IBM Research

Lattice-based cryptography (I)

Thijs Laarhoven

mail@thijs.com
http://www.thijs.com/

PQCrypto Summer School 2017
(June 20, 2017)

IBM Research

Part 1: Lattices, cryptography, and lattice basis reduction

Thijs Laarhoven

mail@thijs.com
http://www.thijs.com/

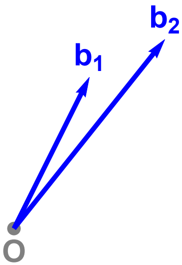
PQCrypto Summer School 2017
(June 20, 2017)

Lattices

What is a lattice?

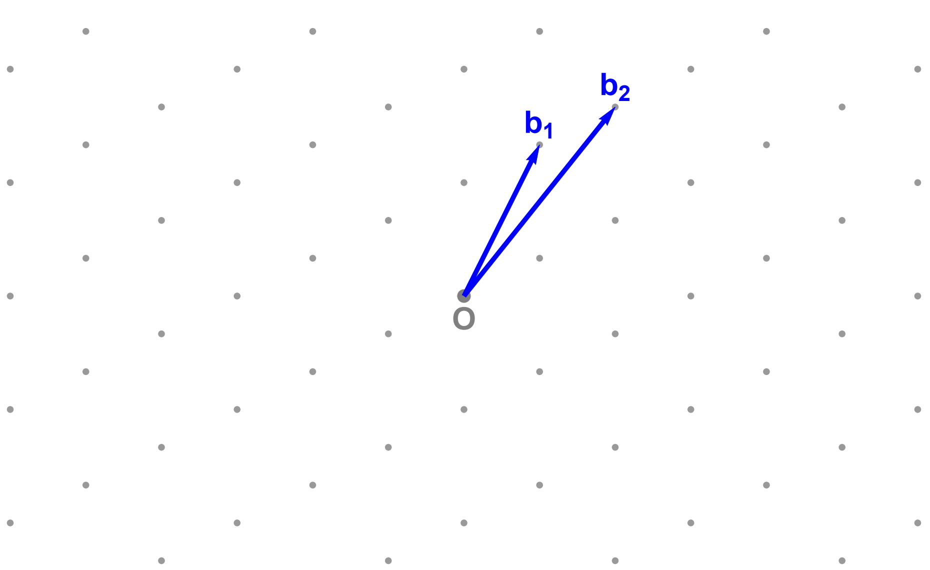
Lattices

What is a lattice?



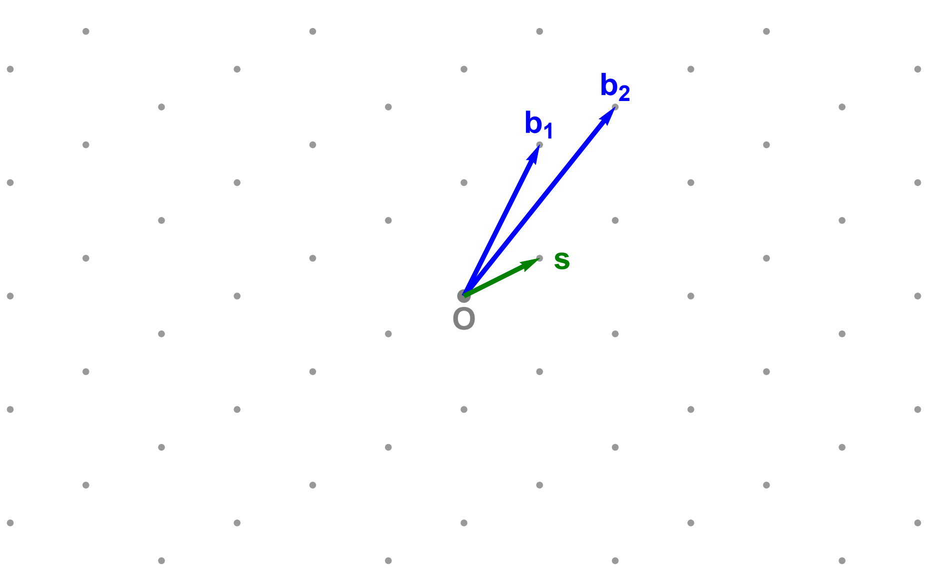
Lattices

What is a lattice?



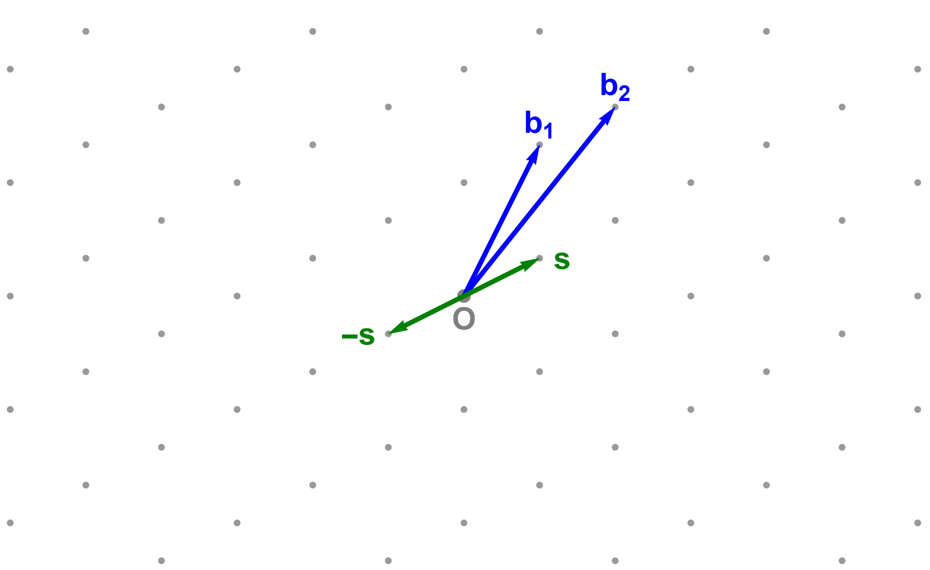
Lattices

Shortest Vector Problem (SVP)



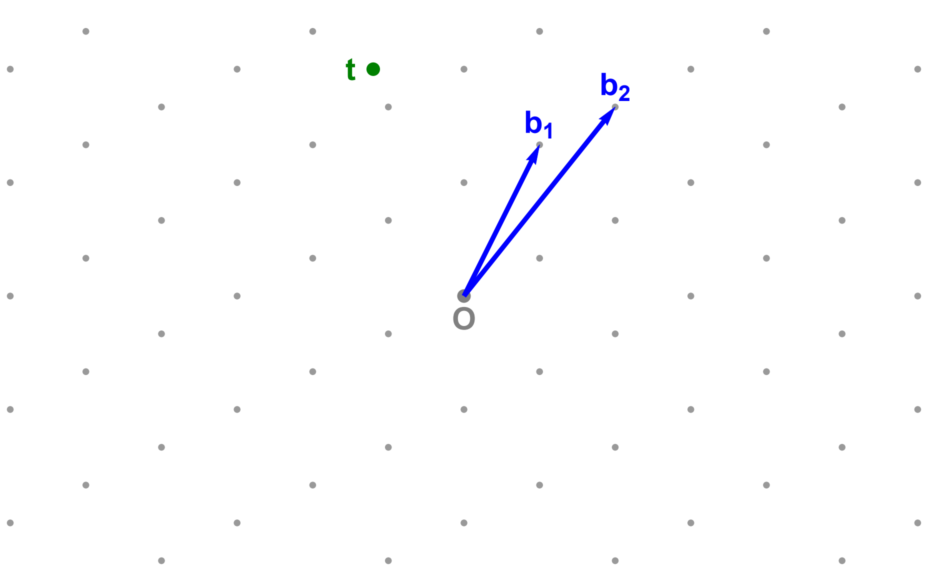
Lattices

Shortest Vector Problem (SVP)



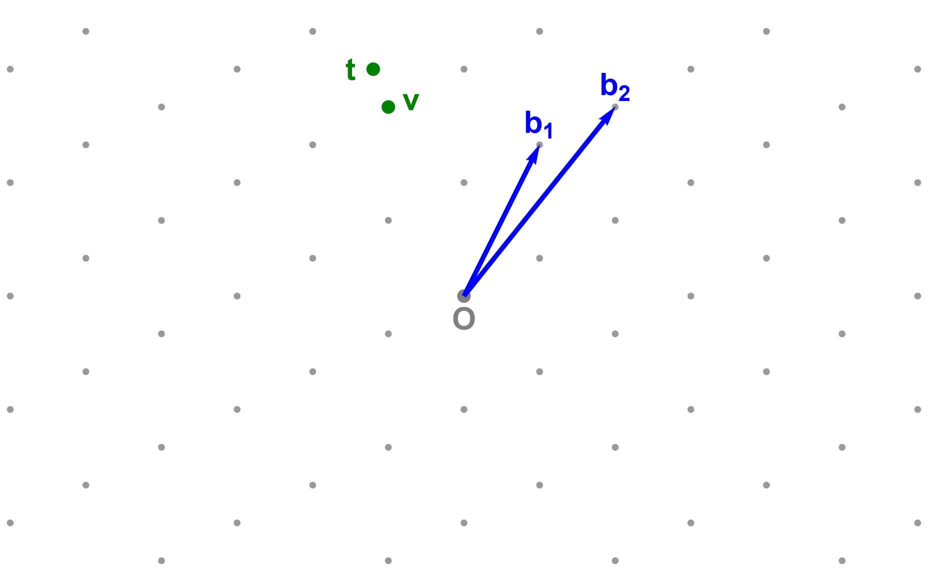
Lattices

Closest Vector Problem (CVP)



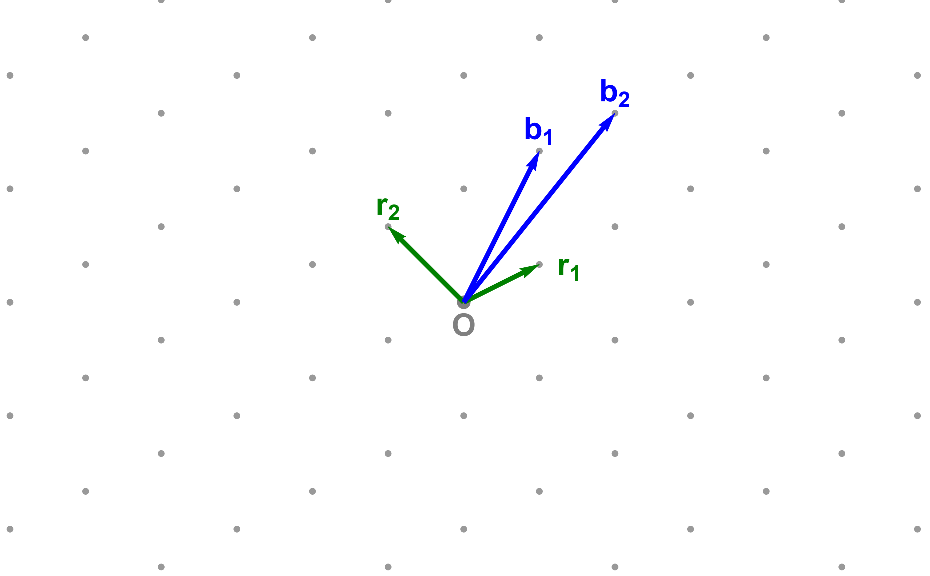
Lattices

Closest Vector Problem (CVP)



Lattices

Lattice basis reduction



Outline

Motivation: GGH encryption

Lattice basis reduction

- Gauss reduction
- LLL reduction
- BKZ reduction

Outline

Motivation: GGH encryption

Lattice basis reduction

- Gauss reduction
- LLL reduction
- BKZ reduction

GGH cryptosystem

Overview

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

$$\boldsymbol{v} = \boldsymbol{m}B$$
$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$

GGH cryptosystem

Private key

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

$$\boldsymbol{v} = \boldsymbol{m}B$$
$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$

GGH cryptosystem

Private key

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

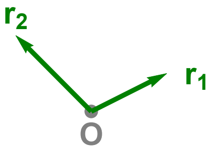
Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

$$\boldsymbol{v} = \boldsymbol{m}B$$
$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



GGH cryptosystem

Public key

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

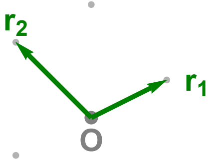
$$\boldsymbol{v} = \boldsymbol{m}B$$

$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$

$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



GGH cryptosystem

Public key

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

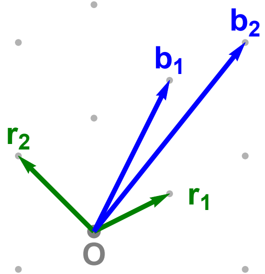
Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt m:

$$v = mB$$
$$c = v + e$$

Decrypt c:

$$v' = \lfloor cR^{-1} \rceil R$$
$$m' = v' B^{-1}$$



GGH cryptosystem

Encryption

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

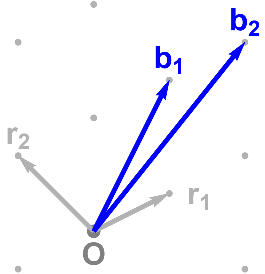
$$\boldsymbol{v} = \boldsymbol{m}B$$

$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$v' = \lfloor cR^{-1} \rceil R$$

$$m' = v'B^{-1}$$



GGH cryptosystem

Encryption

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

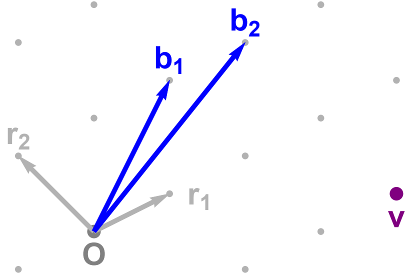
$$\boldsymbol{v} = \boldsymbol{m}B$$

$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$v' = \lfloor cR^{-1} \rceil R$$

$$m' = v' B^{-1}$$



GGH cryptosystem

Encryption

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

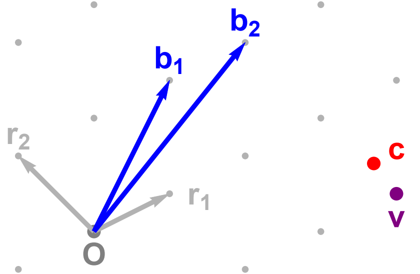
Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

$$\boldsymbol{v} = \boldsymbol{m}B$$
$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$v' = \lfloor cR^{-1} \rceil R$$
$$m' = v'B^{-1}$$



GGH cryptosystem

Decryption with good basis

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

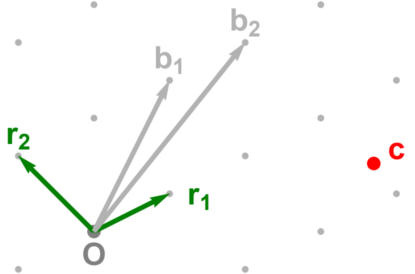
Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt m:

$$v = mB$$
$$c = v + e$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



GGH cryptosystem

Decryption with good basis

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

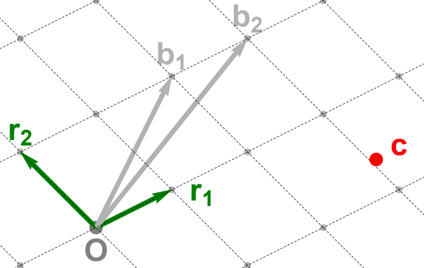
Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt m:

$$v = mB$$
$$c = v + e$$

Decrypt c:

$$v' = \lfloor cR^{-1} \rceil R$$
$$m' = v' B^{-1}$$



GGH cryptosystem

Decryption with good basis

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

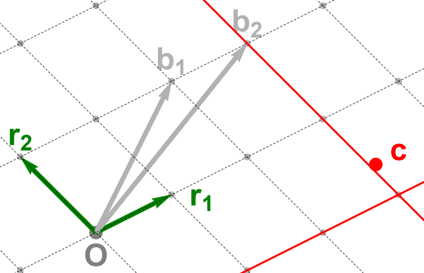
Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

$$\boldsymbol{v} = \boldsymbol{m}B$$
$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



GGH cryptosystem

Decryption with good basis

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

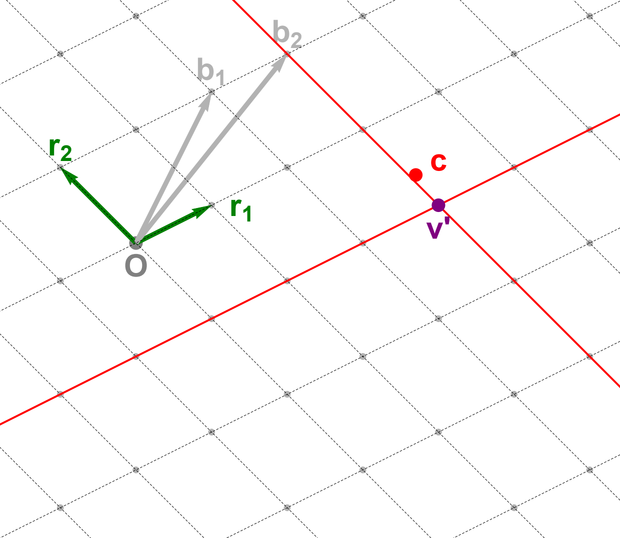
$$\boldsymbol{v} = \boldsymbol{m}B$$

$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$

$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



GGH cryptosystem

Decryption with bad basis

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

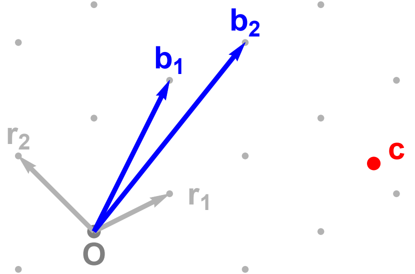
Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt m:

$$v = mB$$
$$c = v + e$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



GGH cryptosystem

Decryption with bad basis

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

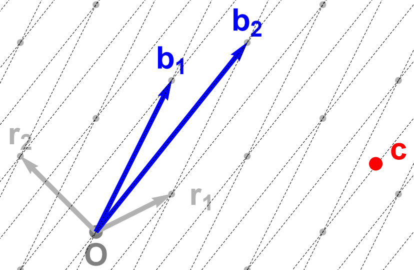
Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt m:

$$v = mB$$
$$c = v + e$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$
$$\boldsymbol{m}' = \boldsymbol{v}' B^{-1}$$



GGH cryptosystem

Decryption with bad basis

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

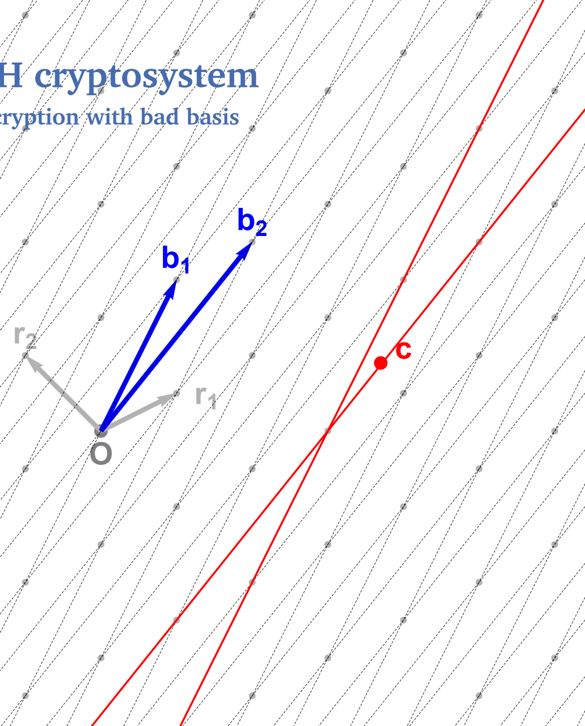
Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt m:

$$v = mB$$
$$c = v + e$$

Decrypt c:

$$v' = \lfloor cR^{-1} \rceil R$$
$$m' = v' B^{-1}$$



GGH cryptosystem

Decryption with bad basis

Private key: $R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Encrypt m:

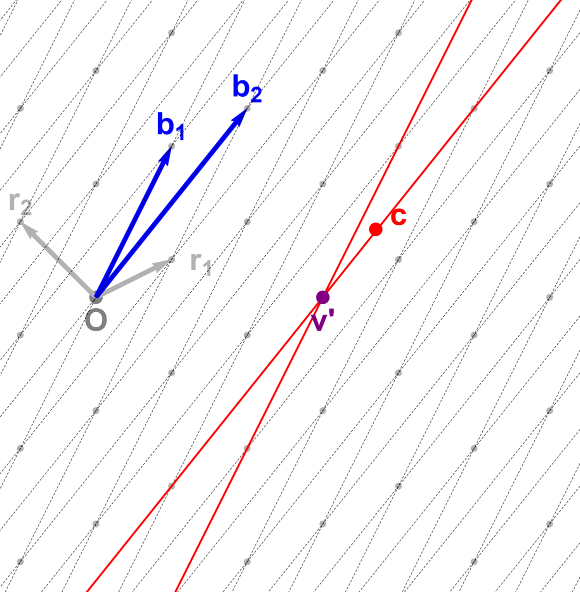
$v = mB$

$c = v + e$

Decrypt $\mathbf{c}$:

$v' = \lfloor cR^{-1} \rceil R$

$m' = v' B^{-1}$



GGH cryptosystem

Overview

Private key: $R = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix}$

Public key: $B = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}$

Encrypt $\boldsymbol{m}$:

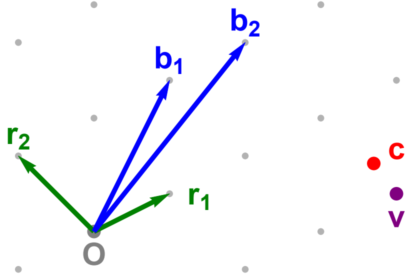
$$\boldsymbol{v} = \boldsymbol{m}B$$

$$\boldsymbol{c} = \boldsymbol{v} + \boldsymbol{e}$$

Decrypt $\boldsymbol{c}$:

$$\boldsymbol{v}' = \lfloor \boldsymbol{c}R^{-1} \rceil R$$

$$\boldsymbol{m}' = \boldsymbol{v}'B^{-1}$$



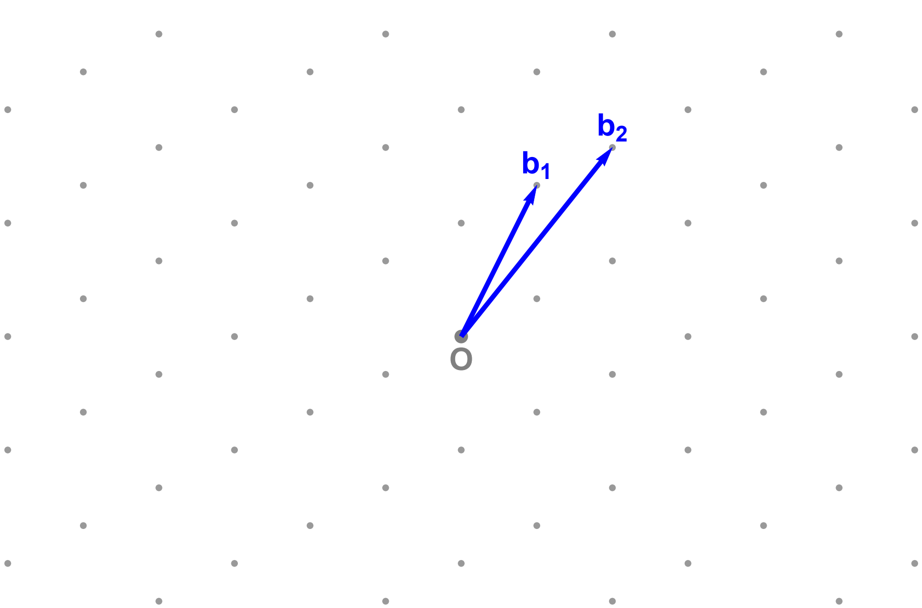
Outline

Motivation: GGH encryption

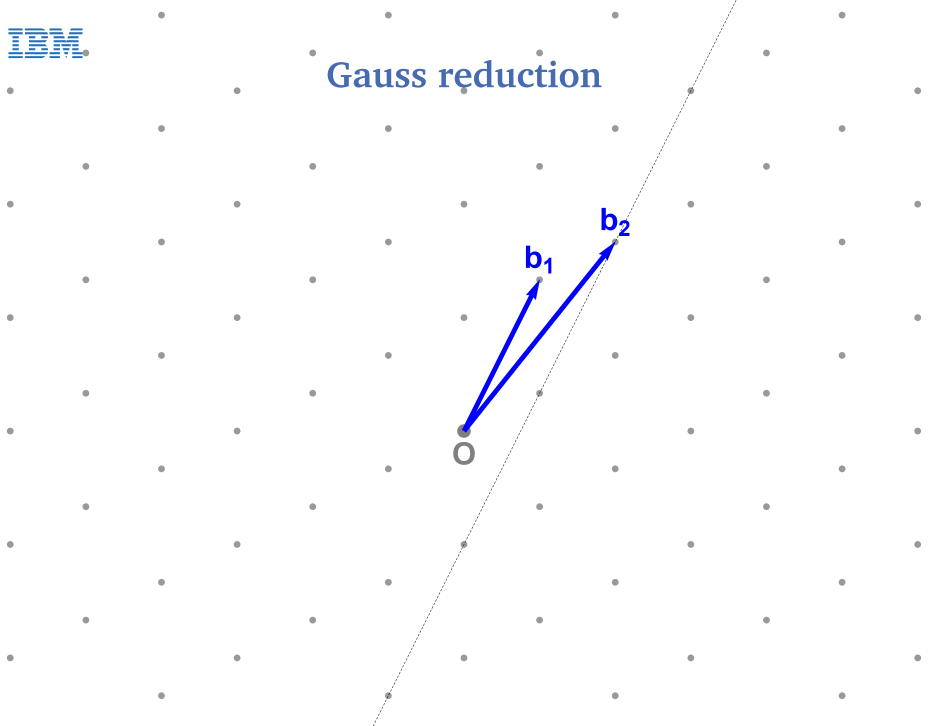
Lattice basis reduction

- Gauss reduction
- LLL reduction
- BKZ reduction

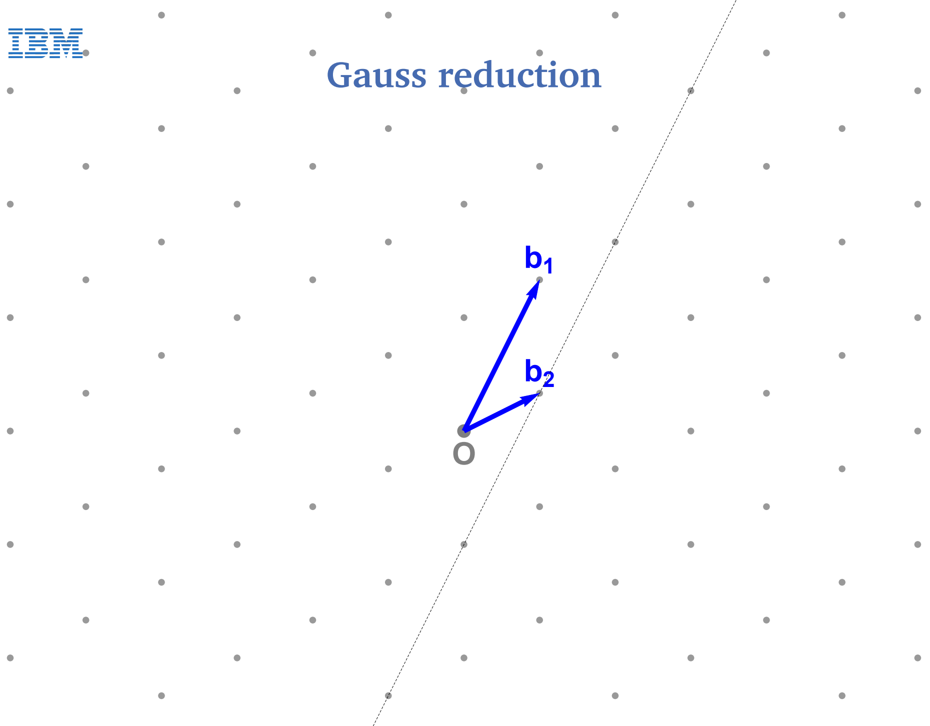
Gauss reduction



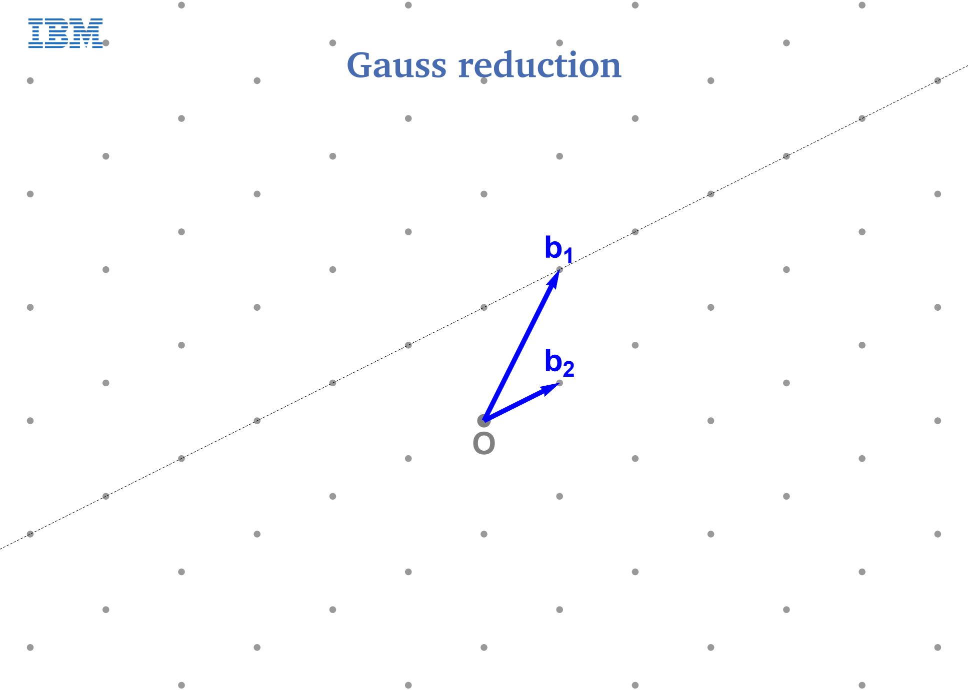
Gauss reduction



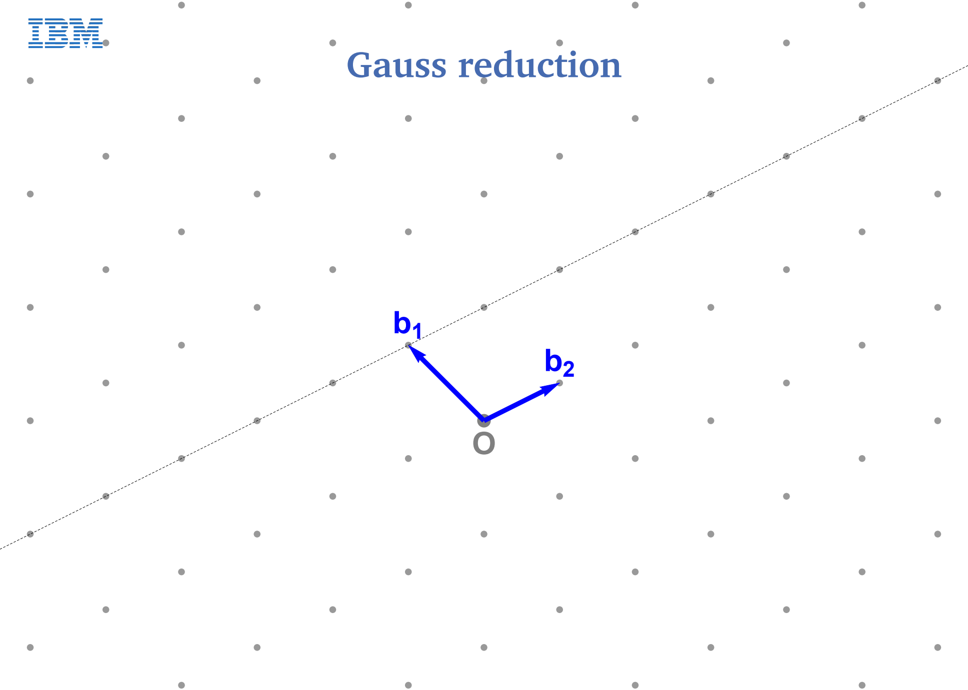
Gauss reduction



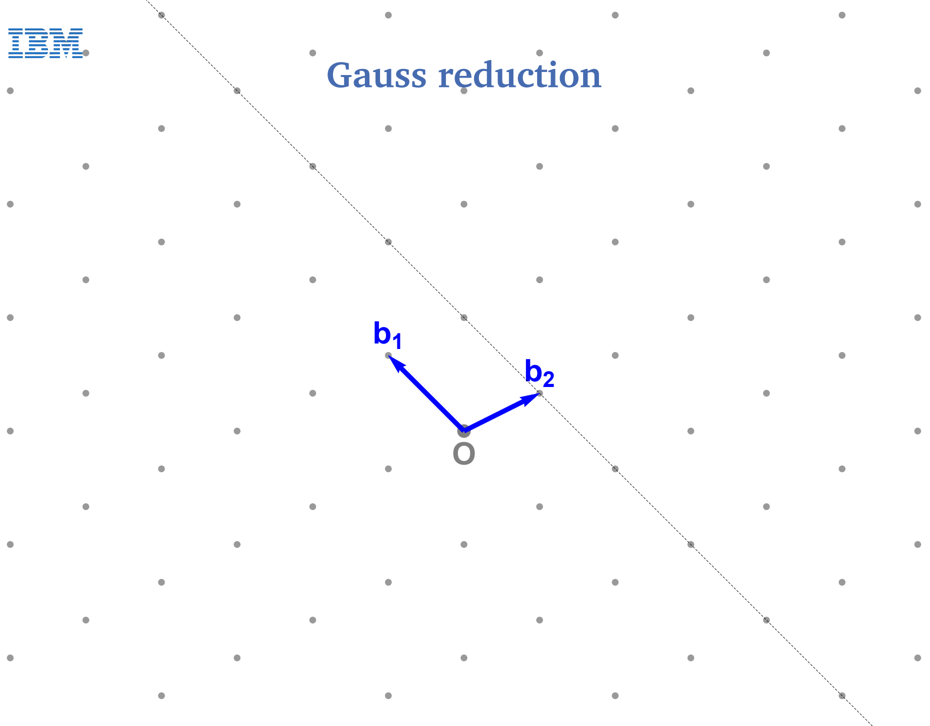
Gauss reduction



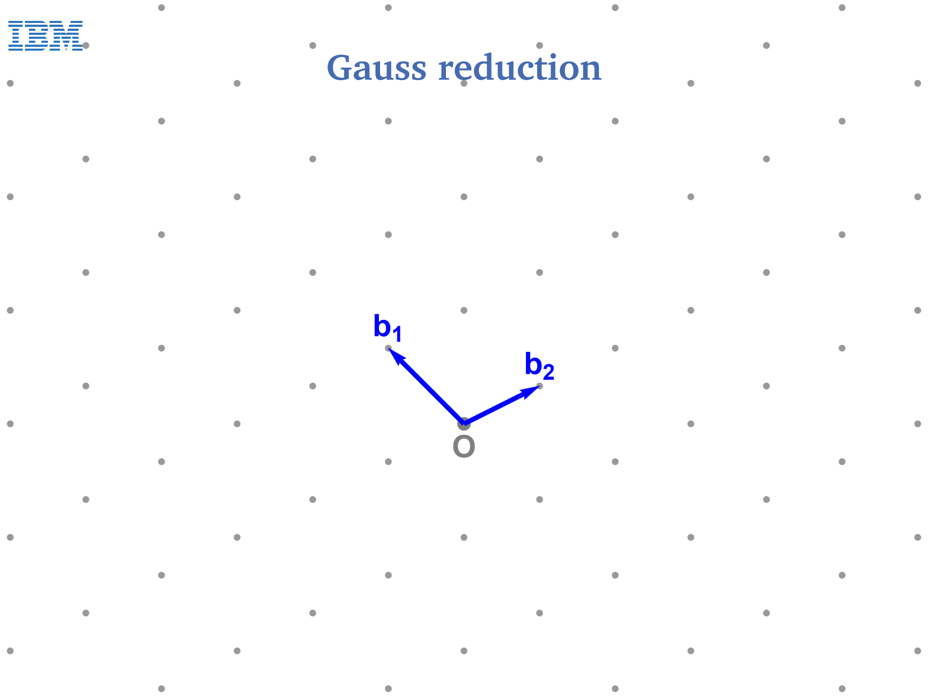
Gauss reduction



Gauss reduction



Gauss reduction



Gauss reduction

Given $\mathbf{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2\}$, repeat two steps:

- **Swap**: If $\|\boldsymbol{b}_1\| > \|\boldsymbol{b}_2\|$, then swap $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$.
- **Reduce**: While $\|\boldsymbol{b}_2 \pm \boldsymbol{b}_1\| < \|\boldsymbol{b}_2\|$, replace $\boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 \pm \boldsymbol{b}_1$.

Gauss reduction

Given $\mathbf{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2\}$, repeat two steps:

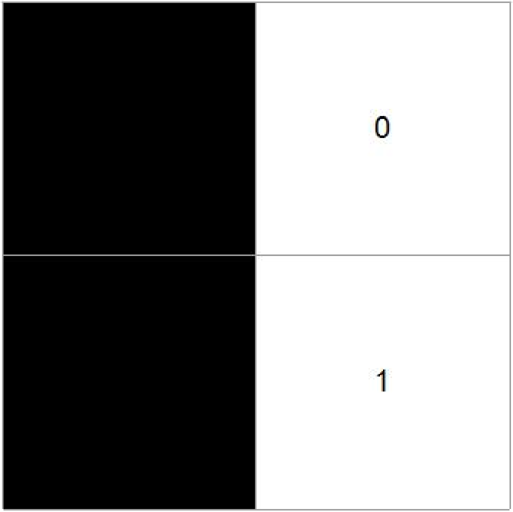
- **Swap**: If $\|\boldsymbol{b}_1\| > \|\boldsymbol{b}_2\|$, then swap $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$.
- **Reduce**: While $\|\boldsymbol{b}_2 \pm \boldsymbol{b}_1\| < \|\boldsymbol{b}_2\|$, replace $\boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 \pm \boldsymbol{b}_1$.

At the end, $\boldsymbol{b}_1$ is a shortest (non-zero) lattice vector and $\boldsymbol{b}_2$ a "second shortest" (non-zero) lattice vector.

Gauss reduction

3141592653	0
5897932384	1

Gauss reduction



LLL algorithm

Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$

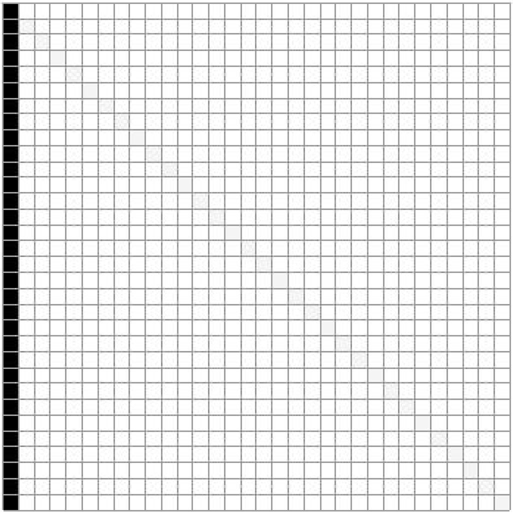
LLL algorithm

	0	0	0	0	0	0	0
	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	1	0	0
	0	0	0	0	0	1	0
	0	0	0	0	0	0	1

LLL algorithm

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

LLL algorithm



BKZ algorithm

Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$

BKZ algorithm

Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$
- Basis quality deteriorates with the dimension n
  - Theoretically: $\|\boldsymbol{b}_1\| \leq 1.075^n \cdot \det(\mathscr{L})$
  - Experimentally: $\|\boldsymbol{b}_1\| \approx 1.022^n \cdot \det(\mathscr{L})$

BKZ algorithm

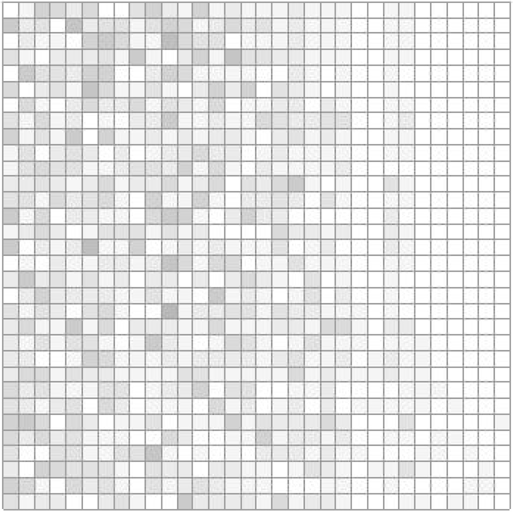
Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$
- Basis quality deteriorates with the dimension n
  - Theoretically: $\|\boldsymbol{b}_1\| \leq 1.075^n \cdot \det(\mathscr{L})$
  - Experimentally: $\|\boldsymbol{b}_1\| \approx 1.022^n \cdot \det(\mathscr{L})$

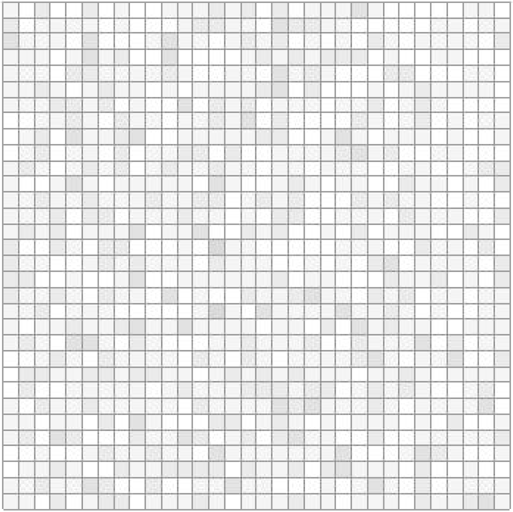
Blockwise Korkine-Zolotarev (BKZ) reduction [Sch87, SE94]

- Blockwise generalization of Korkine-Zolotarev reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \ldots, \boldsymbol{b}_{i+k-1})$ for $i = 1, \ldots, n-k+1$
- Blocksize k offers time-quality tradeoff

LLL algorithm



BKZ algorithm



BKZ algorithm

Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$
- Basis quality deteriorates with the dimension n
  - Theoretically: $\|\boldsymbol{b}_1\| \leq 1.075^n \cdot \det(\mathcal{L})$
  - Experimentally: $\|\boldsymbol{b}_1\| \approx 1.022^n \cdot \det(\mathcal{L})$

Blockwise Korkine-Zolotarev (BKZ) reduction [Sch87, SE94]

- Blockwise generalization of Korkine-Zolotarev reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \ldots, \boldsymbol{b}_{i+k-1})$ for $i = 1, \ldots, n-k+1$
- Blocksize k offers time-quality tradeoff

BKZ algorithm

Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$
- Basis quality deteriorates with the dimension n
  - Theoretically: $\|\boldsymbol{b}_1\| \leq 1.075^n \cdot \det(\mathscr{L})$
  - Experimentally: $\|\boldsymbol{b}_1\| \approx 1.022^n \cdot \det(\mathscr{L})$

Blockwise Korkine-Zolotarev (BKZ) reduction [Sch87, SE94]

- Blockwise generalization of Korkine-Zolotarev reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \ldots, \boldsymbol{b}_{i+k-1})$ for $i = 1, \ldots, n-k+1$
- Blocksize k offers time-quality tradeoff

BKZ uses **exact SVP** algorithm in dimension k as subroutine

BKZ algorithm

Lenstra-Lenstra-Lovasz (LLL) algorithm [LLL82]

- Blockwise generalization of Gauss reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \boldsymbol{b}_{i+1})$ for $i = 1, \ldots, n-1$
- Basis quality deteriorates with the dimension n
  - Theoretically: $\|\boldsymbol{b}_1\| \leq 1.075^n \cdot \det(\mathscr{L})$
  - Experimentally: $\|\boldsymbol{b}_1\| \approx 1.022^n \cdot \det(\mathscr{L})$

Blockwise Korkine-Zolotarev (BKZ) reduction [Sch87, SE94]

- Blockwise generalization of Korkine-Zolotarev reduction
- Do reductions/swaps on $(\boldsymbol{b}_i, \ldots, \boldsymbol{b}_{i+k-1})$ for $i = 1, \ldots, n-k+1$
- Blocksize k offers time-quality tradeoff

BKZ uses **exact SVP** algorithm in dimension k as subroutine
Next hour: How to solve exact SVP in high dimensions?