recall that breaking Ring-LWE implies (quantumly) solving approx-SVP, usually for small poly factors, on any ideal in the ring.

  Do log-unit attacks apply to this problem? To Ring-LWE itself?

In cyclotomic rings $R = \mathbb{Z}[\zeta_m]$ of prime-power conductor m:

Upper Bound [CDPR'16]

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx SVP on $\mathcal{I}$.

Average Case Versus Worst Case

- Cryptanalysis of KeyGen exploited the promise that the public principal ideal has a 'quite short' generator g (for BDD on $\mathbf{g} + \Lambda$).

  How (a)typical are such principal ideals?

- Recall that breaking Ring-LWE implies (quantumly) solving approx-SVP, usually for small poly factors, on any ideal in the ring.

  Do log-unit attacks apply to this problem? To Ring-LWE itself?

In cyclotomic rings $R = \mathbb{Z}[\zeta_m]$ of prime-power conductor m:

Upper Bound [CDPR'16]

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx SVP on $\mathcal{I}$.

Lower Bound [CDPR'16]

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

Proof Outline: Upper Bound

Theorem

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx-SVP on $\mathcal{I}$.

Proof Outline: Upper Bound

Theorem

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx-SVP on $\mathcal{I}$.

(Cf. average case from $\textsc{KeyGen}$, where we solve SVP exactly.)

Proof Outline: Upper Bound

Theorem

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx-SVP on $\mathcal{I}$.

(Cf. average case from $\textsc{KeyGen}$, where we solve SVP exactly.)

- For principal ideal hR, the generators have log-embeddings $\operatorname{Log} h + \Lambda$.

Proof Outline: Upper Bound

Theorem

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx-SVP on $\mathcal{I}$.

(Cf. average case from $\textsc{KeyGen}$, where we solve SVP exactly.)

- For principal ideal hR, the generators have log-embeddings $\operatorname{Log} h + \Lambda$.
- Make ℓ_∞ norm of $\operatorname{Log} g \in \operatorname{Log} h + \Lambda$ small to get a short-ish generator.

  (Note: $\langle \operatorname{Log} g, \mathbf{1} \rangle = \log \mathrm{N}(\mathcal{I})$ for all generators g.)

Proof Outline: Upper Bound

Theorem

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx-SVP on $\mathcal{I}$.

(Cf. average case from KeyGen, where we solve SVP exactly.)

- For principal ideal hR, the generators have log-embeddings $\operatorname{Log} h + \Lambda$.
- Make ℓ_∞ norm of $\operatorname{Log} g \in \operatorname{Log} h + \Lambda$ small to get a short-ish generator.

  (Note: $\langle \operatorname{Log} g, \mathbf{1} \rangle = \log \mathrm{N}(\mathcal{I})$ for all generators g.)
- A simple randomized round-off algorithm using the "good" (almost-)basis $\mathbf{B}$ of Λ yields

$$\|\operatorname{Log} g\|_\infty \leq O(\sqrt{m \log m}) + \tfrac{1}{n} \log \mathrm{N}(\mathcal{I}).$$

Proof Outline: Upper Bound

Theorem

Given any generator of a principal ideal $\mathcal{I}$ (e.g., via quantum PIP algorithm), we can efficiently solve $2^{O(\sqrt{m \log m})}$-approx-SVP on $\mathcal{I}$.

(Cf. average case from $\textsc{KeyGen}$, where we solve SVP exactly.)

- For principal ideal hR, the generators have log-embeddings $\mathrm{Log}\, h + \Lambda$.
- Make ℓ_∞ norm of $\mathrm{Log}\, g \in \mathrm{Log}\, h + \Lambda$ small to get a short-ish generator.

  (Note: $\langle \mathrm{Log}\, g, \mathbf{1} \rangle = \log \mathrm{N}(\mathcal{I})$ for all generators g.)
- A simple randomized round-off algorithm using the "good" (almost-)basis $\mathbf{B}$ of Λ yields

$$\|\mathrm{Log}\, g\|_\infty \leq O(\sqrt{m \log m}) + \tfrac{1}{n} \log \mathrm{N}(\mathcal{I}).$$

- Therefore, $\|g\| \leq 2^{O(\sqrt{m \log m})} \cdot \mathrm{N}(\mathcal{I})^{1/n} \leq 2^{O(\sqrt{m \log m})} \cdot \lambda_1(\mathcal{I})$.

Proof Outline: Lower Bound

Theorem

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

Proof Outline: Lower Bound

Theorem

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

So returning a generator yields $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx, in the worst case.

Proof Outline: Lower Bound

Theorem

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

So returning a generator yields $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx, in the worst case.

- For any $\mathbf{g} = \operatorname{Log} g \in \operatorname{span}(\Lambda)$, some coordinate is $\geq s = \|\mathbf{g}\|_1/(2n)$, so $\|g\| \geq \exp(s)$.

Proof Outline: Lower Bound

Theorem

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

So returning a generator yields $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx, in the worst case.

- For any $\mathbf{g} = \operatorname{Log} g \in \operatorname{span}(\Lambda)$, some coordinate is $\geq s = \|\mathbf{g}\|_1/(2n)$, so $\|g\| \geq \exp(s)$.
- Therefore, we care about the ℓ_1 covering radius:

$$\mu_1(\Lambda) := \max_{\mathbf{h} \in \operatorname{span}(\Lambda)} \min_{\mathbf{g} \in \mathbf{h}+\Lambda} \|\mathbf{g}\|_1.$$

Proof Outline: Lower Bound

Theorem

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

So returning a generator yields $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx, in the worst case.

- For any $\mathbf{g} = \mathrm{Log}\, g \in \mathrm{span}(\Lambda)$, some coordinate is $\geq s = \|\mathbf{g}\|_1/(2n)$, so $\|g\| \geq \exp(s)$.
- Therefore, we care about the ℓ_1 covering radius:

$$\mu_1(\Lambda) := \max_{\mathbf{h} \in \mathrm{span}(\Lambda)} \min_{\mathbf{g} \in \mathbf{h}+\Lambda} \|\mathbf{g}\|_1.$$

In the worst case, a shortest generator approximates SVP to only a $\exp(\Omega(\mu_1(\Lambda)/n))$ factor.

Proof Outline: Lower Bound

Theorem

For "most" principal ideals, their shortest generators are only $2^{\Omega(\sqrt{m}/\log m)}$ SVP approximations. (Assuming $h^+ = 2^{O(m)}$.)

So returning a generator yields $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx, in the worst case.

- For any $\mathbf{g} = \operatorname{Log} g \in \operatorname{span}(\Lambda)$, some coordinate is $\geq s = \|\mathbf{g}\|_1/(2n)$, so $\|g\| \geq \exp(s)$.
- Therefore, we care about the ℓ_1 covering radius:

$$\mu_1(\Lambda) := \max_{\mathbf{h} \in \operatorname{span}(\Lambda)} \min_{\mathbf{g} \in \mathbf{h}+\Lambda} \|\mathbf{g}\|_1.$$

  In the worst case, a shortest generator approximates SVP to only a $\exp(\Omega(\mu_1(\Lambda)/n))$ factor.
- Bound $\mu_1(\Lambda) \geq \Omega(m^{3/2}/\log m)$ using volume argument.

Open Problems

1. Extend to non-principal ideals. [CramerDucasWesolowski'16, preprint]

Open Problems

1. Extend to non-principal ideals. [CramerDucasWesolowski'16, preprint]
2. Extend to proposed non-cyclotomic number fields, e.g., $R = \mathbb{Z}[X]/(X^p - X - 1)$. [Bernstein'14]

   Seems sufficient to find a 'good' full-rank set in $\Lambda = \operatorname{Log} R^\times$.

   It's easy to find several 'good' units; full rank is unclear.

Open Problems

1. Extend to non-principal ideals. [CramerDucasWesolowski'16, preprint]
2. Extend to proposed non-cyclotomic number fields, e.g., $R = \mathbb{Z}[X]/(X^p - X - 1)$. [Bernstein'14]

   Seems sufficient to find a 'good' full-rank set in $\Lambda = \operatorname{Log} R^\times$.

   It's easy to find several 'good' units; full rank is unclear.
3. Circumvent the $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx barrier for generators.

   Find a short generator of a cleverly chosen ideal $\mathcal{I}\mathcal{J}$?

Open Problems

1. Extend to non-principal ideals. [CramerDucasWesolowski'16, preprint]
2. Extend to proposed non-cyclotomic number fields, e.g., $R = \mathbb{Z}[X]/(X^p - X - 1)$. [Bernstein'14]

   Seems sufficient to find a 'good' full-rank set in $\Lambda = \operatorname{Log} R^\times$.

   It's easy to find several 'good' units; full rank is unclear.
3. Circumvent the $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx barrier for generators.

   Find a short generator of a cleverly chosen ideal $\mathcal{I}\mathcal{J}$?
4. Apply or extend any of these techniques against NTRU/Ring-LWE.

   Ideal-SVP is a lower bound for Ring-LWE; is it an upper bound?

Open Problems

1. Extend to non-principal ideals. [CramerDucasWesolowski'16, preprint]
2. Extend to proposed non-cyclotomic number fields, e.g., $R = \mathbb{Z}[X]/(X^p - X - 1)$. [Bernstein'14]

   Seems sufficient to find a 'good' full-rank set in $\Lambda = \operatorname{Log} R^\times$.

   It's easy to find several 'good' units; full rank is unclear.
3. Circumvent the $2^{\tilde{\Omega}(\sqrt{m})}$ SVP approx barrier for generators.

   Find a short generator of a cleverly chosen ideal $\mathcal{I}\mathcal{J}$?
4. Apply or extend any of these techniques against NTRU/Ring-LWE.

   Ideal-SVP is a lower bound for Ring-LWE; is it an upper bound?

Thanks!