RUB

RUHR-UNIVERSITÄT BOCHUM

Parallel Implementation of BDD enumeration for LWE

ACNS, 22.06.16

E. Kirshanova **A. May** **F. Wiemer**
Horst Görtz Institute for IT Security
Ruhr University Bochum

hgi
Horst Görtz Institut
für IT-Sicherheit

Goal

Q: How hard is the Learning with Errors in practice?

Theory: best Run-time & Memory trade-off (BKW, sieving)

This work: *practical* cryptanalysis for LWE

Outline of the results

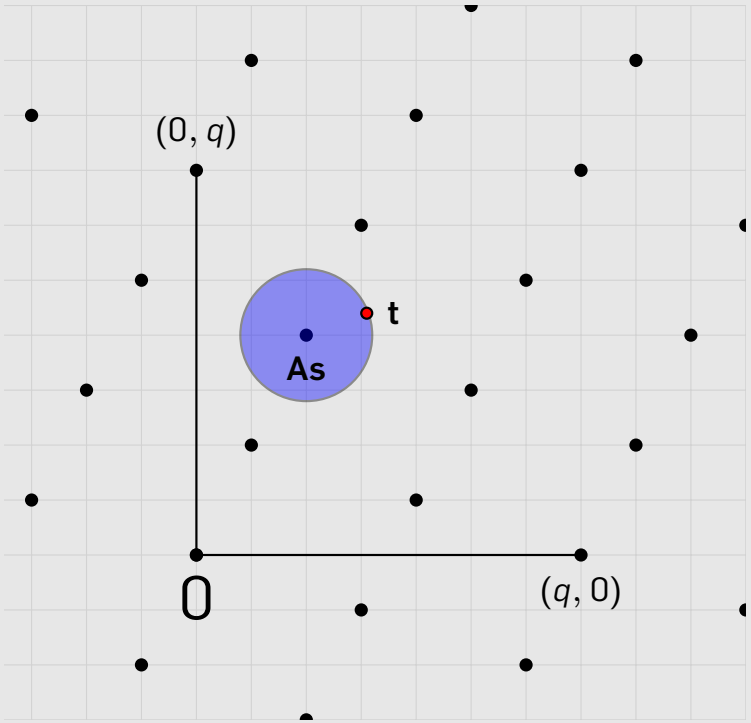
We present *experimental* results on

- Standard LWE parameters (Gaussian noise, random secret)
- 'Special' LWE parameters (binary noise, binary secret)
- Binary-matrix LWE

We make use of parallellized enumeration for the LWE problem

Bounded Distance Decoding = LWE

Given ($\mathbf{A} \in \mathbb{Z}_q^{m \times n}, \mathbf{t} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \bmod q, \|\mathbf{e}\| - \text{small}$), find $\mathbf{s}$.

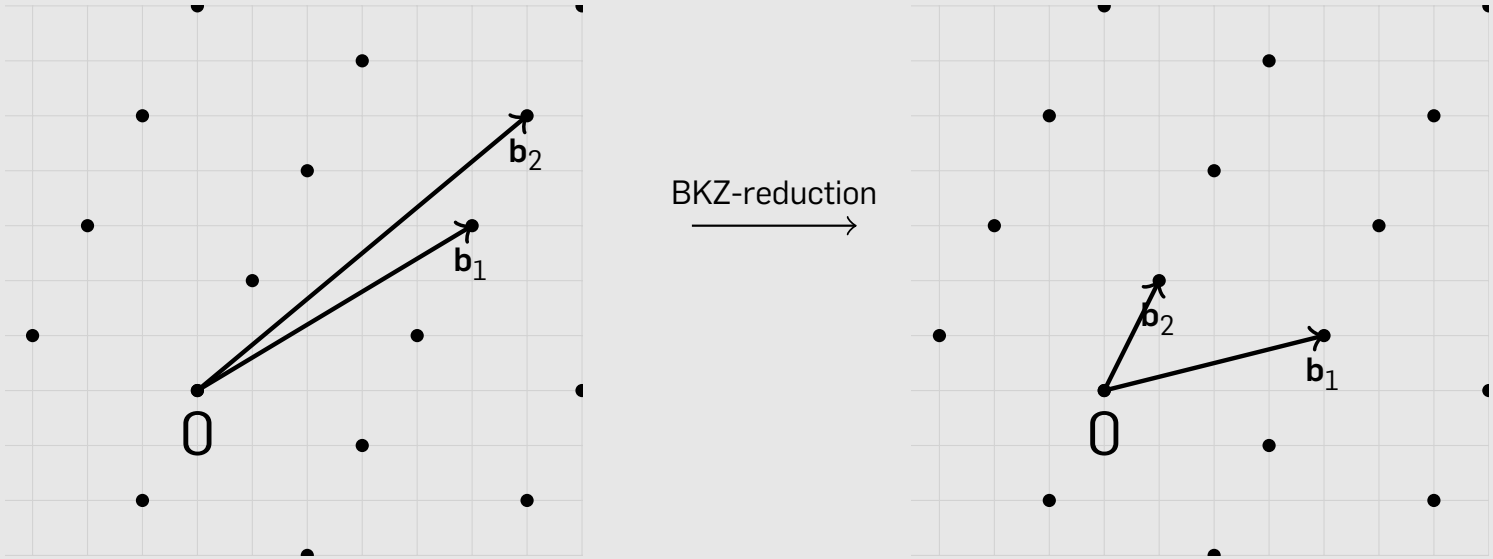


3 parameters: $n, q, \|\mathbf{e}\|$.

Algorithm

Lattice-basis reduction + Enumeration

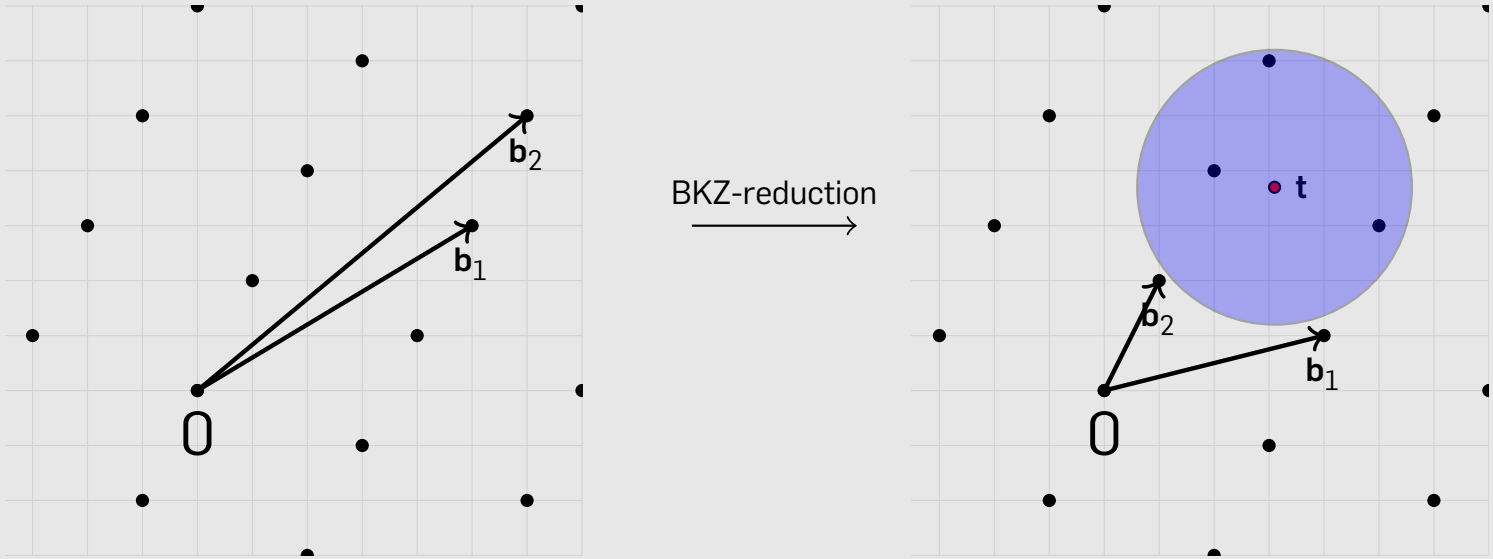
Step 1: Find a ‘good’ basis for $\Lambda_q(\mathbf{A})$:



Algorithm

Lattice-basis reduction + Enumeration

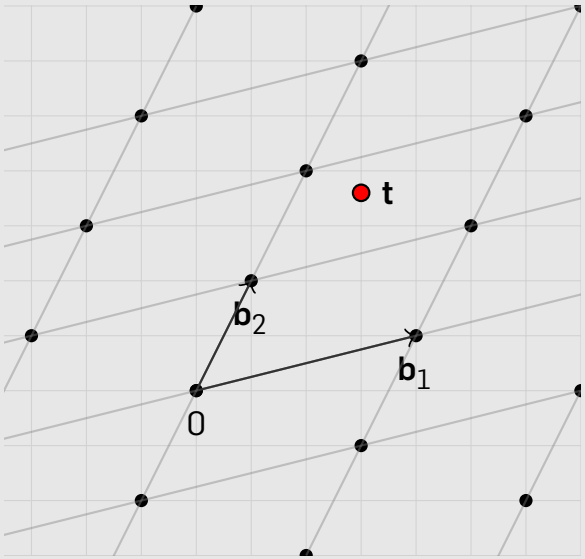
Step 1: Find a ‘good’ basis for $\Lambda_q(\mathbf{A})$:



Step 2: Consider all points within radius $\|\mathbf{e}\|$ to $\mathbf{t}$

Enumeration

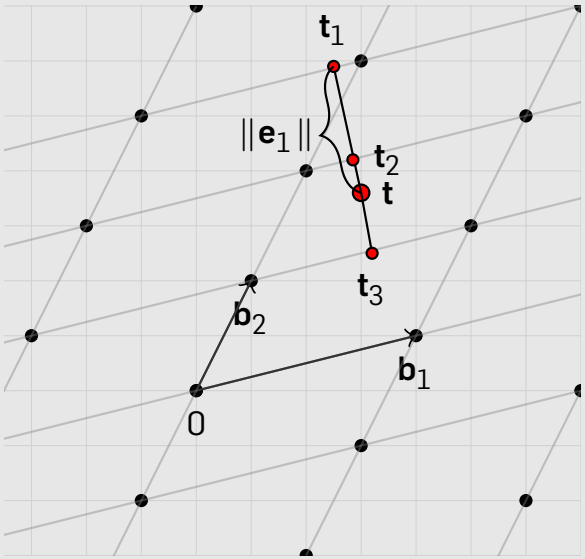
Closest point search via depth-first tree-traversal:

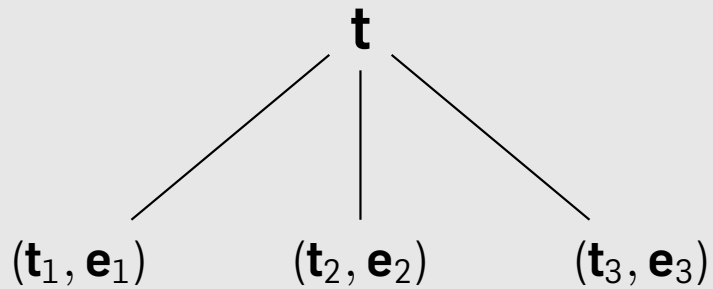


t

Enumeration

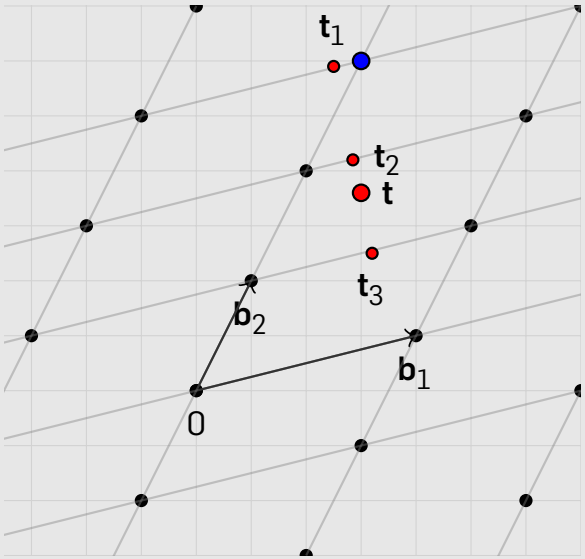
Closest point search via depth-first tree-traversal:

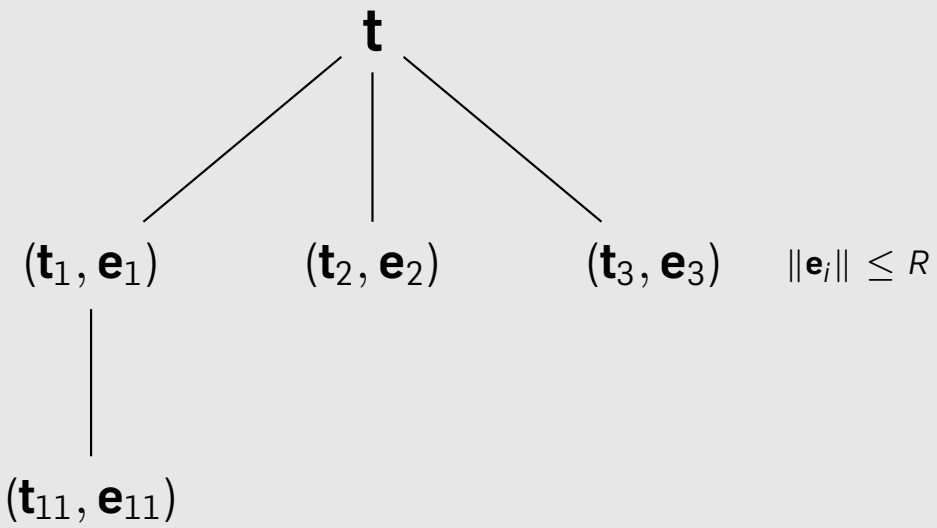




Enumeration

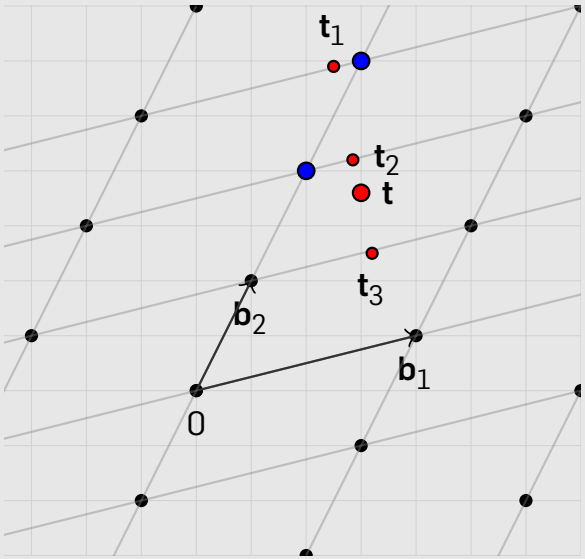
Closest point search via depth-first tree-traversal:

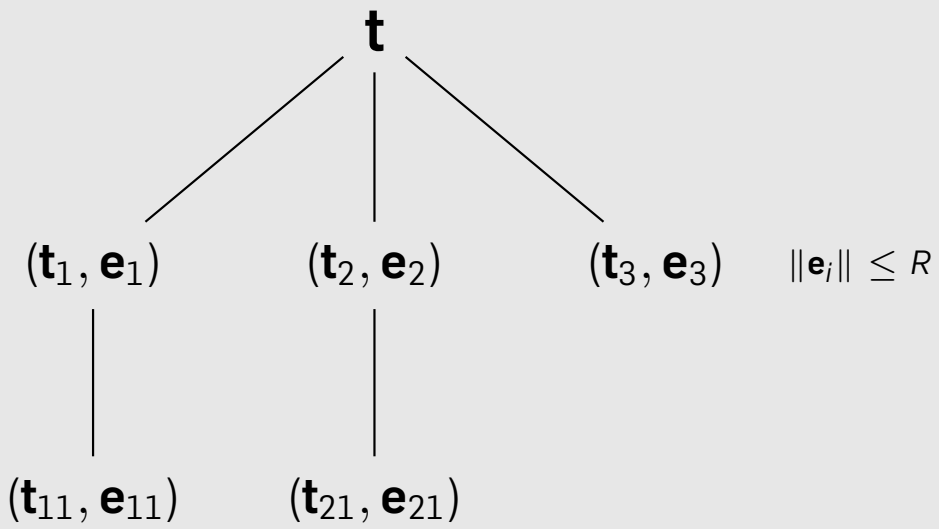




Enumeration

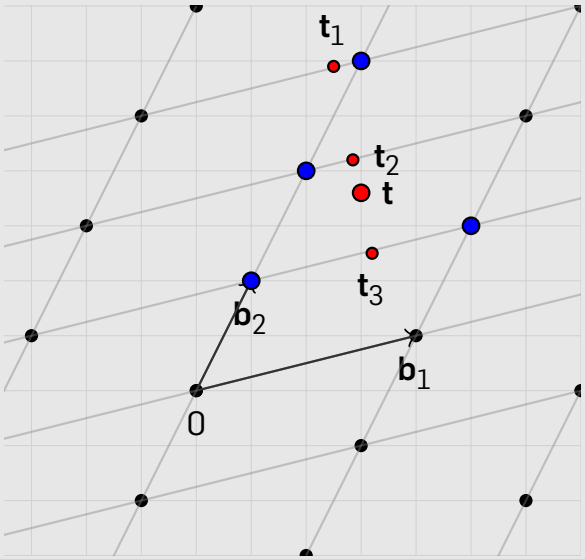
Closest point search via depth-first tree-traversal:

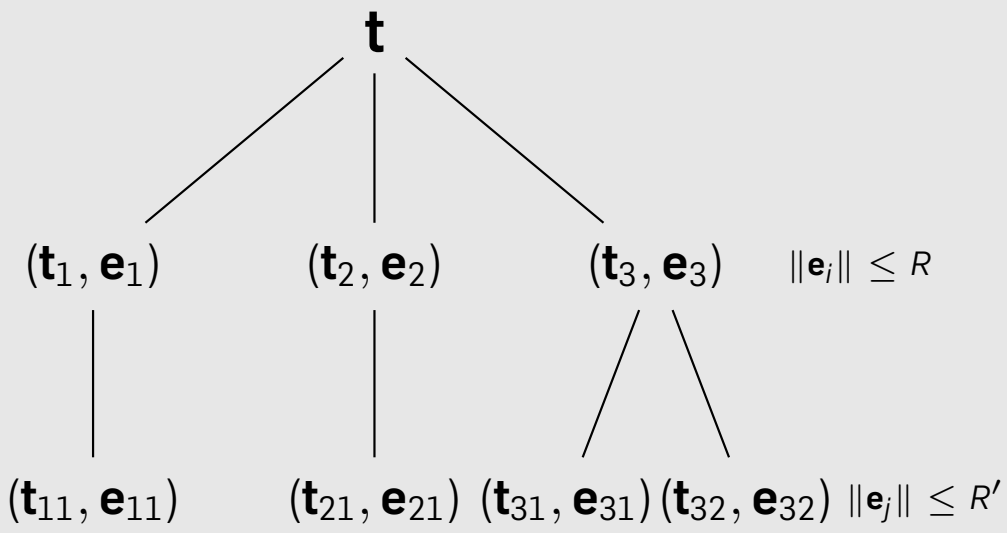




Enumeration

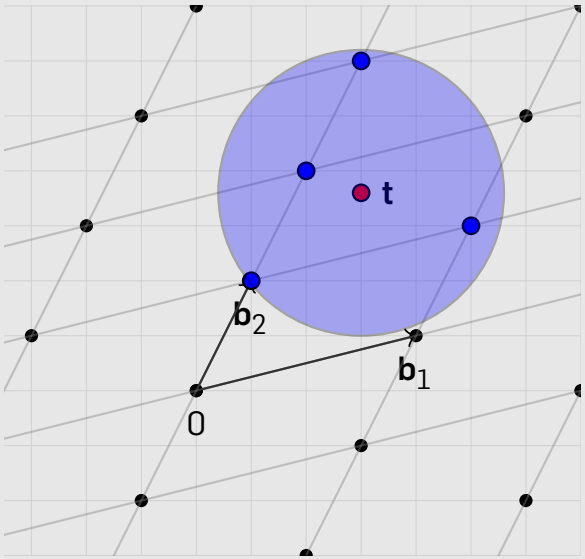
Closest point search via depth-first tree-traversal:

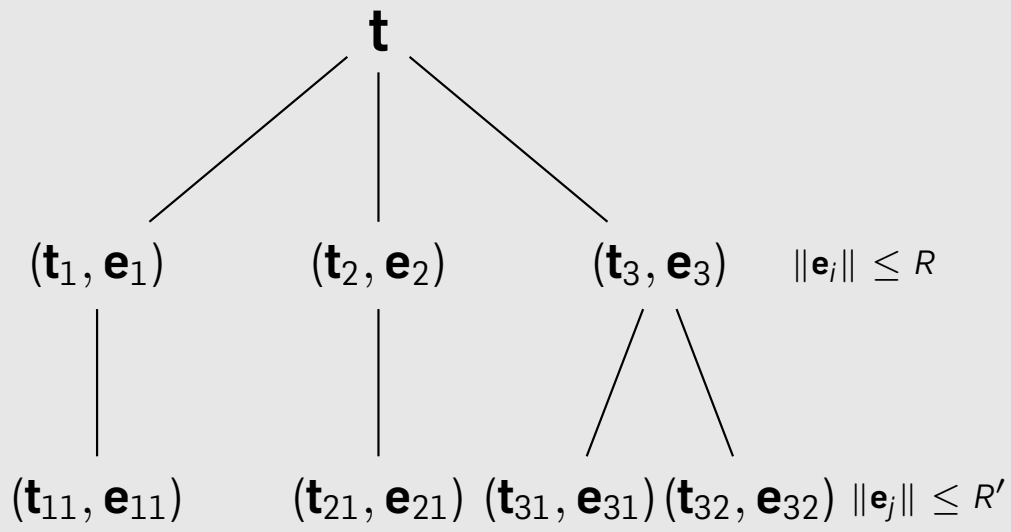




Enumeration

Closest point search via depth-first tree-traversal:





Leaves to visit $= 2^{n \log n}$ for n-dim BDD

Complexity

Running time of reduction (Step 1):

NTL's Implementation	Kannan's	Sieving (2^n - Memory)
2^{n^2}	$2^{n \log n}$	2^n

Running time of enumeration (Step 2): $2^{n \log n}$

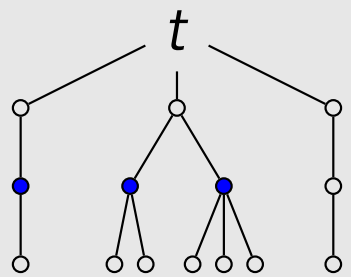
Idea: move the workload to Step 2 and use parallelization

Parallel Enumeration

Given N threads, find level with #Nodes $\geq N$:

1. Traverse the tree in *breadth-first* manner until the level is found
2. Spawn threads for N subtrees
3. Once a thread finishes a depth-first traversal, run the remaining subtree

Subtrees are unbalanced $\Longrightarrow$ choose a level with #Nodes $\gg N$

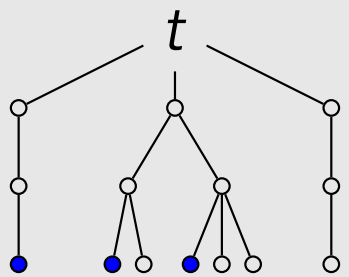


Parallel Enumeration

Given N threads, find level with #Nodes $\geq N$:

1. Traverse the tree in *breadth-first* manner until the level is found
2. Spawn threads for N subtrees
3. Once a thread finishes a depth-first traversal, run the remaining subtree

Subtrees are unbalanced $\Longrightarrow$ choose a level with #Nodes $\gg N$

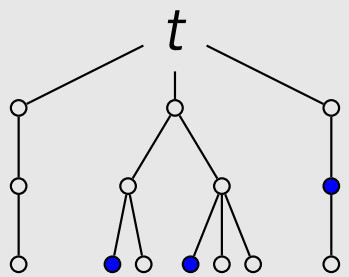


Parallel Enumeration

Given N threads, find level with #Nodes $\geq N$:

1. Traverse the tree in *breadth-first* manner until the level is found
2. Spawn threads for N subtrees
3. Once a thread finishes a depth-first traversal, run the remaining subtree

Subtrees are unbalanced $\Longrightarrow$ choose a level with #Nodes $\gg N$

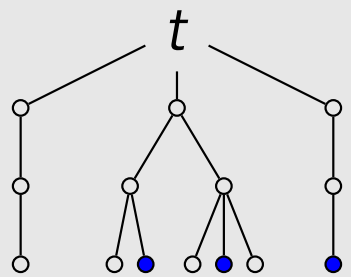


Parallel Enumeration

Given N threads, find level with #Nodes $\geq N$:

1. Traverse the tree in *breadth-first* manner until the level is found
2. Spawn threads for N subtrees
3. Once a thread finishes a depth-first traversal, run the remaining subtree

Subtrees are unbalanced $\Longrightarrow$ choose a level with #Nodes $\gg N$



Some results: standard LWE

Asymptotics: $\log(T_{\mathrm{LWE}}) = n\frac{\log n}{\log(q/\|\mathbf{e}\|)}$

LWE-parameters			BKZ-reduction	Enumeration	
n	q	$\lvert\mathbf{e}\rvert \leq$	T	# Threads	T
90	4093	10	11.3h	1	35h
90	4093	10	11.3h	10	3.6h
100	4093	10	7h	24	2.7h

LWE-samples $\approx 2n$.

To be compared with: $(n = 192, |\mathbf{e}| < 18, q = 4093)$ reaches 2^{87}-security level [1]

[1] Lindner, Peikert, 'Better Key Sizes (and Attacks) for LWE-Based Encryption', CT-RSA'11

Results: variants of LWE: small error

$$\text{Asymptotics: } \log(T_{\text{LWE}}) = n\frac{\log n}{\log(q/\|\mathbf{e}\|)}$$

$$\mathbf{e} \in \{0,1\}^m \text{ or } \mathbf{e} \in \{-1,0,1\}^m \ (m - \#\text{LWE samples})$$

LWE-parameters			BKZ-reduction	Enumeration	
n	q	m	T	# Threads	T
120	4093	170	1.5h	1	30min
130	4093	200	3.1h	1	1h
130	4093	200	3.1h	5	20min

Results: variants of LWE: small secret

Asymptotics: $\log(T_{\mathrm{LWE}}) = n\frac{\log n}{\log(q/\|\mathbf{e}\|)}$

$$(\mathbf{A}, \mathbf{t} = \mathbf{A}\cdot\mathbf{s} + \mathbf{e} \bmod q), \mathbf{s} \in \{0,1\}^n \iff$$
$$((\mathbf{e}, \mathbf{s}) - (\mathbf{t}, \mathbf{0}^n)) \begin{pmatrix} \mathbf{I}_m \\ \mathbf{A} \end{pmatrix} = 0 \mod q.$$

LWE-parameters			BKZ-reduction	Enumeration	
n	q	m	T	# Threads	T
140	16411	170	12h	1	16h
140	16411	170	12h	10	1.7h

To be compared with: ($n = 128, q = 16411, m = 2^{28}, T = 13\text{h}$) for combinatorial attack on LWE[2]

[2] Kirchner, Fouque, 'An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices'

Results: variants of LWE: binary matrix

$(\mathbf{A}, \mathbf{t} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \bmod q), \mathbf{A} \in \mathbb{Z}_2^{m \times n}$

For encryption: $\mathbf{u} \in \{0,1\}^m, (C_1, C_2) = (\mathbf{Au}, \langle \mathbf{u}, \mathbf{b} \rangle + m\lceil q/2 \rceil)$

Task: recover $\mathbf{u}$ given $(\mathbf{A}, \mathbf{Au})$.

$\Longrightarrow$ BDD instance over $\Lambda_q^{\perp}(\mathbf{A})$ - kernel of $\mathbf{A}$

parameters		LLL-reduction	Enumeration
n	m	T	T
256	440	4.5h	2min

Conclusions

- Shift the workload from BKZ to the Enumeration Step
- The enumeration is *efficiently parallelizable*
- The dimension reached by our experiments:
  - Standard LWE: $n = 100$
  - Binary error: $n = 140$
  - Binary secret: $n = 140$

Conclusions

- Shift the workload from BKZ to the Enumeration Step
- The enumeration is *efficiently parallelizable*
- The dimension reached by our experiments:
  - Standard LWE: $n = 100$
  - Binary error: $n = 140$
  - Binary secret: $n = 140$
- Better BKZ is needed for further improvements

Conclusions

- Shift the workload from BKZ to the Enumeration Step
- The enumeration is *efficiently parallelizable*
- The dimension reached by our experiments:
  - Standard LWE: $n = 100$
  - Binary error: $n = 140$
  - Binary secret: $n = 140$
- Better BKZ is needed for further improvements
- LWE challenge

Conclusions

- Shift the workload from BKZ to the Enumeration Step
- The enumeration is *efficiently parallelizable*
- The dimension reached by our experiments:
  - Standard LWE: $n = 100$
  - Binary error: $n = 140$
  - Binary secret: $n = 140$
- Better BKZ is needed for further improvements
- LWE challenge

Thank you! Q?