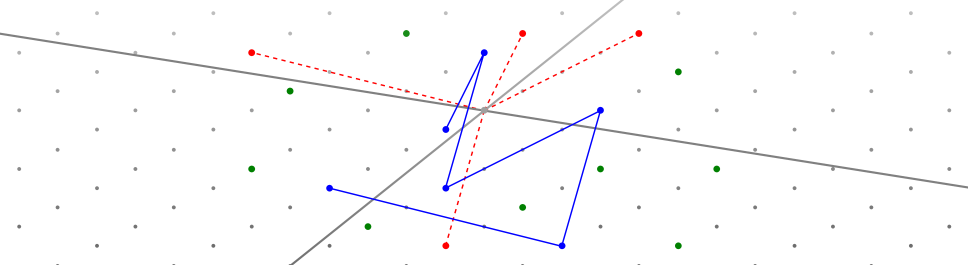
# Search problems in cryptography

## From fingerprinting to lattice sieving

Thijs Laarhoven



# Part I: Fingerprinting

# Part I: Fingerprinting

- Problem: preventing digital piracy

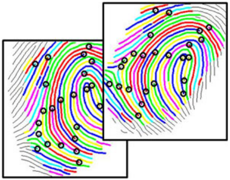
# Part I: Fingerprinting


- Problem: preventing digital piracy
- Construct unique, digital fingerprints

# Part I: Fingerprinting

- Problem: preventing digital piracy
- Construct unique, digital fingerprints
- Embed different fingerprints in each copy

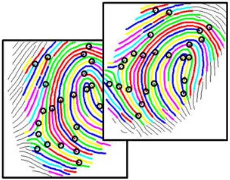
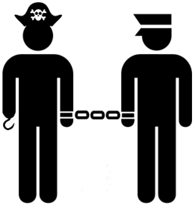
# Part I: Fingerprinting



- Problem: preventing digital piracy
- Construct unique, digital fingerprints
- Embed different fingerprints in each copy
- Content owners can compare pirate copies with database

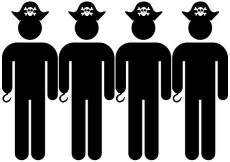
# Part I: Fingerprinting



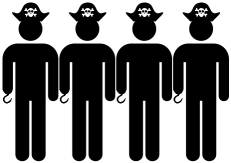
- Problem: preventing digital piracy
- Construct unique, digital fingerprints
- Embed different fingerprints in each copy
- Content owners can compare pirate copies with database
- Allows content owners to catch pirates

# Part I: Fingerprinting



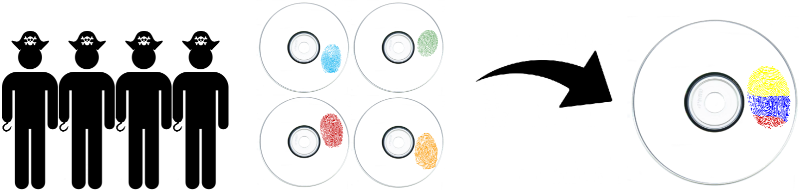
- Problem: collusion attacks

# Part I: Fingerprinting



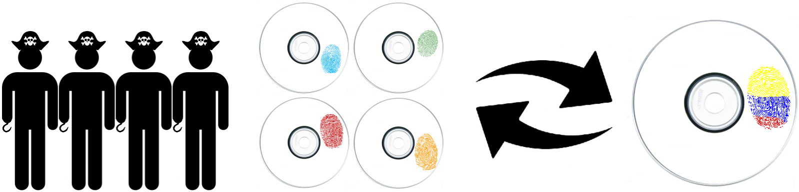
- Problem: collusion attacks
- Different copies have different fingerprints

# Part I: Fingerprinting



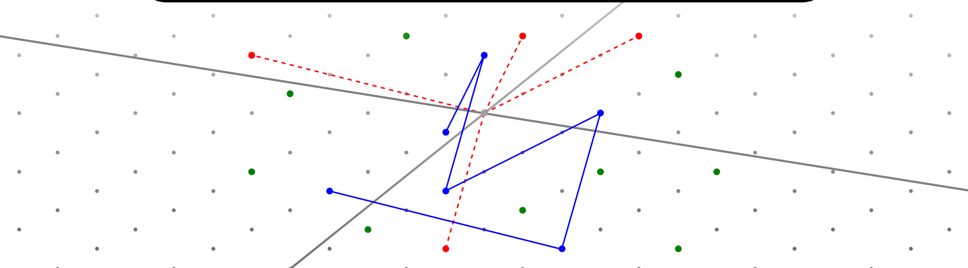
- Problem: collusion attacks
- Different copies have different fingerprints
- Mix several copies to distort the fingerprint

# Part I: Fingerprinting



- Problem: collusion attacks
- Different copies have different fingerprints
- Mix several copies to distort the fingerprint
- Part I: collusion-resistant fingerprinting schemes

# Part II: Lattice sieving

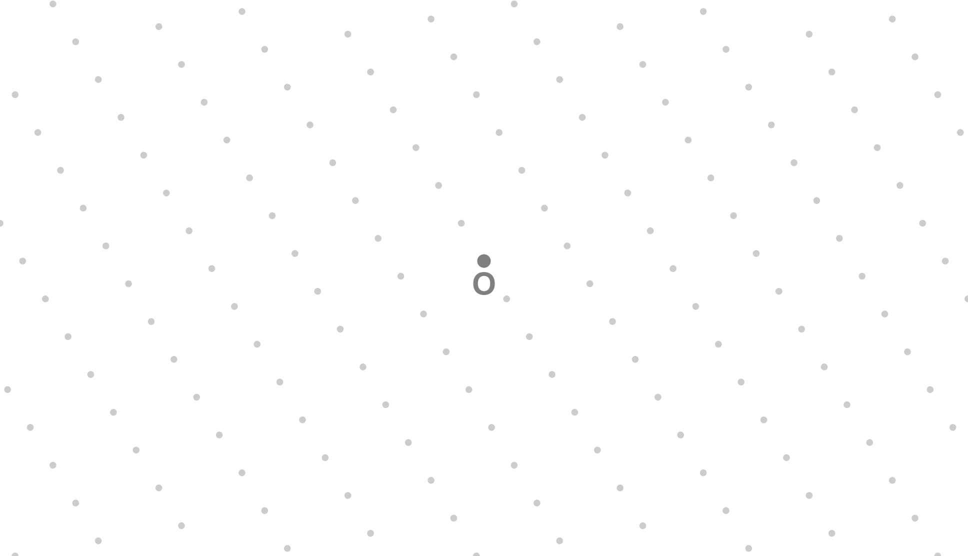


# Part II: Lattice sieving

- Lattice: structured grid of points

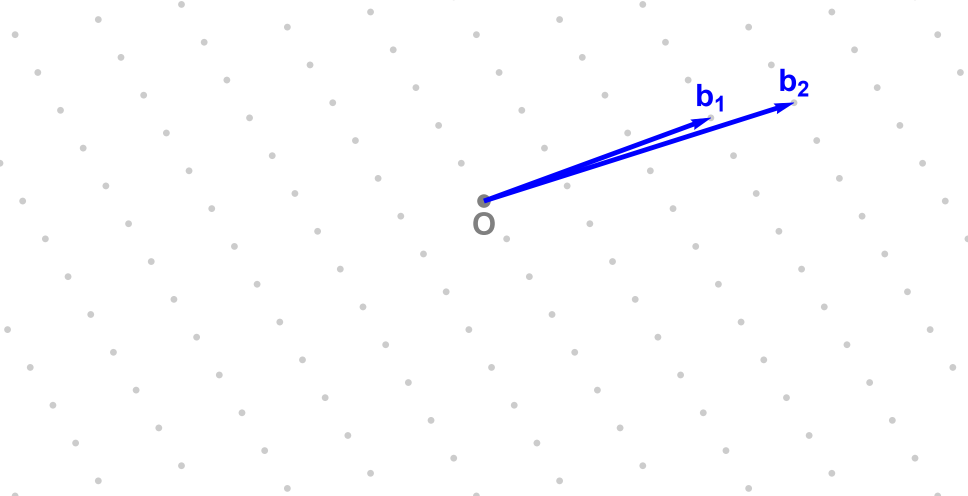
# Part II: Lattice sieving

- Lattice: structured grid of points



# Part II: Lattice sieving

- Lattice: structured grid of points
- Mathematically: discrete subgroup of $\mathbb{R}^d$



# Part II: Lattice sieving

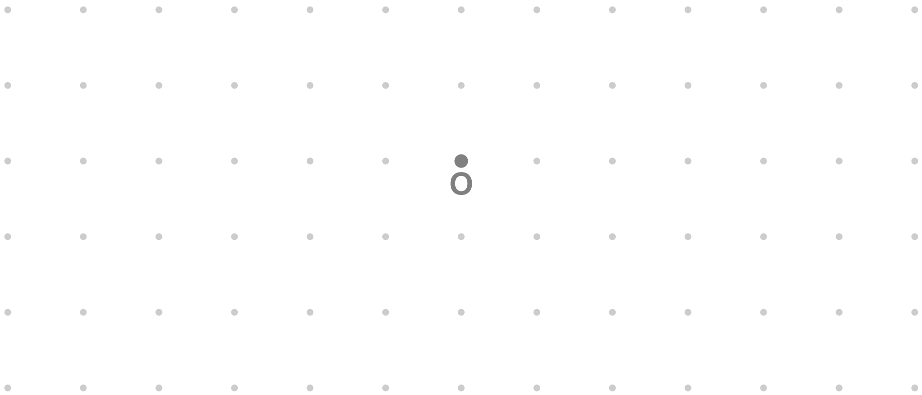
- Lattice: structured grid of points
- Mathematically: discrete subgroup of $\mathbb{R}^d$
- Lattices come in various shapes and sizes

# Part II: Lattice sieving

- Lattice: structured grid of points
- Mathematically: discrete subgroup of $\mathbb{R}^d$
- Lattices come in various shapes and sizes

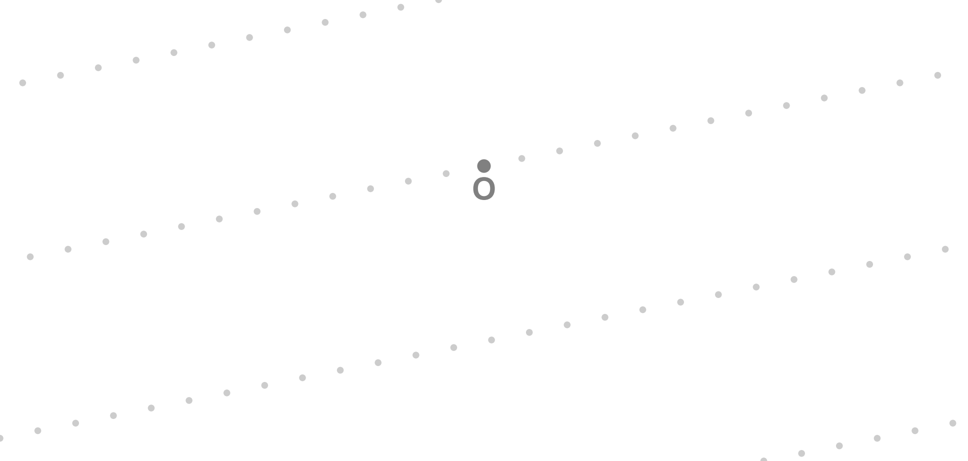
# Part II: Lattice sieving

- Lattice: structured grid of points
- Mathematically: discrete subgroup of $\mathbb{R}^d$
- Lattices come in various shapes and sizes



# Part II: Lattice sieving

- Lattice: structured grid of points
- Mathematically: discrete subgroup of $\mathbb{R}^d$
- Lattices come in various shapes and sizes

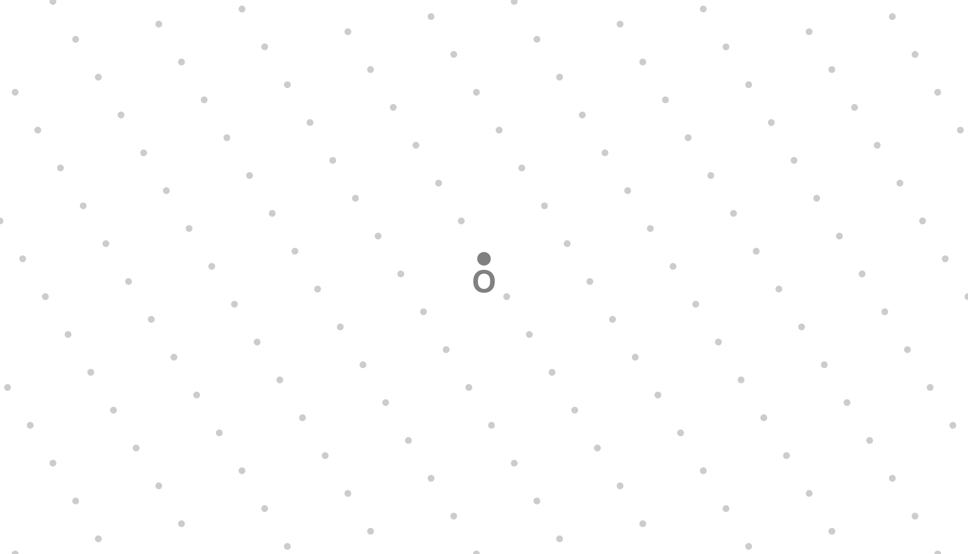


# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors

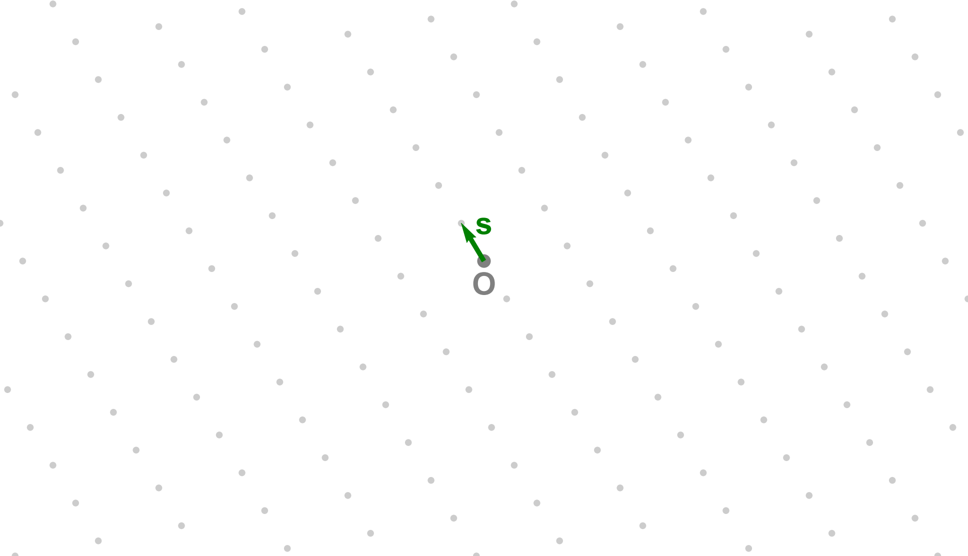
# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors



# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors

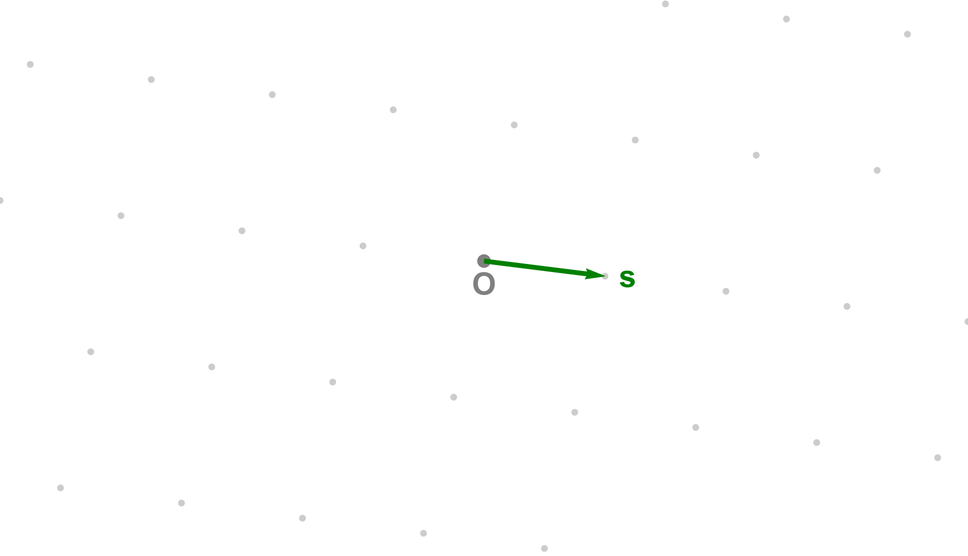


# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors

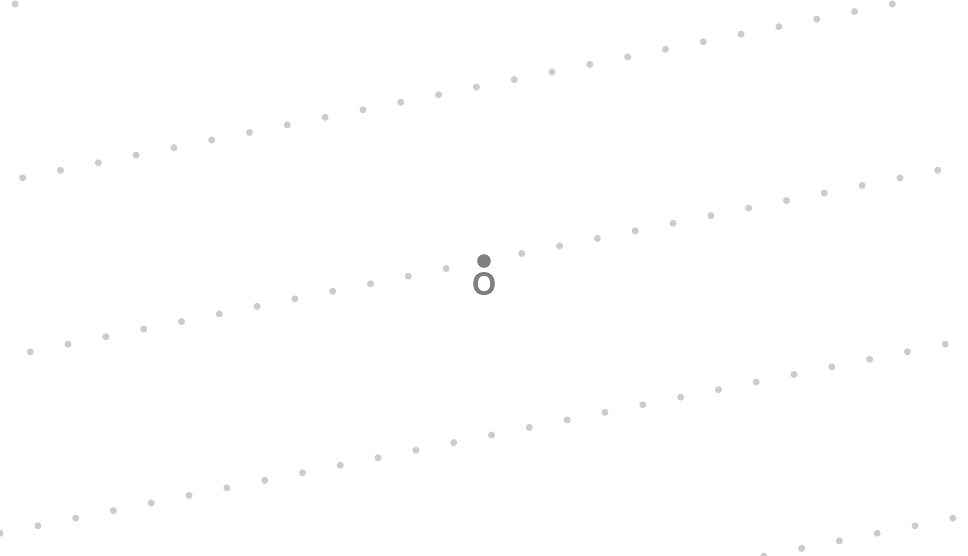
# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors



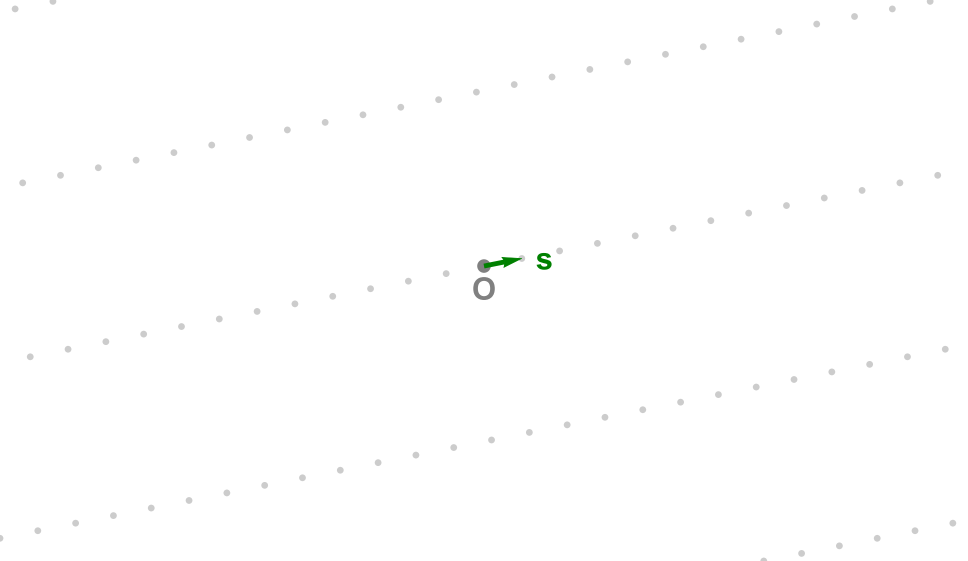
# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors



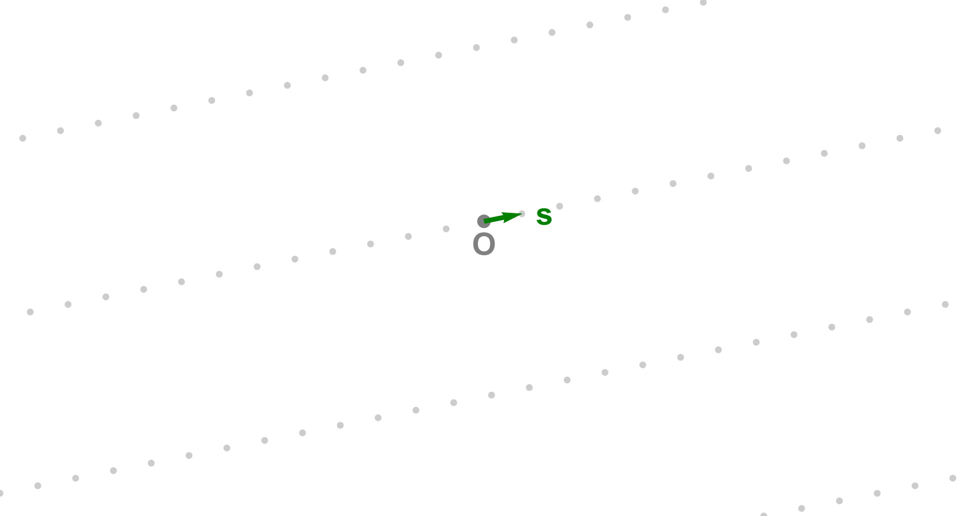
# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors



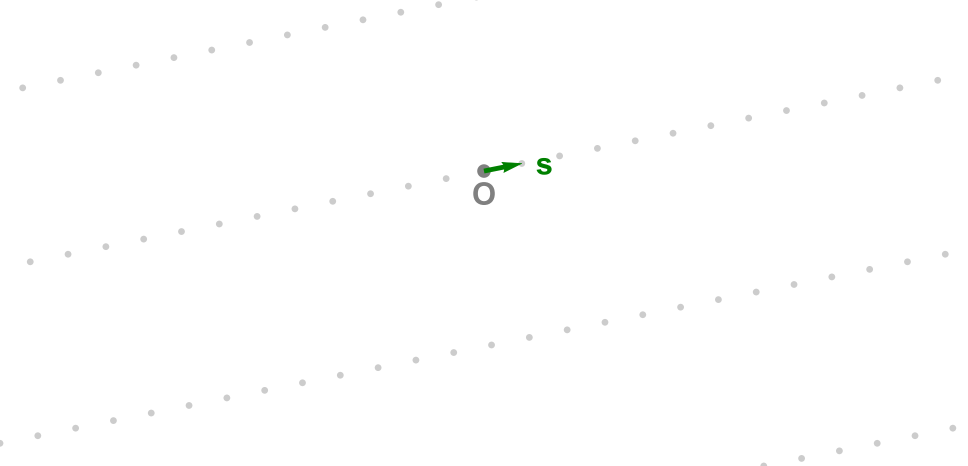
# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors
- Easy in low dimensions, hard in high dimensions



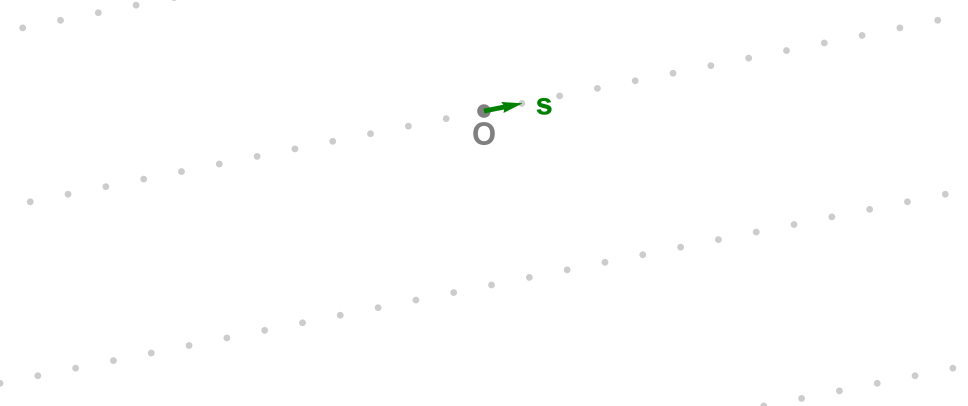
# Part II: Lattice sieving

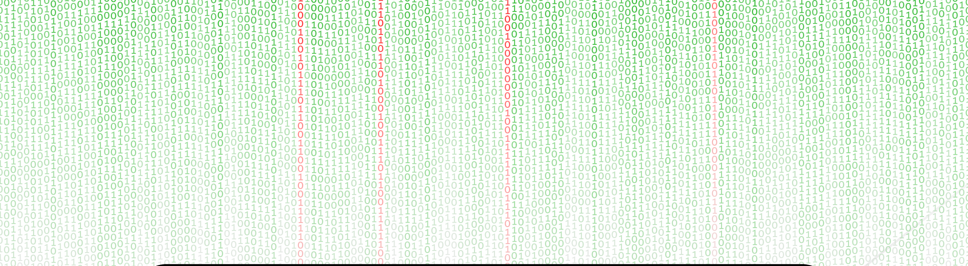
- Problem: finding shortest (non-zero) lattice vectors
- Easy in low dimensions, hard in high dimensions
- Lattice sieving fastest method in high dimensions



# Part II: Lattice sieving

- Problem: finding shortest (non-zero) lattice vectors
- Easy in low dimensions, hard in high dimensions
- Lattice sieving fastest method in high dimensions
- Part II: faster lattice sieving methods

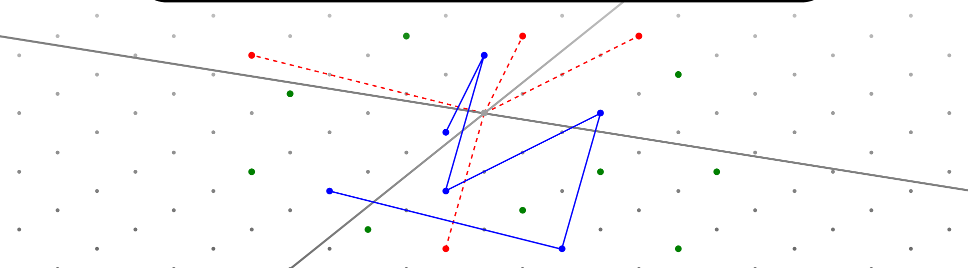




# Search problems in cryptography

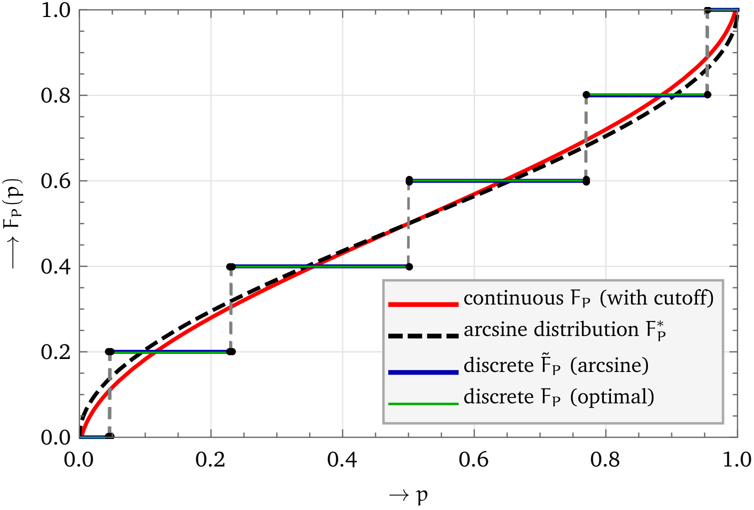
From fingerprinting to lattice sieving

Thijs Laarhoven



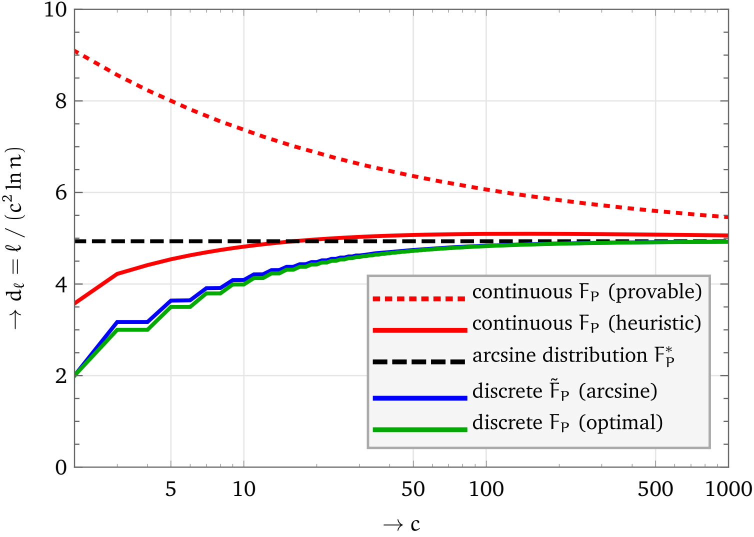
# Figures and tables

**Figure 2.1**



# Figures and tables

**Figure 2.2**



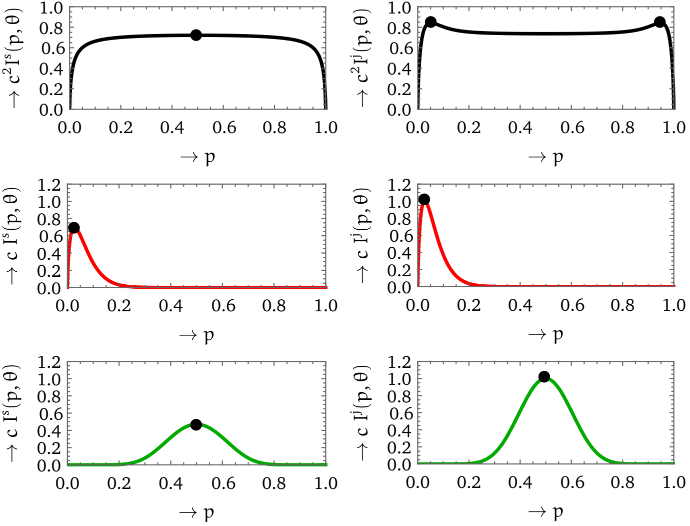
# Figures and tables

**Table 3.1**

| Pirate strategy | Simple capacity | Joint capacity |
| --- | --- | --- |
| $\boldsymbol{\theta}_{\text{int}}$: interleaving attack | $\left(\frac{1}{2\ln 2}\right)/c^2 \approx 0.72/c^2$ | $\beta/c^2 \approx 0.84/c^2$ |
| $\boldsymbol{\theta}_{\text{all1}}$: all-1 attack | $(\ln 2)/c \approx 0.69/c$ | $1/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{maj}}$: majority voting | $\left(\frac{1}{\pi\ln 2}\right)/c \approx 0.46/c$ | $1/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{min}}$: minority voting | $(\ln 2)/c \approx 0.69/c$ | $1/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{coin}}$: coin-flip attack | $\left(\frac{1}{4}\ln 2\right)/c \approx 0.17/c$ | $\left(\log_2 \frac{5}{4}\right)/c \approx 0.32/c$ |

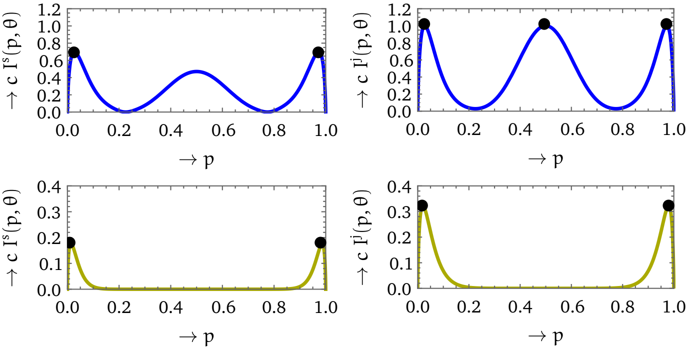
# Figures and tables

## Figure 3.1 (a-f)



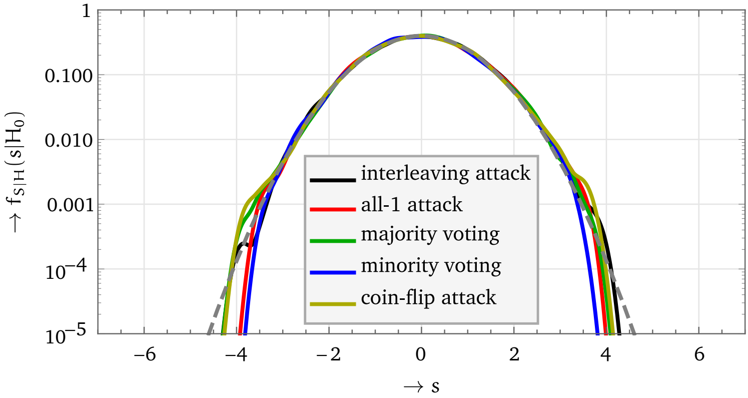
# Figures and tables

## Figure 3.1 (g-j)



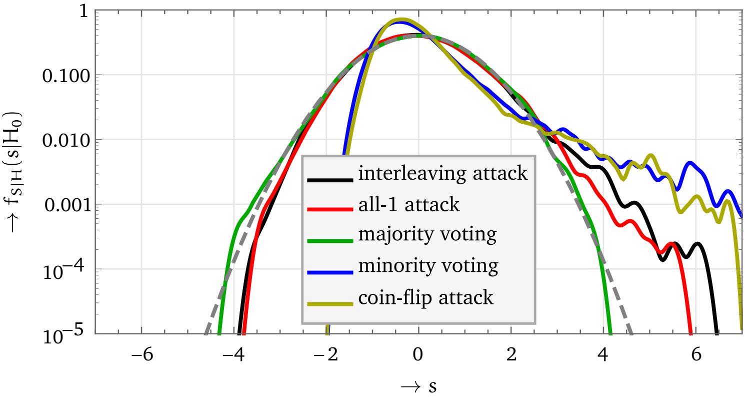
# Figures and tables

## Figure 4.1 (a)



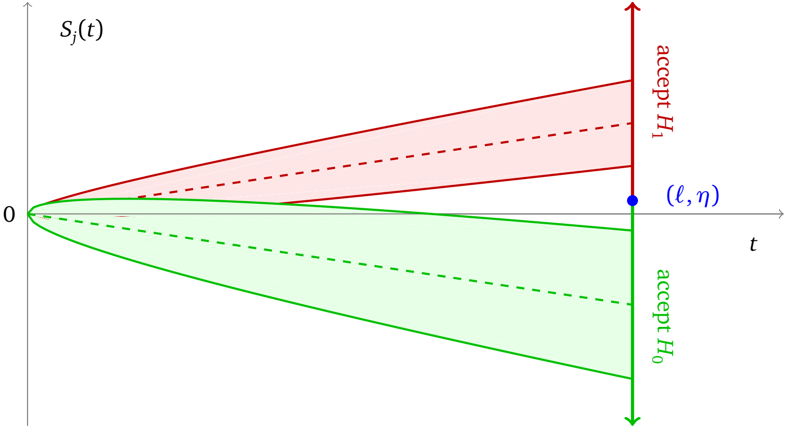
# Figures and tables

## Figure 4.1 (b)



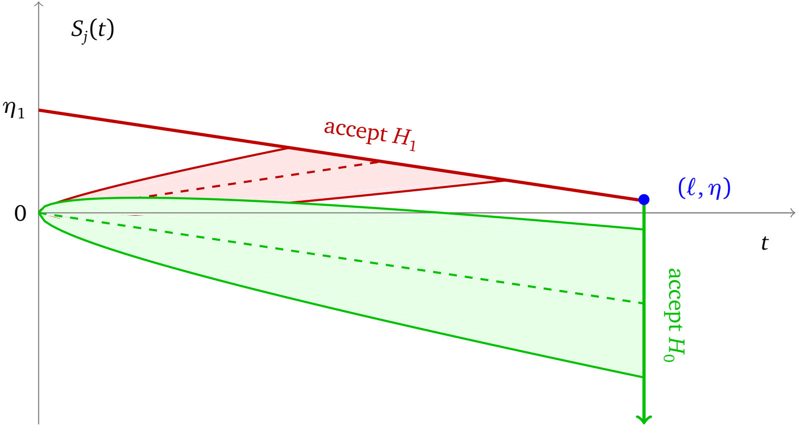
# Figures and tables

**Figure 5.1 (a)**



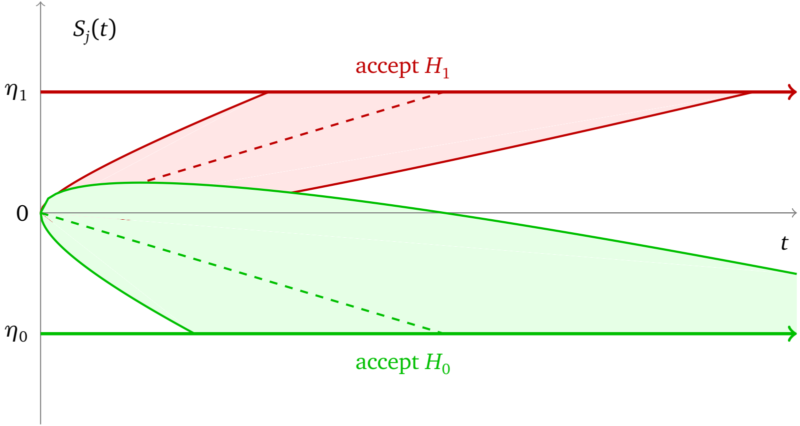
# Figures and tables

**Figure 5.1 (b)**



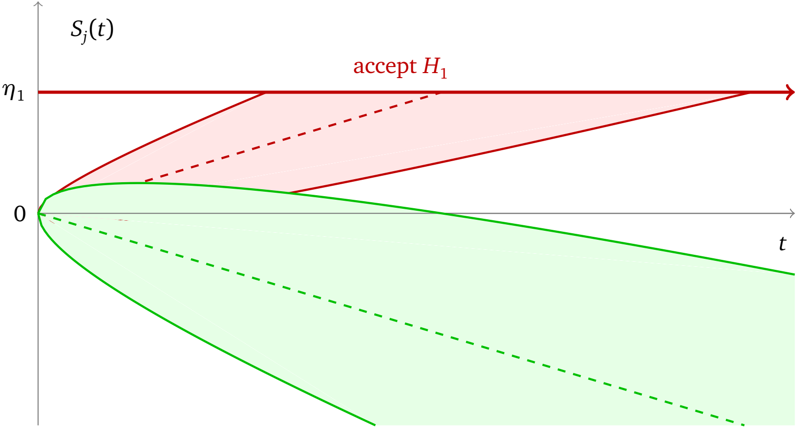
# Figures and tables

**Figure 5.1 (c)**



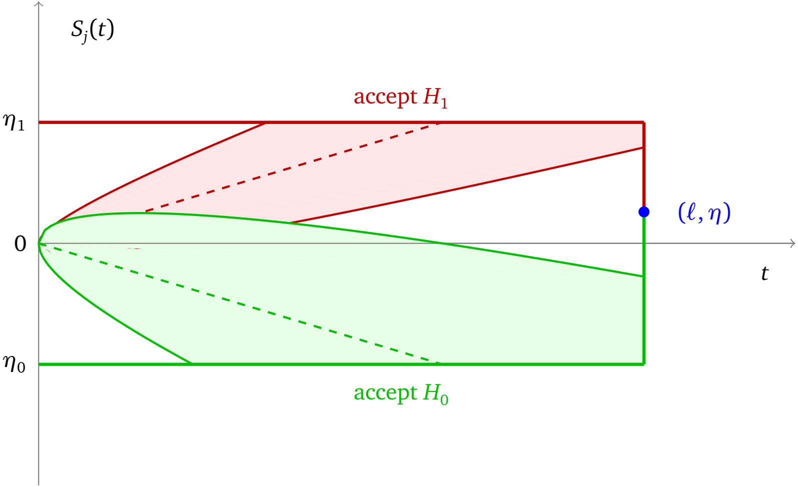
# Figures and tables

**Figure 5.1 (d)**



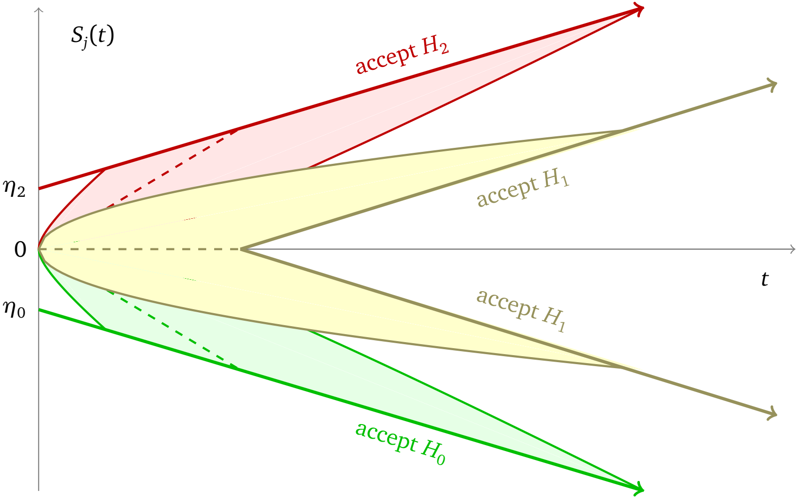
# Figures and tables

**Figure 5.1 (e)**



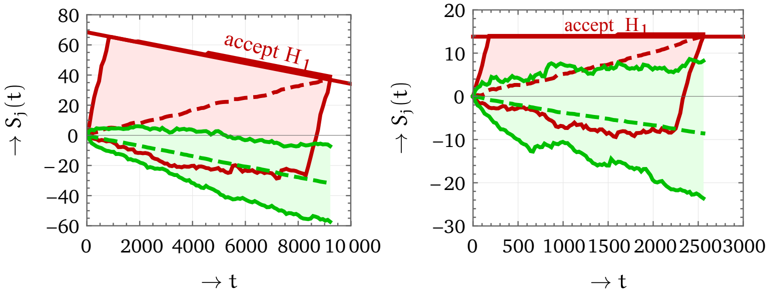
# Figures and tables

**Figure 5.1 (f)**



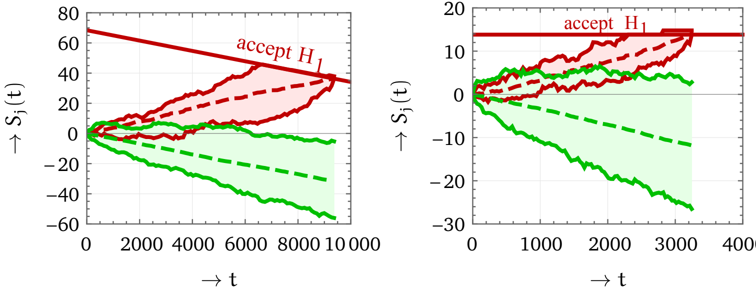
# Figures and tables

**Figure 5.2 (a-b)**



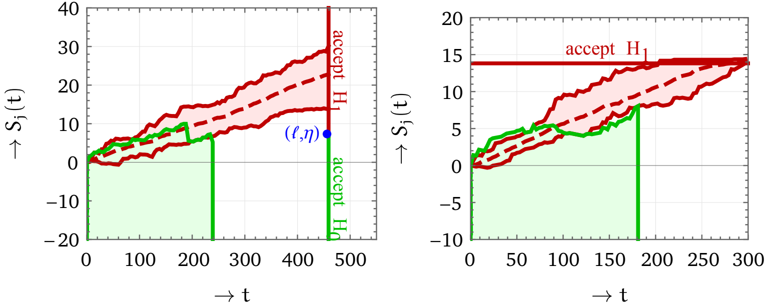
# Figures and tables

**Figure 5.2 (c-d)**



# Figures and tables

## Figure 5.2 (e-f)



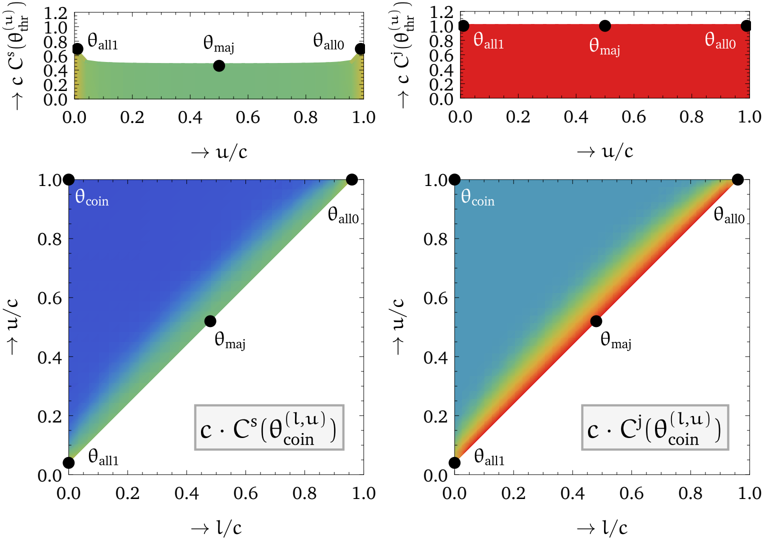
# Figures and tables

## Table 6.1

| Model | Simple capacities | Joint capacities |
|---|---|---|
| $\boldsymbol{\theta}_{\text{all1}}$: classical model | $(\ln 2)/c \approx 0.69/c$ | $(1)/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{add}}$: additive noise model | $(\ln 2 - r)/c \approx 0.69/c$ | $(1 - \frac{1}{2}h(r))/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{dil}}$: dilution noise model | $(\ln 2 - O(r \ln r))/c \approx 0.69/c$ | $(1 - \frac{1}{2}h(r)\ln 2)/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{thr}}^{(u)}$: threshold (no gap) | between $0.46/c$ and $0.69/c$ | $(1)/c \approx 1.00/c$ |
| $\boldsymbol{\theta}_{\text{int}}^{(l,u)}$: threshold (int. gap) | between $0.72/c^2$ and $0.69/c$ | between $0.84/c^2$ and $1.00/c$ |
| $\boldsymbol{\theta}_{\text{coin}}^{(l,u)}$: threshold (coin. gap) | between $0.17/c$ and $0.69/c$ | between $0.32/c$ and $1.00/c$ |

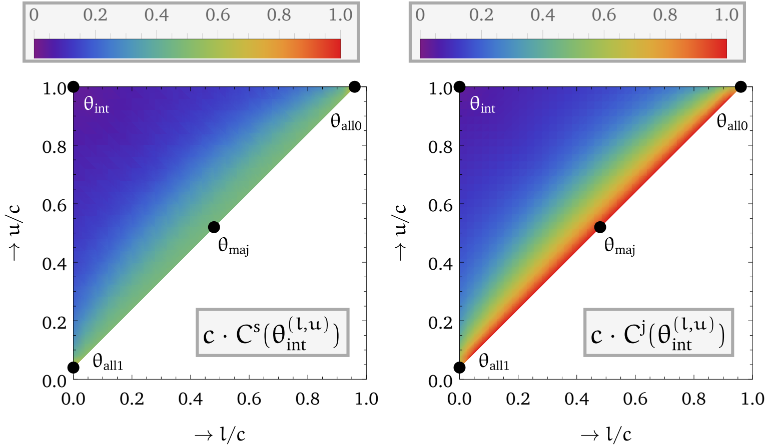
# Figures and tables

## Figure 6.1 (a-d)



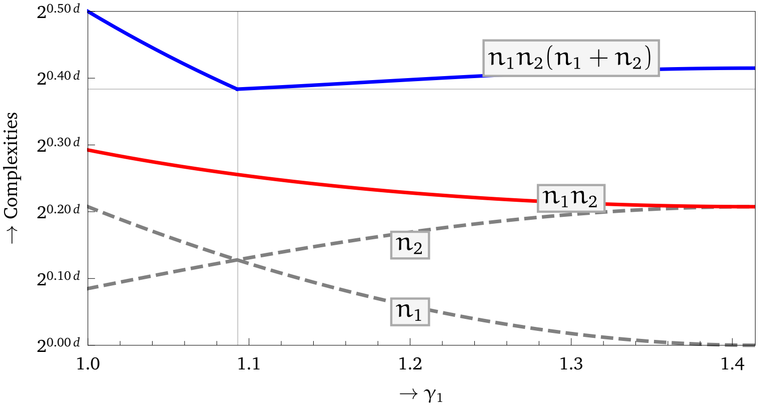
# Figures and tables

## Figure 6.1 (e-f)



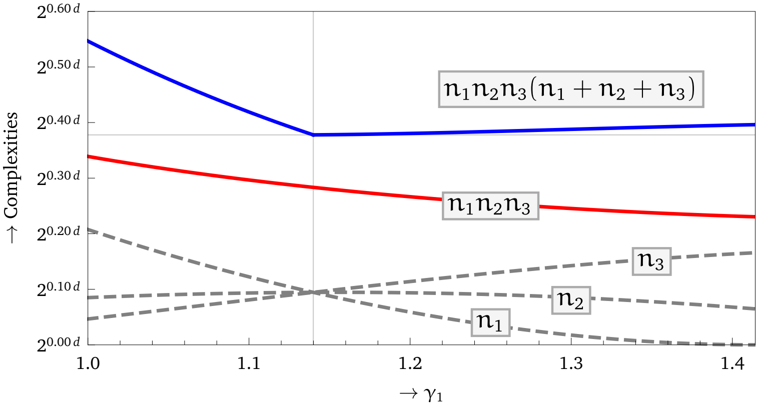
# Figures and tables

## Figure 9.1



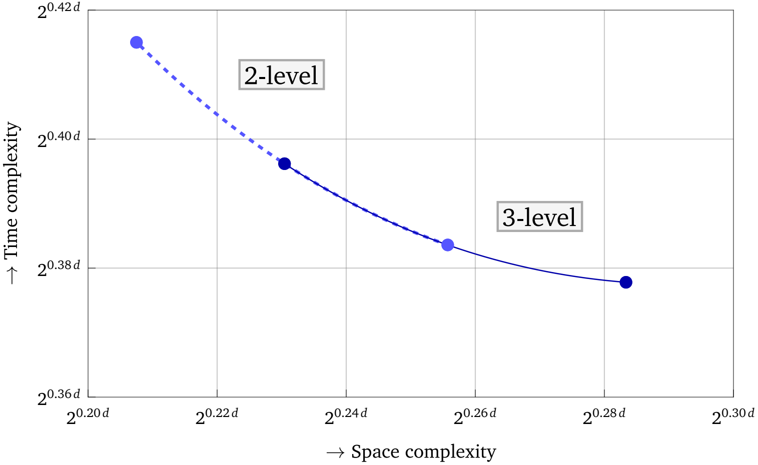
# Figures and tables

## Figure 9.2



# Figures and tables

## Figure 9.3



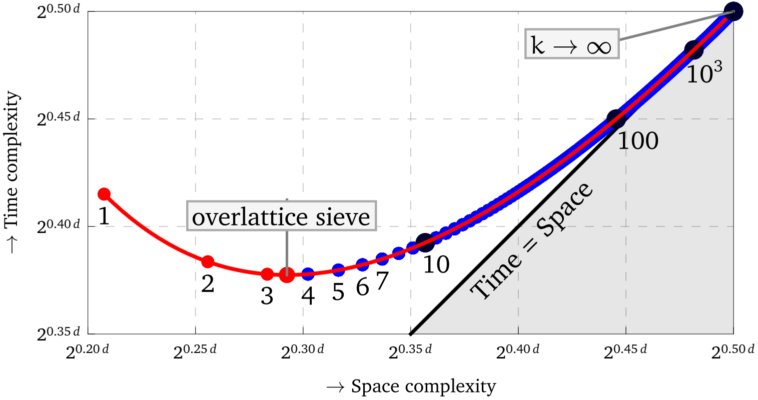
# Figures and tables

## Table 9.1

| Algorithm Name | Parameters Variables = Values | Polynomial $(x : p(x) = 0)$ | Complexities Time | Space | Exponents $c_{\text{time}}$ | $c_{\text{space}}$ |
|---|---|---|---|---|---|---|
| 1-level sieve | $(\gamma_1) = (1)$ | $x^2 - 4x^2 + 4$ | $x^{2n}$ | $x^n$ | 0.4150 | 0.2075 |
| 2-level sieve | $(\gamma_1, \gamma_2) = (x, 1)$ | $x^6 - 4x^4 + 4$ | $x^{3n}$ | $x^{2n}$ | 0.3836 | 0.2557 |
| 3-level sieve | $(\gamma_1, \gamma_2, \gamma_3) = (x^2, x, 1)$ | $x^{10} - 4x^6 + 4$ | $x^{4n}$ | $x^{3n}$ | 0.37780 | 0.2833 |
| 4-level sieve | $(\gamma_1, \dots, \gamma_4) = (x^3, \dots, 1)$ | $x^{14} - 4x^8 + 4$ | $x^{5n}$ | $x^{4n}$ | 0.37783 | 0.3023 |
| 5-level sieve | $(\gamma_1, \dots, \gamma_5) = (x^4, \dots, 1)$ | $x^{18} - 4x^{10} + 4$ | $x^{6n}$ | $x^{5n}$ | 0.3797 | 0.3164 |
| ... | | | | | | |

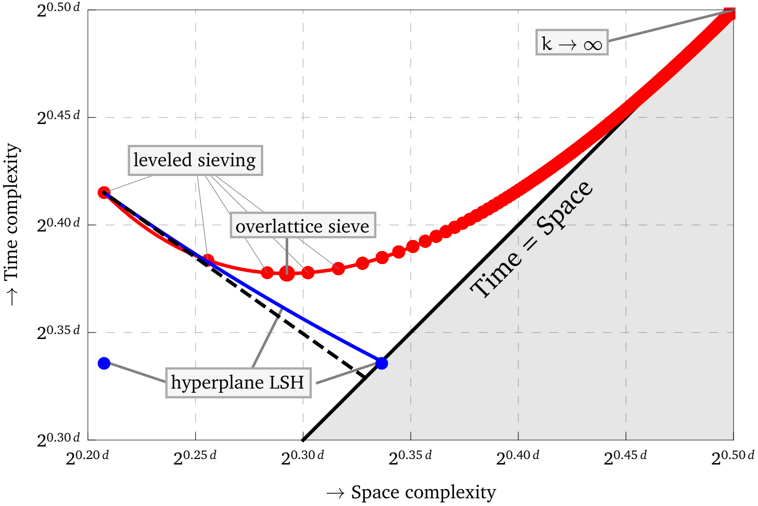
# Figures and tables

## Figure 9.4



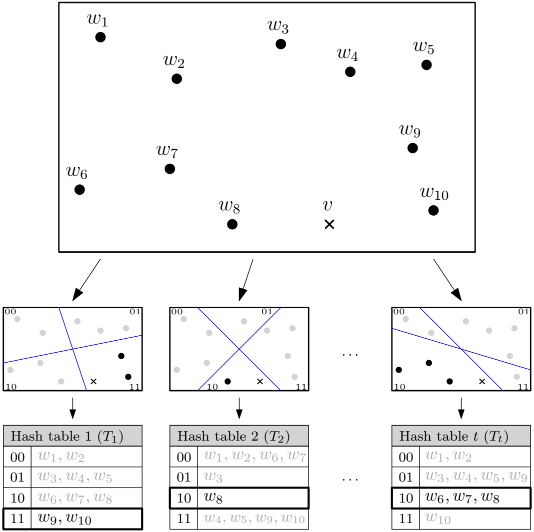
# Figures and tables

## Figure 10.1



# Figures and tables

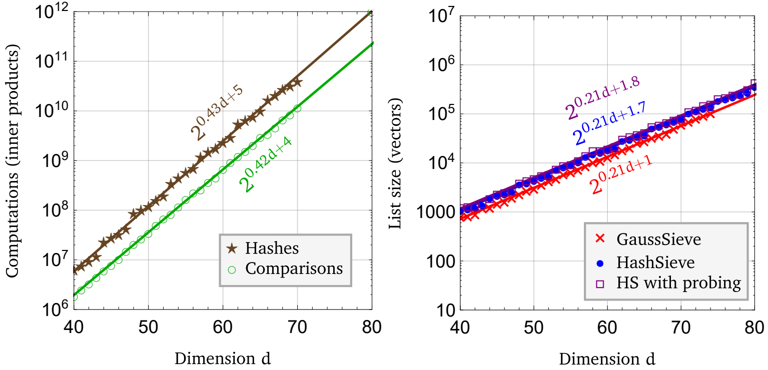
## Figure 10.2



# Figures and tables

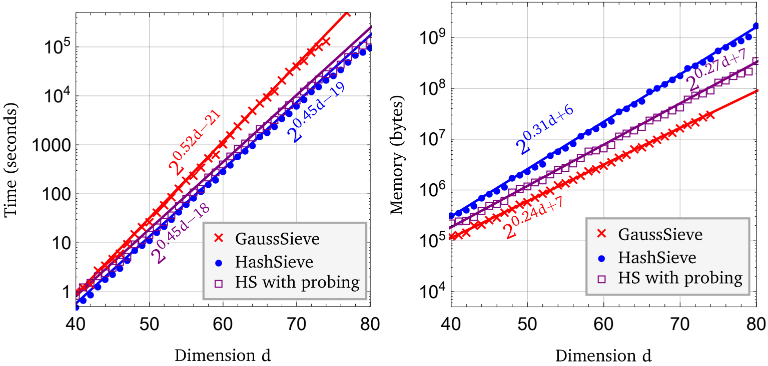
## Figure 10.4 (a-c)

| Dimension ($d$) | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 | 95 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hash length ($k$) | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 17 | 18 | 19 | 20 | 21 | 22 |
| Number of hash tables... | | | | | | | | | | | | | |
| ...without probing ($t$) | 36 | 56 | 87 | 137 | 214 | 334 | 523 | 817 | 1278 | 1999 | 3126 | 4888 | 7643 |
| ...with 1-level probing ($t_1$) | 7 | 9 | 13 | 20 | 29 | 42 | 62 | 86 | 128 | 190 | 284 | 425 | 637 |
| ...with 2-level probing ($t_2$) | 2 | 3 | 4 | 6 | 8 | 11 | 15 | 19 | 26 | 38 | 53 | 76 | 110 |



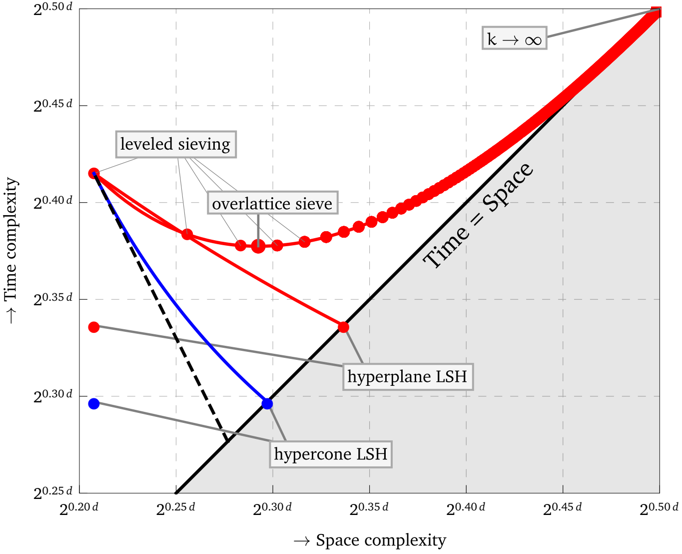
# Figures and tables

## Figure 10.4 (d-e)



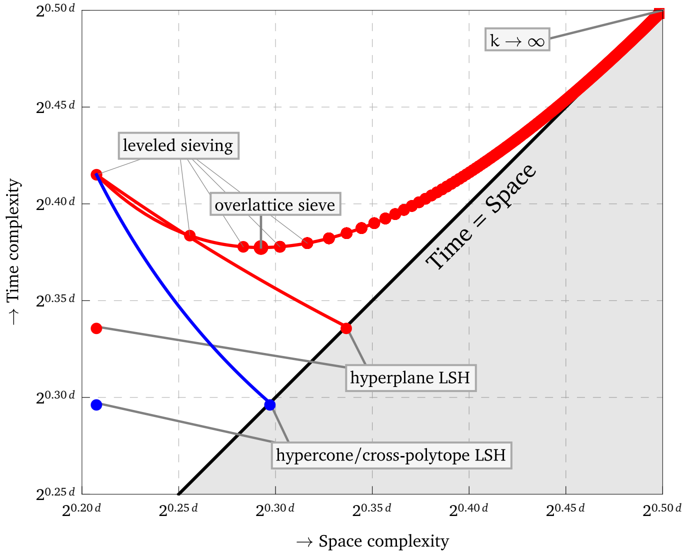
# Figures and tables

## Figure 11.1



# Figures and tables

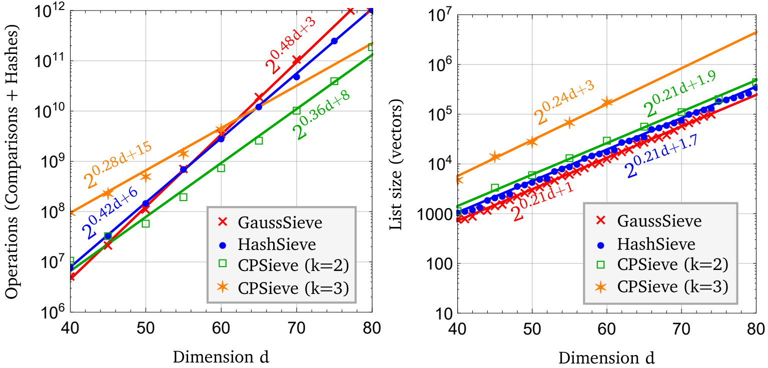
## Figure 12.1



# Figures and tables

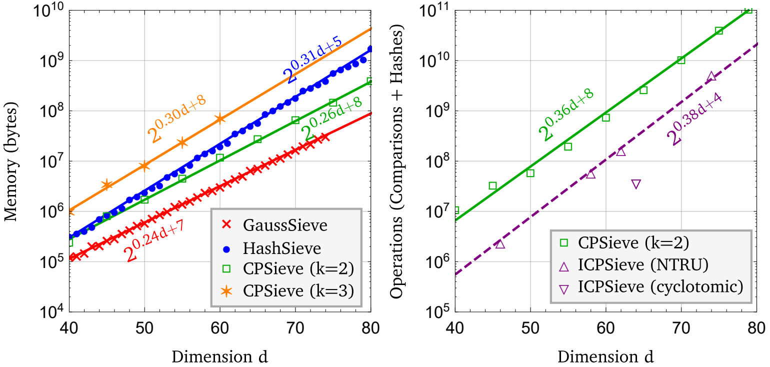
## Figure 12.2 (a-c)

| Dimension ($d$) | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 | 95 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hash length ($k$) | 2.0 | 2.2 | 2.4 | 2.6 | 2.7 | 2.9 | 3.1 | 3.2 | 3.4 | 3.6 | 3.7 | 3.9 | 4.1 |
| Hash tables ($t$) | 12 | 16 | 22 | 30 | 42 | 57 | 77 | 105 | 144 | 196 | 268 | 365 | 498 |



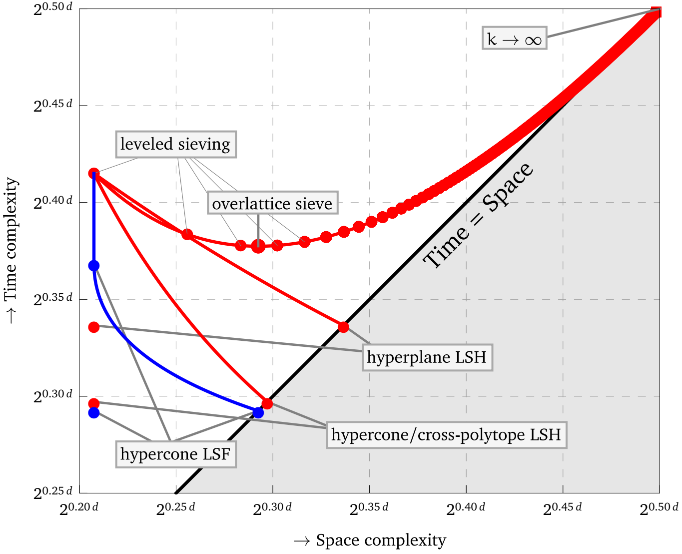
# Figures and tables

## Figure 12.2 (d-e)



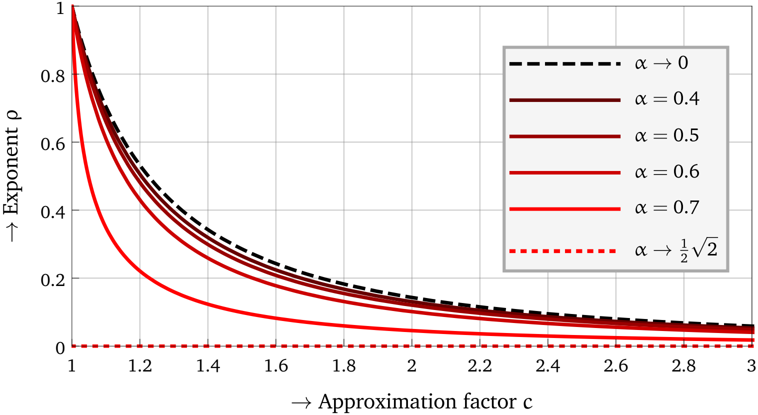
# Figures and tables

## Figure 13.1



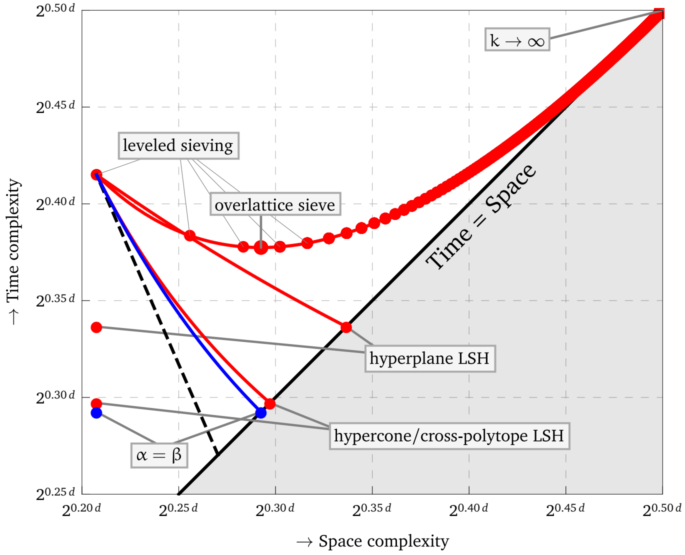
# Figures and tables

## Figure 13.2



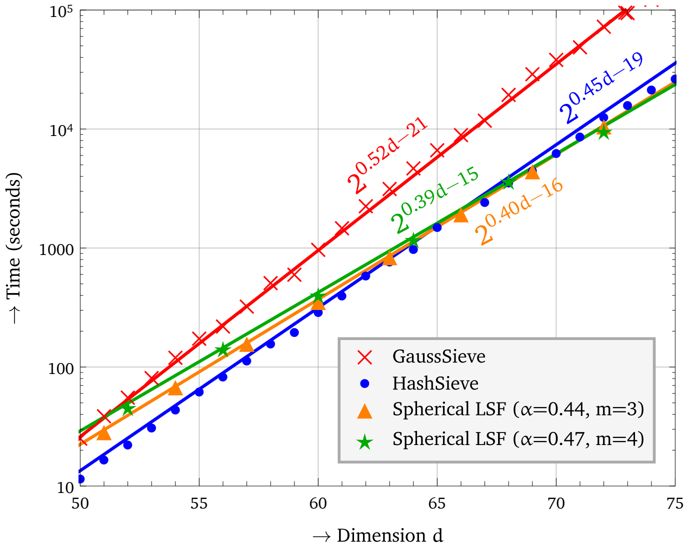
# Figures and tables

## Figure 13.3



# Figures and tables

## Figure 13.4



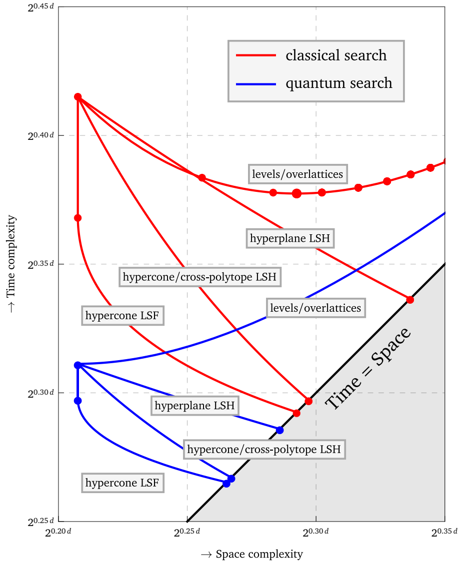
# Figures and tables

## Table 14.1

| | Algorithm | Classical search | | Quantum search | |
|---|---|---|---|---|---|
| | Name | $\log_2(\text{Time})$ | $\log_2(\text{Space})$ | $\log_2(\text{Time})$ | $\log_2(\text{Space})$ |
| **Heuristic SVP** | Nguyễn–Vidick sieve | $0.415d$ | $0.208d$ | $0.311d$ | $0.208d$ |
| | GaussSieve | $0.415d$ | $0.208d$ | $0.311d$ | $0.208d$ |
| | 2-level sieve | $0.384d$ | $0.256d$ | $0.311d$ | $0.208d$ |
| | 3-level sieve | $0.3778d$ | $0.283d$ | $0.311d$ | $0.208d$ |
| | Overlattice sieve | $0.3774d$ | $0.293d$ | $0.311d$ | $0.208d$ |
| | High-level sieving (Chapter 9) | $0.3774d$ | $0.293d$ | $0.311d$ | $0.208d$ |
| | Hyperplane LSH (Chapter 10) | $0.337d$ | $0.208d$ | $0.286d$ | $0.208d$ |
| | Hypercone LSH (Chapter 11) | $0.298d$ | $0.208d$ | $0.268d$ | $0.208d$ |
| | Cross-polytope LSH (Chapter 12) | $0.298d$ | $0.208d$ | $0.268d$ | $0.208d$ |
| | Hypercone filtering (Chapter 13) | $0.292d$ | $0.208d$ | $0.265d$ | $0.208d$ |

# Figures and tables

## Figure 14.1



# Figures and tables

## Table 14.2

| | Algorithm | Classical search | | Quantum search | |
|---|---|---|---|---|---|
| | Name | $\log_2$(Time) | $\log_2$(Space) | $\log_2$(Time) | $\log_2$(Space) |
| **Provable SVP** | Enumeration algorithms | $\Omega(d \log d)$ | $O(\log d)$ | $\Omega(d \log d)$ | $O(\log d)$ |
| | AKS-sieve | $3.398d$ | $1.985d$ | $2.672d$ | $1.877d$ |
| | ListSieve | $3.199d$ | $1.327d$ | $2.527d$ | $1.351d$ |
| | Voronoi cell algorithm | $2.000d$ | $1.000d$ | $2.000d$ | $1.000d$ |
| | AKS-sieve-birthday | $2.648d$ | $1.324d$ | $1.986d$ | $1.324d$ |
| | ListSieve-birthday | $2.465d$ | $1.233d$ | $1.799d$ | $1.286d$ |
| | Discrete Gaussian sampling | $1.000d$ | $0.500d$ | $1.000d$ | $0.500d$ |
| | ($\mathrm{SVP}_\delta$) ListSieve-birthday | $0.802d$ | $0.401d$ | $0.602d$ | $0.401d$ |