An introduction to lattice-based cryptography.

Andrea Lesavourey

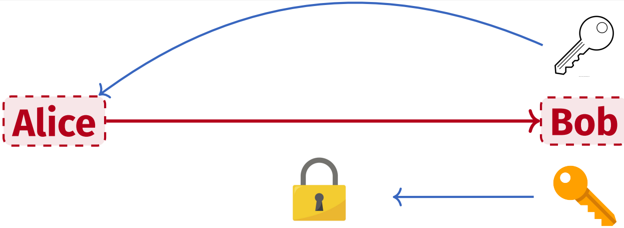
INRIA Bordeaux

May 15, 2024

STORM
Static Optimizations – Runtime Methods

Introduction

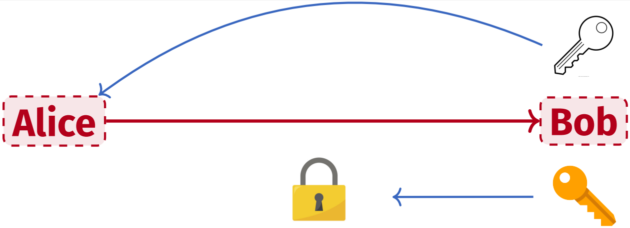
Cryptographie à clef publique



Security based on a ***hard mathematical problem.***

Exemples : Factorisation (RSA) ou Logarithme discret (courbes elliptiques).

Cryptographie à clef publique

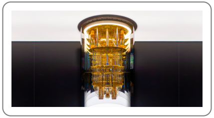


Security based on a ***hard mathematical problem.***

Exemples : Factorisation (RSA) ou Logarithme discret (courbes elliptiques).

Applications :

Cryptographie post-quantique



Problem : Shor's algorithms
Quantum polynomial time.

Need for a **post-quantum** cryptography :
classical computations;
safe under quantum attacks.

Euclidean lattices, Error correcting codes,
Polynomial systems, Hash functions
Algebraic variety (elliptic curves).

Calls for standardisation

NIST in 2016.

End (almost) of the process.

Encryption schemes :
Lattices : KYBER.

Signatures :
Lattices : DILITHIUM, FALCON.
Hash functions : SPHINCS+.

Un round de plus :
Codes : BIKE, CLASSIC MCELIECE, HQC

Outline of the presentation

1. Quantum computing and Shor's algorithm.

2. Lattice-based cryptography.

Quantum Computing

Quantum bits

- One bit : 0 or 1

Quantum bits

- One bit : 0 or 1

  One quantum bit or qubit : $\alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$

Quantum bits

- One bit : 0 or 1

  One quantum bit or qubit : $\alpha\,|0\rangle + \beta\,|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$

- Two bits : 00, 01, 10, 11

Quantum bits

- One bit : 0 or 1

  One quantum bit or qubit : $\alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$

- Two bits : 00, 01, 10, 11

  Two qubits : $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ with $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$

Quantum bits

- One bit : 0 or 1

  One quantum bit or qubit : $\alpha |0\rangle + \beta |1\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$

- Two bits : 00, 01, 10, 11

  Two qubits : $\alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle$ with $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$

- n bits : $i_1 i_2 \cdots i_n$

Quantum bits

- One bit : 0 or 1

  One quantum bit or qubit : $\alpha\left|0\right\rangle + \beta\left|1\right\rangle$ with $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$

- Two bits : 00, 01, 10, 11

  Two qubits : $\alpha\left|00\right\rangle + \beta\left|01\right\rangle + \gamma\left|10\right\rangle + \delta\left|11\right\rangle$ with $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$

- n bits : $i_1 i_2 \cdots i_n$

  n qubits : $\sum_{i=0}^{2^n-1} \alpha_i \left|i\right\rangle$ with $\alpha_i \in \mathbb{C}$ such that $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$

Operations

Evolution of a quantum system : described by a unitary operator $U \in U_{2^n}(\mathbb{C})$.

Typical examples for a single qubit include :

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix}$$

$$H(\alpha \left|0\right\rangle + \beta \left|1\right\rangle) = \alpha(\tfrac{1}{\sqrt{2}} \left|0\right\rangle + \tfrac{1}{\sqrt{2}} \left|1\right\rangle) + \beta(\tfrac{1}{\sqrt{2}} \left|0\right\rangle - \tfrac{1}{\sqrt{2}} \left|1\right\rangle)$$

Superposition allows quick multi-evaluation

Measurements

Quantum measurements : set $\{M_m\}$ of measurement operators. m are the possible outcomes

- $|\psi\rangle \longrightarrow \mathbb{P}(m) = \| M_m |\psi\rangle \|^2$
- $|\psi\rangle \longmapsto \dfrac{M_m |\psi\rangle}{\sqrt{\| M_m |\psi\rangle \|}}$

In general : operators correspond to canonical basis

Example

For $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

- $\mathbb{P}(0) = \mathbb{P}(1) = \frac{1}{2}$
- If 0 measured then $|\psi\rangle = |0\rangle$

Example

For $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

- $\mathbb{P}(0) = \mathbb{P}(1) = \frac{1}{2}$
- If 0 measured then $|\psi\rangle = |0\rangle$

For $|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle) + \frac{1}{\sqrt{2}}|11\rangle$

- Measure the second register : $P(1) = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$
- If 1 measured then $|\psi\rangle = \frac{1}{\sqrt{3}}|01\rangle + \frac{\sqrt{2}}{\sqrt{3}}|11\rangle$

Fast computation

Quantum superposition : allows fast computation by multi-evaluation.

Fast computation

Quantum superposition : allows fast computation by multi-evaluation.

$U = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ and $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$ then applying U gives

$$\frac{1}{\sqrt{2}}(|10\rangle + |11\rangle)$$

Fast computation

Consider $f : \{0,1\}^n \to \{0,1\}^m$.

Assume there is a unitary transform

$$U_f : |x\rangle\,|y\rangle \longmapsto |x\rangle\,|y \oplus f(x)\rangle\,.$$

Fast computation

Consider $f : \{0,1\}^n \to \{0,1\}^m$.

Assume there is a unitary transform

$$U_f : |x\rangle |y\rangle \longmapsto |x\rangle |y \oplus f(x)\rangle .$$

$$\sum_x \alpha_x |x\rangle |0\rangle$$

Fast computation

Consider $f : \{0,1\}^n \to \{0,1\}^m$.

Assume there is a unitary transform

$$U_f : |x\rangle\,|y\rangle \longmapsto |x\rangle\,|y \oplus f(x)\rangle\,.$$

$$U_f \cdot \sum_x \alpha_x\,|x\rangle\,|0\rangle = \underbrace{\sum_x \alpha_x\,|x\rangle\,|f(x)\rangle}_{\text{all values } f(x) \text{ are present}}$$

Fast computation

Consider $f : \{0,1\}^n \to \{0,1\}^m$.

Assume there is a unitary transform

$$U_f : |x\rangle\,|y\rangle \longmapsto |x\rangle\,|y \oplus f(x)\rangle\,.$$

$$U_f \cdot \sum_x \alpha_x\,|x\rangle\,|0\rangle \;= \underbrace{\sum_x \alpha_x\,|x\rangle\,|f(x)\rangle}_{\text{all values } f(x) \text{ are present}}$$

Problem : Find the desired information through measurement.

Grover's algorithm

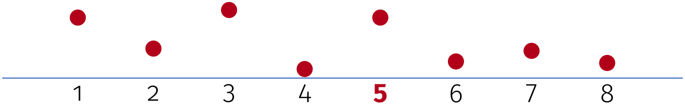
Our goal is to find *one* element within a set of size $N (= 2^n)$.

Assume as well that we have access to an oracle $\mathcal{O}$, efficiently computable.

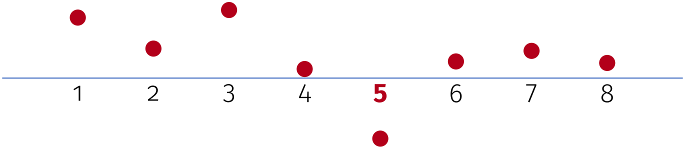
We will use two operators :

1. $U_{\mathcal{O}} : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus \mathcal{O}(x)\rangle$. *(Call to oracle)*
2. $S : \sum_x \alpha_x |x\rangle \mapsto \sum_x (2\bar{\alpha} - \alpha_x) |x\rangle$. *(Symmetry around mean of amplitudes)*

Grover's algorithm



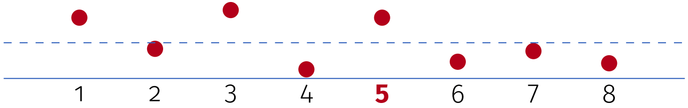
Grover's algorithm



When $|y\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$,

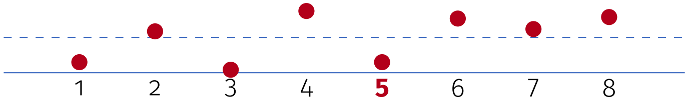
$$U_{\mathcal{O}} \sum_x \alpha_x |x\rangle |y\rangle = \sum_x (-1)^{\mathcal{O}(x)} \alpha_x |x\rangle |y\rangle$$

Grover's algorithm



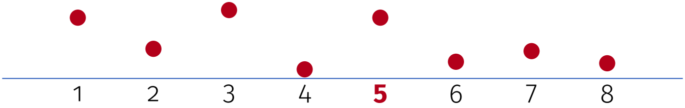
S operates a symmetry around the average amplitude !

Grover's algorithm



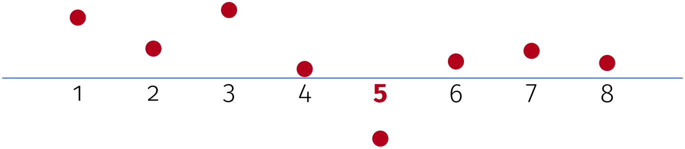
S operates a symmetry around the average amplitude !

Grover's algorithm



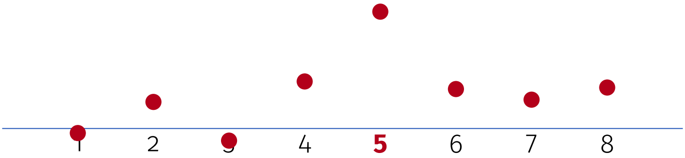
What happens when we apply $U_{\mathcal{O}}$ and S one after another ?

Grover's algorithm



What happens when we apply $U_{\mathcal{O}}$ and S one after another ?

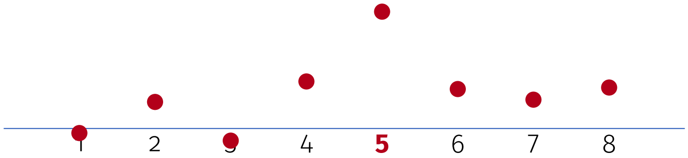
Grover's algorithm



What happens when we apply $U_{\mathcal{O}}$ and S one after another ?

Amplification of amplitude !

Grover's algorithm



What happens when we apply $U_{\mathcal{O}}$ and S one after another ?

Amplification of amplitude !

Need around $\sqrt{N}$ iterations to retrieve the solution with a high enough probability.

Shor's algorithm

There are **two** core ingredients of Shor's algorithms :

1. the fast computation of a Quantum Fourier Transform (QFT) ;
2. the computation of the hidden period of a given function f.

First let us denote by ζ_N a Nth root of unity, i.e. $\zeta_N = \exp 2i\pi/N$.

In the classical setting, we have the *Discrete Fourier Transform* :

$$DFT : (x_0, \ldots, x_{N-1}) \mapsto (y_0, \ldots, y_{N-1})$$

with

$$y_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} x_i \cdot \zeta_N^{-i \cdot k}.$$

First let us denote by ζ_N a Nth root of unity, i.e. $\zeta_N = \exp 2i\pi/N$.

In the classical setting, we have the *Discrete Fourier Transform* :

$$DFT : (x_0, \dots, x_{N-1}) \mapsto (y_0, \dots, y_{N-1})$$

with

$$y_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} x_i \cdot \zeta_N^{-i \cdot k}.$$

In the quantum setting, we have the *Quantum Fourier Transform* :

$$QFT : \sum_{i=0}^{N-1} x_i \left| i \right\rangle \mapsto \sum_{i=0}^{N-1} y_i \left| i \right\rangle$$

with

$$y_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} x_i \cdot \zeta_N^{i \cdot k}.$$

We can *factorise* the QFT :

$$QFT : \sum_{i=0}^{N-1} x_i \left|i\right\rangle \mapsto \frac{1}{\sqrt{N}} \bigotimes_{i=1}^{n} \left(\left|0\right\rangle + \zeta_N^{x \cdot 2^{n-i}} \left|1\right\rangle \right).$$

If we adopt the notation $[x_1, \cdots x_k] = \sum_{i=1}^{k} x_i \cdot 2^{-i}$, we also have :

$$QFT : \sum_{i=0}^{N-1} x_i \left|i\right\rangle \mapsto \frac{1}{\sqrt{N}} \bigotimes_{j=1}^{n} \left(\left|0\right\rangle + e^{2i\pi[x_{n-j+1}, \ldots, x_n]} \left|1\right\rangle \right).$$

We can *factorise* the QFT :

$$QFT : \sum_{i=0}^{N-1} x_i |i\rangle \mapsto \frac{1}{\sqrt{N}} \bigotimes_{i=1}^{n} \left(|0\rangle + \zeta_N^{x \cdot 2^{n-i}} |1\rangle \right).$$

If we adopt the notation $[x_1, \cdots x_k] = \sum_{i=1}^{k} x_i \cdot 2^{-i}$, we also have :

$$QFT : \sum_{i=0}^{N-1} x_i |i\rangle \mapsto \frac{1}{\sqrt{N}} \bigotimes_{j=1}^{n} \left(|0\rangle + e^{2i\pi[x_{n-j+1}, \dots, x_n]} |1\rangle \right).$$

This can be computed by successive application of rotation gates :

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp(2i\pi/2^k) \end{bmatrix}$$

We can *factorise* the QFT :

$$QFT : \sum_{i=0}^{N-1} x_i \left|i\right\rangle \mapsto \frac{1}{\sqrt{N}} \bigotimes_{i=1}^{n} \left(\left|0\right\rangle + \zeta_N^{x \cdot 2^{n-i}} \left|1\right\rangle\right).$$

If we adopt the notation $[x_1, \cdots x_k] = \sum_{i=1}^{k} x_i \cdot 2^{-i}$, we also have :

$$QFT : \sum_{i=0}^{N-1} x_i \left|i\right\rangle \mapsto \frac{1}{\sqrt{N}} \bigotimes_{j=1}^{n} \left(\left|0\right\rangle + e^{2i\pi[x_{n-j+1}, \ldots, x_n]} \left|1\right\rangle\right).$$

This can be computed by successive application of rotation gates :

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp(2i\pi/2^k) \end{bmatrix}$$

We obtain a circuit with $O(n^2)$ gates, where $N = 2^n$ i.e. $O(\log N)$ gates.

We are given a r-periodic function f efficiently computable through U_f and we wish to recover r.

1. Prepare the state $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |0\rangle$.
2. Apply f as $U_f |\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |f(x)\rangle$.
3. Measure wrt to the 2nd register : $\frac{1}{\sqrt{N/r}} \sum_{k=0}^{N/r-1} |x_0 + k \cdot r\rangle$ for a given x_0.
4. Apply the QFT : $\frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \alpha_j \, |j \frac{N}{r}\rangle$.
5. Measure to obtain $jN/r \implies j/r$; if $\gcd(j, r) = 1$ then r can be recovered efficiently.

Shor's algorithm

Conclusion

This fast period-finding strategy can be applied to :

- factorise integers;
- solve the DLP;
- solve the phase estimation problem.

This fast period-finding strategy can be applied to :

- factorise integers;
- solve the DLP;
- solve the phase estimation problem.

There is more ! Generalisation of this approach can be used to solve classical number theoretical problems, such as :

- the computation of (S-)units of a number field;
- determination of the class group;
- finding the generator of a principal ideal $I = (g)$.

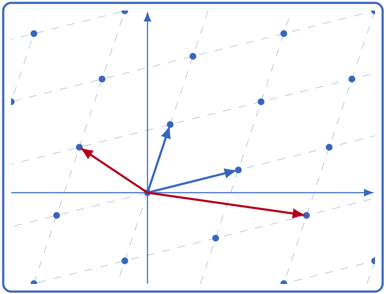
Conclusion

- Superposition : fast multi-evaluation
- Quantum Fourier Transform : detect period
  - Almost all of exponential speed-ups
- Problem : Find desired result without structure
  - Search algorithm : only quadratic speed-up

Euclidean lattices

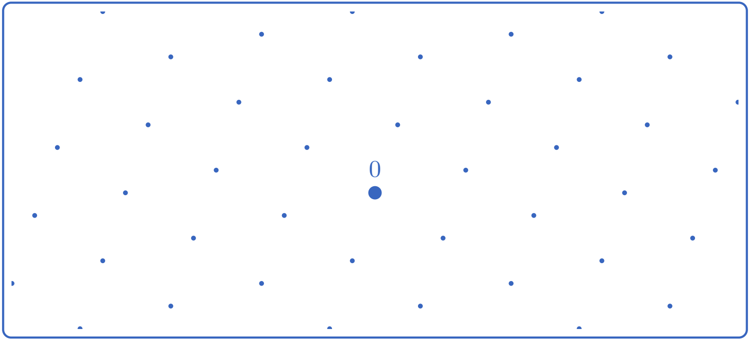
Definition

We call *lattice* any discrete subgroup $\mathcal{L}$ of $\mathbb{R}^n$ where n is a positive integer.

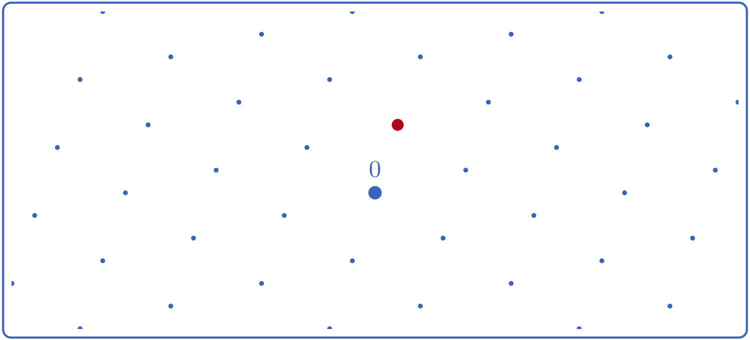


- Any set B of free vectors which generates $\mathcal{L}$ is called a basis.
- There are infinitely many bases.
- Some are better than others : orthogonality, short vectors

Problems on lattices



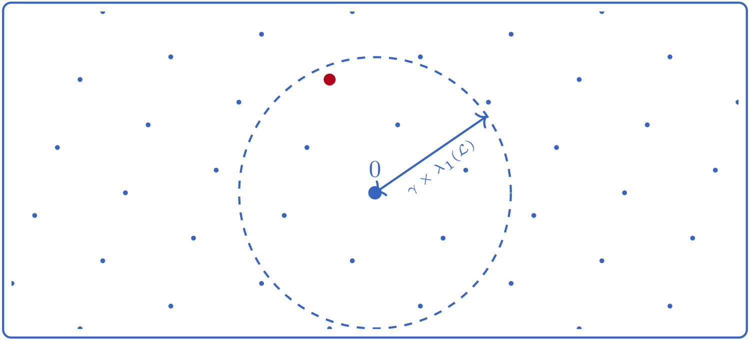
Problems on lattices



Shortest Vector Problem (SVP) : Find a shortest vector of $\mathcal{L} \setminus \{0\}$.

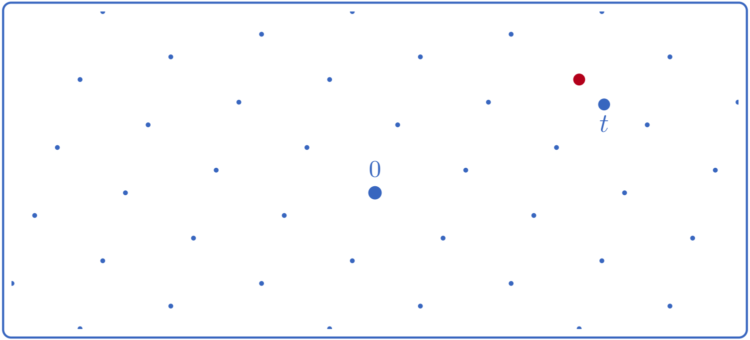
Note $\lambda_1(\mathcal{L})$ its norm.

Problems on lattices



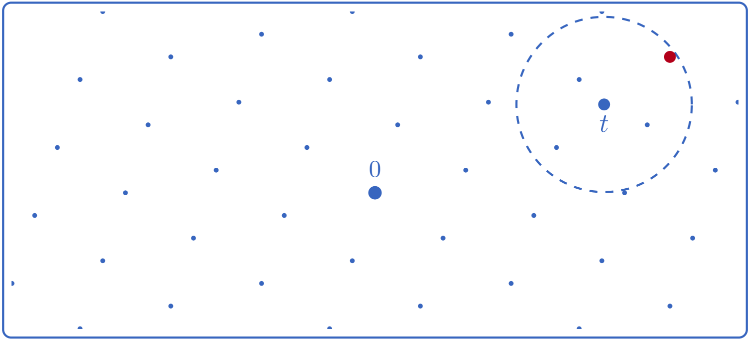
Approximate Shortest Vector Problem (Approx-SVP) : Find a vector of $\mathcal{L}$ with norm less than $\gamma \times \lambda_1(\mathcal{L})$.

Problems on lattices



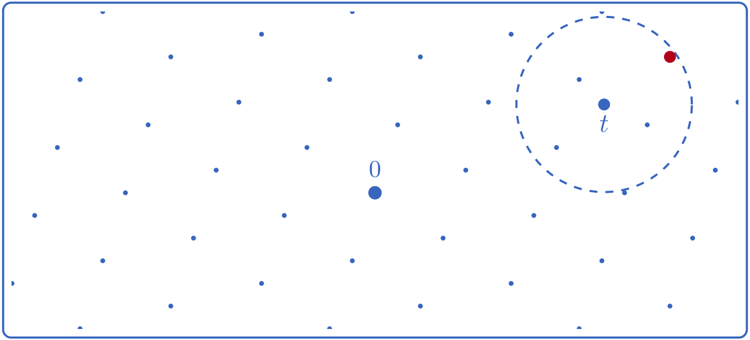
Closest Vector Problem (CVP): Given $\mathbf{t}$ a target vector, find a vector of $\mathcal{L}$ *closest* to $\mathbf{t}$

Problems on lattices

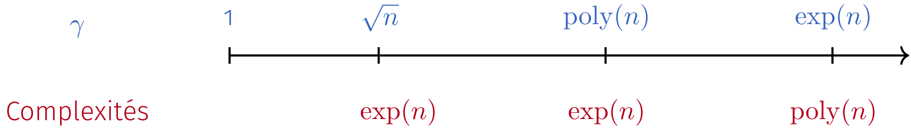


Approximate Closest Vector Problem (Approx-CVP): Given $\mathbf{t}$ a target vector, find a vector of $\mathcal{L}$ within distance $\gamma \times d(\mathbf{t}, \mathcal{L})$ of $\mathbf{t}$

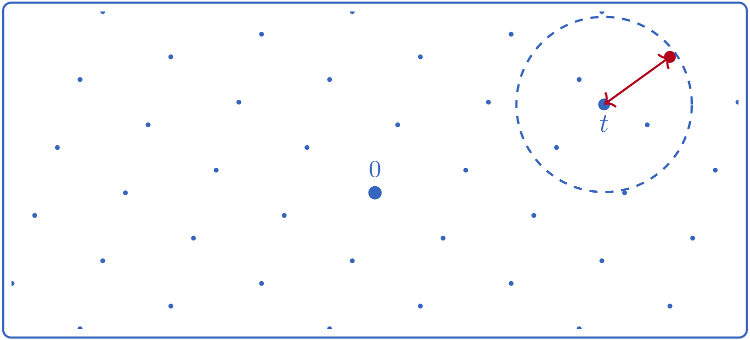
Problems on lattices



Approximate Closest Vector Problem (Approx-CVP): Given $\mathbf{t}$ a target vector, find a vector of $\mathcal{L}$ within distance $\gamma \times d(\mathbf{t}, \mathcal{L})$ of $\mathbf{t}$

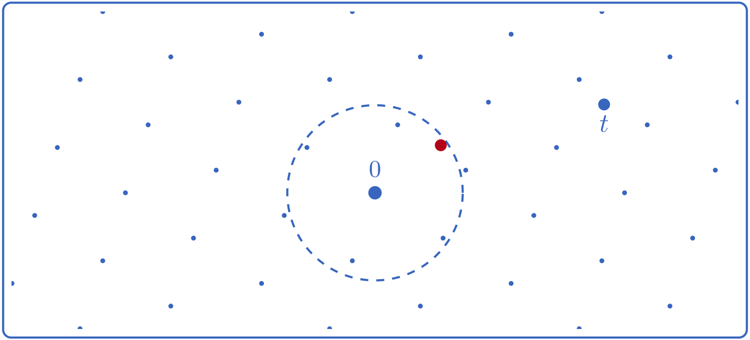


Problems on lattices



Approximate Closest Vector Problem (Approx-CVP): Given $\mathbf{t}$ a target vector, find a vector of $\mathcal{L}$ within distance $\gamma \times d(\mathbf{t}, \mathcal{L})$ of $\mathbf{t}$

Problems on lattices



Approximate Closest Vector Problem (Approx-CVP): Given $\mathbf{t}$ a target vector, find a vector of $\mathcal{L}$ within distance $\gamma \times d(\mathbf{t}, \mathcal{L})$ of $\mathbf{t}$

Equivalently, find small $\mathbf{t}' \equiv \mathbf{t} \bmod \mathcal{L}$ $\rightarrow$ **reduction modulo $\mathcal{L}$**

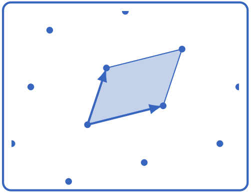
Guaranteed Distance Decoding (GDD) : Given *any* vector $\mathbf{t}$ in $\mathrm{span}(\mathcal{L})$, find $\mathbf{t}' \equiv \mathbf{t} \bmod \mathcal{L}$ such that $\|\mathbf{t}'\| \leqslant \gamma\lambda_1(\mathcal{L})$.(knowing that it exists)

Reducing modulo a lattice

Fix $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ a basis of $\mathcal{L}$ and $\mathbf{t} \in \mathbb{R} \cdot \mathbf{b}_1 \oplus \cdots \oplus \mathbb{R} \cdot \mathbf{b}_n$.

Write $\mathbf{t} = \sum_{i=1}^{n} \mathbf{t}_i \cdot \mathbf{b}_i$, with $t_i \in \mathbb{R}$.

Two main algorithms used in practice :



Babaï's round-off

Output $\sum_{i=1}^{n} (\mathbf{t}_i - \lfloor \mathbf{t}_i \rceil) \cdot \mathbf{b}_i$;

Ensure that the output is in $[-1/2, 1/2[^n \times \mathbf{B}$.

Babaï's nearest plane

Use the GSO $\tilde{\mathbf{B}}$ instead;

Ensure that the output is in $[-1/2, 1/2[^n \times \tilde{\mathbf{B}}$.

GGH-like schemes

Lattice-based cryptography : GGH-like schemes

Encryption

PUBLIC KEY : a "bad" basis $\mathbf{H}$, typically the HNF.

SECRET KEY : a "good" basis, which is a trapdoor for the problem.

ENCRYPTION : $\mathbf{c} = \mathrm{Encrypt}(\mathbf{m}, \mathbf{H}) = s \cdot \mathbf{H} + \mathbf{m}$ where $s \in \mathbb{Z}^n$ and $\mathbf{m}$ is short.

DECRYPTION : $\mathrm{Decrypt}(\mathbf{c}, \mathbf{B}) = \mathtt{Reduce}(\mathbf{c}, \mathbf{B})$ ▷ GDD solver

Assume that :

- $\|\mathbf{m}\| < M$; $\rightarrow$ bound on the message space
- $\|\mathtt{Reduce}(\mathbf{t}, \mathcal{L})\| < R$. $\rightarrow$ bound on the reduction capacity

If $R + M < \lambda_1(\mathcal{L})$ then $\mathtt{Reduce}(\mathbf{c}, \mathcal{L}) = \mathbf{m}$.

PUBLIC KEY : a "bad" basis $\mathbf{H}$, typically the HNF.

SECRET KEY : a "good" basis $\mathbf{B}$, which is the trapdoor of the problem.

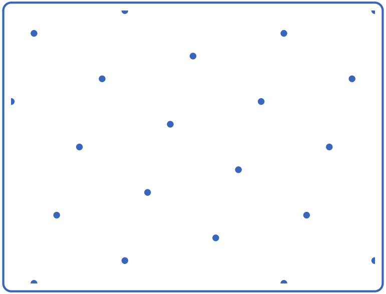
SIGNATURE : $\mathbf{s} = \mathrm{Sign}(\mathbf{m}, \mathbf{B}) = \mathtt{Reduce}(\mathbf{m}, \mathbf{B})$.

VERIFICATION : $\mathbf{s}$ is short and $\mathbf{s} - \mathbf{m} \in \mathcal{L}$.

Problem: Babaï's algorithms leak the secret basis !

- GGH and original NTRUsign use Babaï's round-off;
- Works also on more complex structures (zonotopes);
- Works with more general distribution.

Nguyen-Regev statistical attack

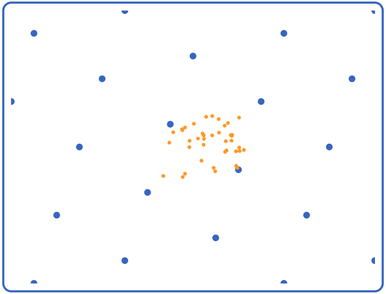


- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^\top] = \mathbf{B} \cdot \mathbf{B}^\top$;
- We can do as follows :
  1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^\top$;
  2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
  3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Nguyen-Regev statistical attack

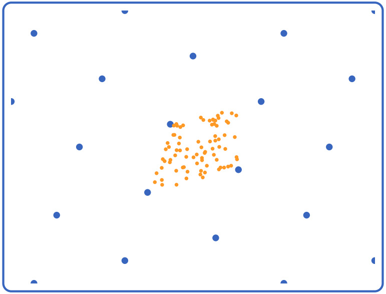


- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^{\intercal}] = \mathbf{B} \cdot \mathbf{B}^{\intercal}$;
- We can do as follows :
    1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^{\intercal}$;
    2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
    3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Nguyen-Regev statistical attack

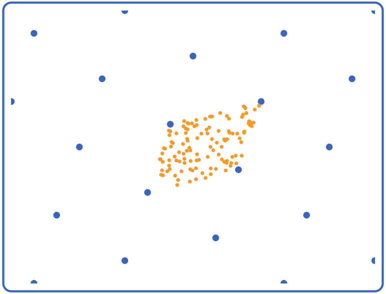


- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^\top] = \mathbf{B} \cdot \mathbf{B}^\top$;
- We can do as follows :
  1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^\top$;
  2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
  3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Nguyen-Regev statistical attack

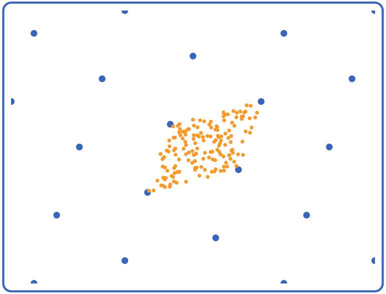


- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^\intercal] = \mathbf{B} \cdot \mathbf{B}^\intercal$;
- We can do as follows :
  1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^\intercal$;
  2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
  3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Nguyen-Regev statistical attack

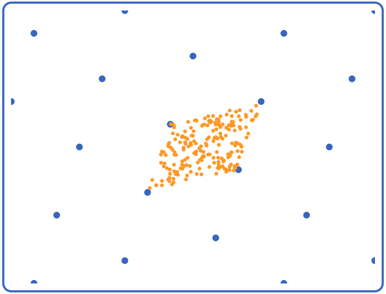


- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^\top] = \mathbf{B} \cdot \mathbf{B}^\top$;
- We can do as follows :
  1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^\top$;
  2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
  3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Nguyen-Regev statistical attack

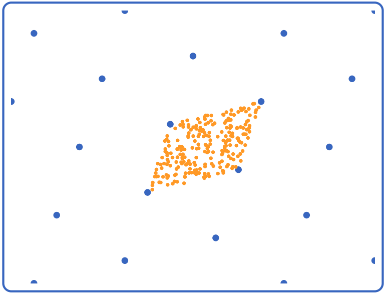


- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^\top] = \mathbf{B} \cdot \mathbf{B}^\top$;
- We can do as follows :
  1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^\top$;
  2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
  3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Nguyen-Regev statistical attack



- $\mathbb{E}[\mathbf{s} \cdot \mathbf{s}^\intercal] = \mathbf{B} \cdot \mathbf{B}^\intercal$;
- We can do as follows :
  1. compute an amproximation of $\mathbf{B} \cdot \mathbf{B}^\intercal$;
  2. find an approximate secret vector with a gradient descent; draw random vector and minimise the 4th moment
  3. recover the secret vector with one of Babaï's algos.

Counter-measure : Draw from distribution independent of the secret basis : discrete gaussian as in [GPV08]

Cons : not *that* efficient and requires floats.

Recent lattice-based cryptography

Lattice-based cryptography[1]

Running time		$\exp(n)$	$\exp(n)$	$\text{poly}(n)$
γ	1	$\sqrt{n}$	$\text{poly}(n)$	$\exp(n)$
Hardness	NP-hard	NP∩coNP	CRYPTO	P

Use intermediate problems

Short Integer Solution (SIS)

Learning With Errors (LWE)

[1] Freely taken from A. Roux-Langlois

Short Integer Solution (SIS)

Fix $q, n \in \mathbb{N}$.

`Input:` $A \overset{\mathcal{U}}{\leftarrow} \mathrm{M}_n(\mathbb{Z}/q\mathbb{Z})$

`Goal:` Find ***short*** $s \in \mathbb{Z}^n \mid As = 0 \bmod q$

Learning With Error (LWE)

Fix $q, n, m \in \mathbb{N}$.

`Input:` $(A, \quad b = As + e)$,

where $A \quad \overset{\mathcal{U}}{\leftarrow} \quad \mathrm{M}_{m,n}(\mathbb{Z}/q\mathbb{Z})$,

$s \overset{\mathcal{D}_s}{\leftarrow} (\mathbb{Z}/q\mathbb{Z})^n, e \overset{\mathcal{D}_e}{\leftarrow} \mathbb{Z}^m$

`Goal:` Find s.

Lattice-based cryptography

SIS and LWE : Two good average case problems

Short Integer Solution (SIS)

Fix $q, n \in \mathbb{N}$.

`Input:` $A \xleftarrow{\mathcal{U}} \mathrm{M}_n(\mathbb{Z}/q\mathbb{Z})$

`Goal:` Find ***short*** $s \in \mathbb{Z}^n \mid As = 0 \bmod q$

Worst-case to average-case

Approx-SVP
$\gamma > \sqrt{n}$

Learning With Error (LWE)

Fix $q, n, m \in \mathbb{N}$.

`Input:` $(A, \quad b = As + e)$,
where $A \xleftarrow{\mathcal{U}} \mathrm{M}_{m,n}(\mathbb{Z}/q\mathbb{Z})$,
$s \xleftarrow{\mathcal{D}_s} (\mathbb{Z}/q\mathbb{Z})^n, e \xleftarrow{\mathcal{D}_e} \mathbb{Z}^m$

`Goal:` Find s.

A closer look at LWE

Problem: Solve a system of m approximate equations in n variables modulo an integer q.

$$s_1 + 2s_2 + 4s_3 \approx 2 \bmod 5$$

$$3s_1 + 4s_2 + 2s_3 \approx 1 \bmod 5$$

$$s_2 + 2s_3 \approx 4 \bmod 5$$

$$2s_1 + 3s_3 \approx 2 \bmod 5$$

$$4s_1 + 2s_2 + 2s_3 \approx 3 \bmod 5$$

A closer look at LWE

More formally, we fix $n \geqslant 1$, $q \geqslant 2$ and $\alpha \in]0, 1[$.

Given $\mathbf{s} = [s_1, \ldots, s_n] \in (\mathbb{Z}/q\mathbb{Z})^n$, we define a LWE sample to be :

$$(\mathbf{a}, (\mathbf{a} \mid \mathbf{s}) + e)\,,$$

where $\mathbf{a} \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^n\right)$ and $e \leftarrow D_{\mathbb{Z},\alpha q}$.

We will write $D_{n,q,\alpha}(\mathbf{s})$ the given distribution.

A closer look at LWE

More formally, we fix $n \geqslant 1$, $q \geqslant 2$ and $\alpha \in]0, 1[$.

Given $\mathbf{s} = [s_1, \ldots, s_n] \in (\mathbb{Z}/q\mathbb{Z})^n$, we define a LWE sample to be :

$$(\mathbf{a}, (\mathbf{a} \mid \mathbf{s}) + e)\,,$$

where $\mathbf{a} \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^n\right)$ and $e \leftarrow D_{\mathbb{Z},\alpha q}$.

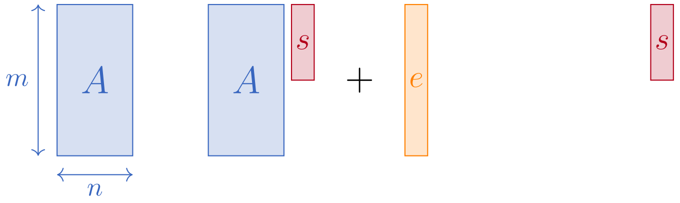
We will write $D_{n,q,\alpha}(\mathbf{s})$ the given distribution.

The LWE$^n_{\alpha,q}$ problem then is :

Given m samples of $D_{n,q,\alpha}(\mathbf{s})$, retrieve $\mathbf{s}$.

A closer look at LWE

Given find



- $A \leftarrow U\left(M_{m,n}(\mathbb{Z}/q\mathbb{Z})\right)$
- $s \leftarrow U\left((\mathbb{Z}/q\mathbb{Z})^n\right)$
- $e \leftarrow D_{\mathbb{Z}^m,\alpha q}$ **short**

- One can vary the distributions.
- Still active area of research.

Lattice-based cryptography

Structured variants of LWE

Lattice-based cryptography

Structured variants of LWE

Problem of efficiency ⟶ Structured variants { Ring-LWE, Module-LWE }

Ring-LWE

Fix $q \in \mathbb{N}$, K a number field, $R_q = \mathcal{O}_K/(q)$

A Ring-LWE sample is $(a, b = as + e)$,
where $a \overset{\mathcal{U}}{\leftarrow} R_q$, $s \overset{\mathcal{D}_s}{\leftarrow} R_q, e \overset{\mathcal{D}_e}{\leftarrow} R$

`Goal:` Find s

Think $K = \mathbb{Q}[X]/(X^n + 1)$
and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$
for $n = 2^k$.

Idea : Replace $\mathbb{Z}^n$ by a polynomial ring !

Fix $q \in \mathbb{N}$, K a number field, $R_q = \mathcal{O}_K/(q)$.

Think $K = \mathbb{Q}[X]/(X^n + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$.

$a \in K$ can be represented by the matrix of its action by left multiplication :

$$[a] : s \mapsto a \cdot s.$$

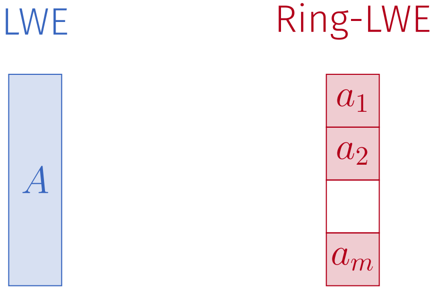
Idea : Replace $\mathbb{Z}^n$ by a polynomial ring !

Fix $q \in \mathbb{N}$, K a number field, $R_q = \mathcal{O}_K/(q)$.

Think $K = \mathbb{Q}[X]/(X^n + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$.

$a \in K$ can be represented by the matrix of its action by left multiplication :

$$[a] : s \mapsto a \cdot s.$$



Idea : Replace $\mathbb{Z}$ by a polynomial ring !

Fix $q \in \mathbb{N}$, K a number field, $R_q = \mathcal{O}_K/(q)$.

Think $K = \mathbb{Q}[X]/(X^n + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$.

$a \in K$ can be represented by the matrix of its action by left multiplication :

$$[a] : s \mapsto a \cdot s.$$

Lattice-based cryptography

Module-LWE

Idea : Replace $\mathbb{Z}$ by a polynomial ring !

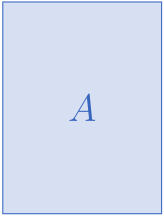
Fix $q \in \mathbb{N}$, K a number field, $R_q = \mathcal{O}_K/(q)$.

Think $K = \mathbb{Q}[X]/(X^n + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$.

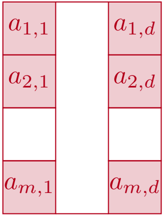
$a \in K$ can be represented by the matrix of its action by left multiplication :

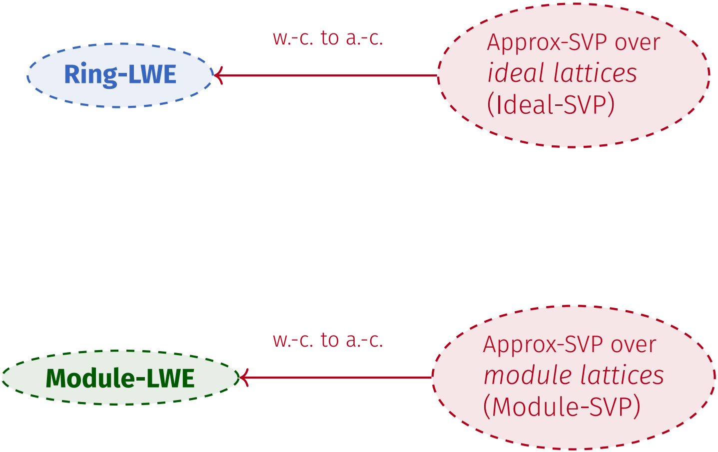
$$[a] : s \mapsto a \cdot s.$$

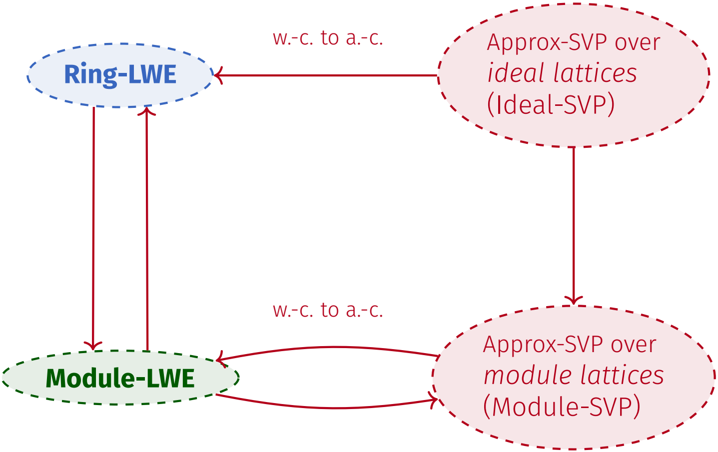
LWE



Module-LWE




w.-c. to a.-c.
Ring-LWE
Approx-SVP over
ideal lattices
(Ideal-SVP)
w.-c. to a.-c.
Module-LWE
Approx-SVP over
module lattices
(Module-SVP)



w.-c. to a.-c.
Approx-SVP over
ideal lattices
(Ideal-SVP)
Ring-LWE
w.-c. to a.-c.
Module-LWE
Approx-SVP over
module lattices
(Module-SVP)

Some definitions

Number field $K \cong \mathbb{Q}[X]/(P(X))$

$g \in K \iff$ pol. with rational coeffs

$g \in K \iff (g_0, \dots, g_{n-1}) \in \mathbb{Q}^n$

θ root of $P(X) \leftrightarrow \sigma$ complex embedding

Minkowski (or canonical) embedding :
$\sigma_K : g \in K \mapsto (\sigma(g))_\sigma = (g(\theta))_\theta$

$\mathbb{Q}(\zeta_8) \cong \mathbb{Q}[X]/(X^4 + 1)$

$g = 1/2 + X + 3X^2 - 2X^3, g_i \in \mathbb{Q}$

$(1/2, 1, 3, -2) \in \mathbb{Q}^4$

$g \mapsto g(\zeta_8) = 1/2 + \zeta_8 + 3\zeta_8^2 - 2\zeta_8^3$

$g \mapsto g(\zeta_8^3) = 1/2 + \zeta_8^3 + 3\zeta_8^6 - 2\zeta_8^9$

Some definitions

Ring of integers $\mathcal{O}_K \sim \mathbb{Z}[X]/(P(X))$

(Not true in general)

$g \in \mathcal{O}_K \iff$ pol. with integral coeffs

$$g \in \mathcal{O}_K \iff (g_0, \ldots, g_{n-1}) \in \mathbb{Z}^n$$

$$\mathbb{Z}(\zeta_8) \cong \mathbb{Z}[X]/(X^4 + 1)$$

$$g = 1 + X + 3X^2 - 2X^3, g_i \in \mathbb{Z}$$

$$(1, 1, 3, -2) \in \mathbb{Z}^4$$

Some definitions

Ring of integers $\mathcal{O}_K \sim \mathbb{Z}[X]/(P(X))$
(Not true in general)

$g \in \mathcal{O}_K \iff$ pol. with integral coeffs

$$g \in \mathcal{O}_K \iff (g_0, \dots, g_{n-1}) \in \mathbb{Z}^n$$

$$\mathbb{Z}(\zeta_8) \cong \mathbb{Z}[X]/(X^4+1)$$

$$g = 1 + X + 3X^2 - 2X^3, g_i \in \mathbb{Z}$$

$$(1, 1, 3, -2) \in \mathbb{Z}^4$$

Ideal $I = (g, h) = g\mathcal{O}_K + h\mathcal{O}_K$
Principal ideal $I = (g) = g\mathcal{O}_K$

Ideal lattice : generated by

coeffs of $gX^i, hX^j, i, j \in [\![1, n]\!]$

or

$$\left(\sigma_K(gX^i)\right)_i, \left(\sigma_K(hX^j)\right)_j$$

$$\begin{bmatrix} 1 & 1 & 3 & -2 \\ 2 & 1 & 1 & 3 \\ -3 & 2 & 1 & 1 \\ -1 & -3 & 2 & 1 \end{bmatrix} \begin{matrix} \leftarrow g \\ \leftarrow gX \\ \leftarrow gX^2 \\ \leftarrow gX^3 \end{matrix}$$

Some definitions

Ring of integers $\mathcal{O}_K \sim \mathbb{Z}[X]/(P(X))$
(Not true in general)

$g \in \mathcal{O}_K \iff$ pol. with integral coeffs

$$g \in \mathcal{O}_K \iff (g_0, \ldots, g_{n-1}) \in \mathbb{Z}^n$$

$$\mathbb{Z}(\zeta_8) \cong \mathbb{Z}[X]/(X^4+1)$$

$$g = 1 + X + 3X^2 - 2X^3, g_i \in \mathbb{Z}$$

$$(1, 1, 3, -2) \in \mathbb{Z}^4$$

Ideal $I = (g, h) = g\mathcal{O}_K + h\mathcal{O}_K$
Principal ideal $I = (g) = g\mathcal{O}_K$

Ideal lattice : generated by

coeffs of $gX^i, hX^j, i, j \in [\![1, n]\!]$

or

$$\left(\sigma_K(gX^i)\right)_i, \left(\sigma_K(hX^j)\right)_j$$

$$\begin{bmatrix} 1 & 1 & 3 & -2 \\ 2 & 1 & 1 & 3 \\ -3 & 2 & 1 & 1 \\ -1 & -3 & 2 & 1 \end{bmatrix} \begin{matrix} \leftarrow g \\ \leftarrow gX \\ \leftarrow gX^2 \\ \leftarrow gX^3 \end{matrix}$$

Polynomial structure $\implies$ efficient for storage and computations

Overview of the situation

SVP_γ is hard over general lattices

Overview of the situation

SVP$_\gamma$ is hard over general lattices

↓ worst-case to average-case

LWE is a good average-case problem

Overview of the situation

SVP$_\gamma$ is hard over general lattices

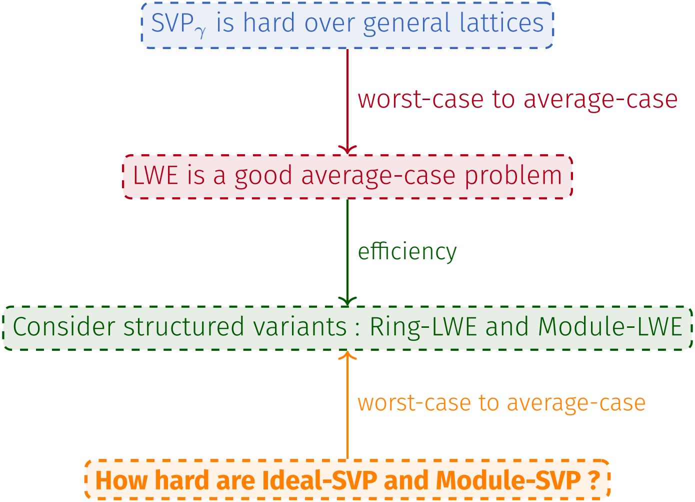
↓ worst-case to average-case

LWE is a good average-case problem

↓ efficiency

Consider structured variants : Ring-LWE and Module-LWE

Overview of the situation



Approx-SVP over ideal lattices

SVP over principal ideals

Consider an intermediate problem.

Short Generator Principal Ideal Problem (SG-PIP):
Given a principal ideal $I = (g)$ such that g is short, retrieve g.

[2] $\mathrm{Log}_K : x \mapsto (\ln |\sigma_1(x)|, \ldots, \ln |\sigma_n(x)|)$

SVP over principal ideals

Consider an intermediate problem.

Short Generator Principal Ideal Problem (SG-PIP):
Given a principal ideal $I = (g)$ such that g is short, retrieve g.

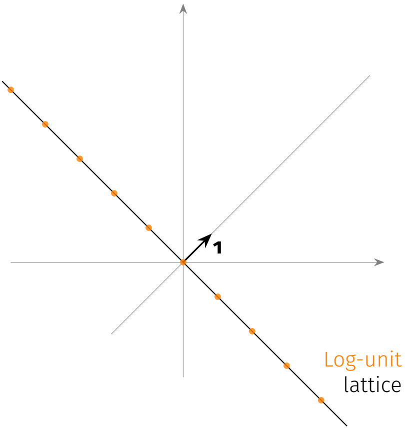
1. Find a generator $h = gu$ of I $(u \in \mathcal{O}_K^\times)$
   Can be done in polynomial time with a quantum computer

2. Find g given h.

Use the Log-embedding[2] and the Log-unit lattice $\mathrm{Log}(\mathcal{O}_K^\times)$

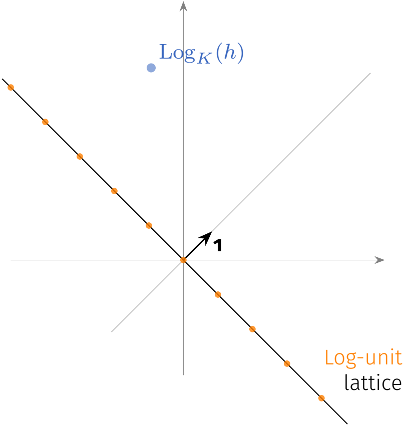
[2] $\mathrm{Log}_K : x \mapsto (\ln|\sigma_1(x)|, \ldots, \ln|\sigma_n(x)|)$

Artistic (?) view of the algorithm[3]



[3] Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Artistic (?) view of the algorithm[3]



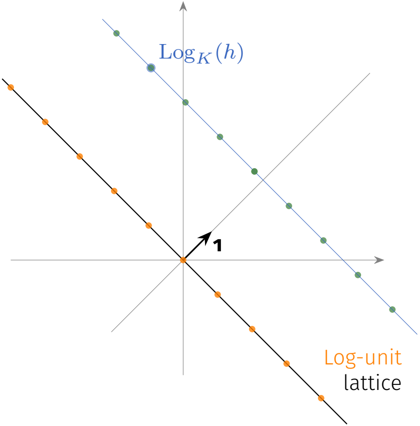
Let I be a challenge ideal.

1. Quantum decomposition
   Apply Log_K
   $\mathrm{Log}_K(h) = \mathrm{Log}_K(g) + \mathrm{Log}_K(u) \in \mathrm{Log}_K(g) + \mathrm{Log}_K(\mathcal{O}_K^\times)$

$$h = g \cdot u$$

[3]Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Artistic (?) view of the algorithm[3]



Let I be a challenge ideal.

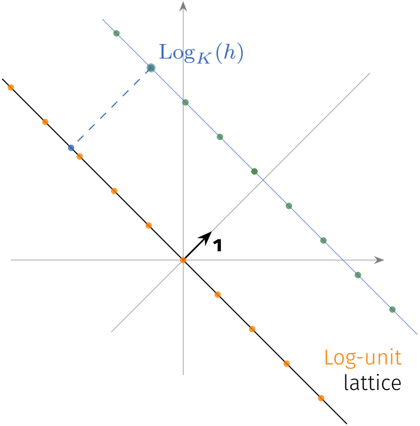
1. Quantum decomposition
   Apply Log_K
   $\mathrm{Log}_K(h) = \mathrm{Log}_K(g) + \mathrm{Log}_K(u) \in \mathrm{Log}_K(g) + \mathrm{Log}_K(\mathcal{O}_K^\times)$

2. *Short* coset representative ?

$$h = g \cdot u$$

[3]Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Artistic (?) view of the algorithm[3]



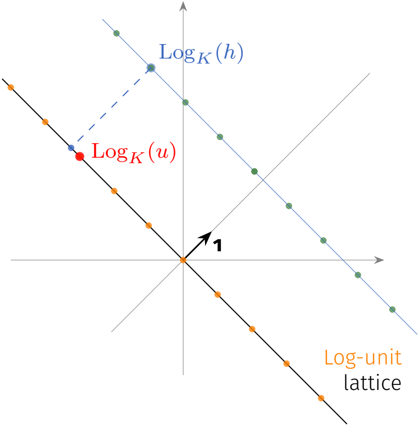
Let I be a challenge ideal.

1. Quantum decomposition
   Apply Log_K
   $\mathrm{Log}_K(h) = \mathrm{Log}_K(g) + \mathrm{Log}_K(u) \in \mathrm{Log}_K(g) + \mathrm{Log}_K(\mathcal{O}_K^\times)$
2. *Short* coset representative ?

$$h = g \cdot u$$

[3] Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Artistic (?) view of the algorithm[3]



Let I be a challenge ideal.

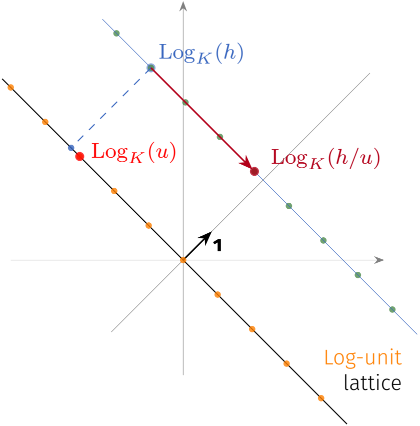
1. Quantum decomposition
   Apply Log_K
   $\mathrm{Log}_K(h) = \mathrm{Log}_K(g) + \mathrm{Log}_K(u) \in \mathrm{Log}_K(g) + \mathrm{Log}_K(\mathcal{O}_K^\times)$

2. *Short* coset representative ?

$$h = g \cdot u$$

[3] Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Artistic (?) view of the algorithm[3]



Let I be a challenge ideal.

1. Quantum decomposition
   Apply Log_K
   $\mathrm{Log}_K(h) = \mathrm{Log}_K(g) + \mathrm{Log}_K(u) \in \mathrm{Log}_K(g) + \mathrm{Log}_K(\mathcal{O}_K^\times)$

2. *Short* coset representative ?

3. Hope this is *short* in I.

$$h = g \cdot u$$
$$(h/u) = g$$

[3]Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Existing works

- [Cra+16] quantum polynomial-time or classical $2^{n^{2/3+\epsilon}}$-time algorithm to solve SG-PIP over cyclotomic fields

- [Bau+17] efficient classical algorithm to solve SG-PIP over multiquadratic fields. Good results in practice.
  $\rightarrow$ of the form $\mathbb{Q}(\sqrt{m_1}, \ldots, \sqrt{m_r})$

Existing works

- [Cra+16] quantum polynomial-time or classical $2^{n^{2/3+\epsilon}}$-time algorithm to solve SG-PIP over cyclotomic fields
- [Bau+17] efficient classical algorithm to solve SG-PIP over multiquadratic fields. Good results in practice.
  $\rightarrow$ of the form $\mathbb{Q}(\sqrt{m_1}, \ldots, \sqrt{m_r})$
- [LPS20] Extend results of [Bau+17] to multicubic fields
  $\rightarrow$ of the form $\mathbb{Q}(\sqrt[3]{m_1}, \ldots, \sqrt[3]{m_r})$
- [LPS21] General real Kummer extensions
  $\rightarrow$ of the form $\mathbb{Q}(\sqrt[p]{m_1}, \ldots, \sqrt[p]{m_r})$
  $\rightarrow$ fields of the form $\mathbb{Q}(\sqrt[p]{2}, \sqrt[p]{3})$ seem to be more resistant

SVP of general ideals

General algorithms

Consider K a number field, I an ideal.

Fix S a set of prime ideals *(generating the class group.)*

[4] $\mathrm{Log}_{K,S} : x \mapsto (\ln|\sigma_1(x)|, \dots, \ln|\sigma_n(x)|, -v_{\mathfrak{p}_1}(x) \ln \mathrm{N}_{K/\mathbb{Q}}(\mathfrak{p}_1), \dots, -v_{\mathfrak{p}_r}(x) \ln \mathrm{N}_{\mathbf{K}/\mathbb{Q}}(\mathfrak{p}_\mathbf{r}))$

Consider K a number field, I an ideal.

Fix S a set of prime ideals *(generating the class group.)*

1. Compute a S-generator of I, i.e. h s.t. $(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v_\mathfrak{p}}$
2. Reduce h, i.e. find $s \in \mathcal{O}_{K,S}^\times$ such that h/s is short.

[4] $\mathrm{Log}_{K,S} : x \mapsto (\ln|\sigma_1(x)|, \ldots, \ln|\sigma_n(x)|, -v_{\mathfrak{p}_1}(x) \ln \mathrm{N}_{K/\mathbb{Q}}(\mathfrak{p}_1), \ldots, -v_{\mathfrak{p}_r}(x) \ln \mathrm{N}_{\mathbf{K}/\mathbb{Q}}(\mathfrak{p}_\mathbf{r}))$

SVP of general ideals

General algorithms

Consider K a number field, I an ideal.

Fix S a set of prime ideals *(generating the class group.)*

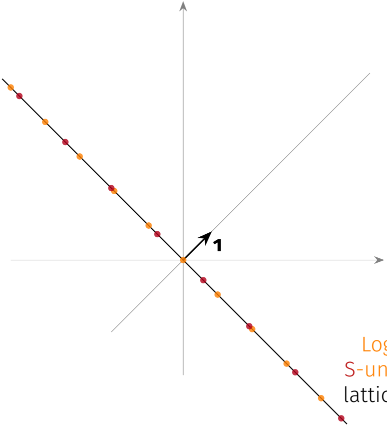
1. Compute a S-generator of I, i.e. h s.t. $(h) = I \cdot \prod_{\mathfrak{p}\in S} \mathfrak{p}^{v_\mathfrak{p}}$
2. Reduce h, i.e. find $s \in \mathcal{O}_{K,S}^\times$ such that h/s is short.

Two variants for step 2.

1. First reduce $\prod_\mathfrak{p} \mathfrak{p}^{v_\mathfrak{p}}$; then find a generator with the Log-embedding.
   - $\rightarrow$ [CDW17] cyclotomic fields, subexponential approximation factor
2. Use the Log-S-embedding [4] to reduce everything.
   - $\rightarrow$ [PHS19] all number fields, exponential preprocessing, subexponential approximation factor
   - $\rightarrow$ [BR20] other def. of $\mathrm{Log}_{K,S}$, same asymptotic results, **good results in practice for cyclotomics up to dimensions 70.**

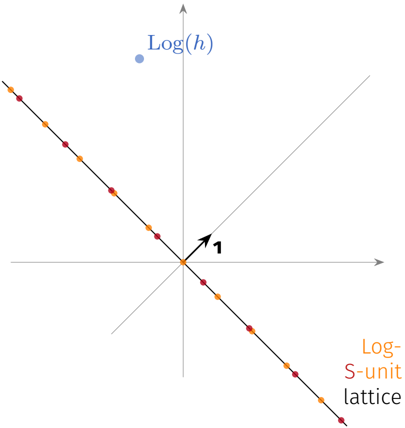
[4] $\mathrm{Log}_{K,S} : x \mapsto (\ln|\sigma_1(x)|, \ldots, \ln|\sigma_n(x)|, -v_{\mathfrak{p}_1}(x)\ln \mathrm{N}_{K/\mathbb{Q}}(\mathfrak{p}_1), \ldots, -v_{\mathfrak{p}_r}(x)\ln \mathrm{N}_{\mathbf{K}/\mathbb{Q}}(\mathfrak{p}_\mathbf{r}))$

View of an S-unit algorithm (Twisted-PHS)[5]



[5] Thanks to O. Bernard for the slide (particularly the `tikz` picture)

View of an S-unit algorithm (Twisted-PHS)[5]



Let I be a challenge ideal.

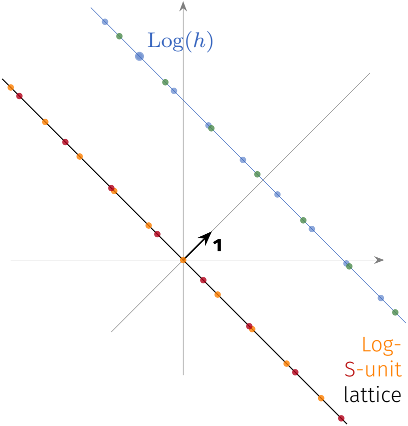
1. Quantum decomposition output

   Apply Log

$$(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v}$$

[5] Thanks to O. Bernard for the slide (particularly the **tikz** picture)

View of an S-unit algorithm (Twisted-PHS)[5]



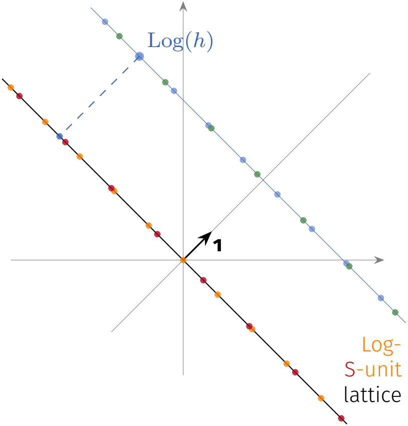
Let I be a challenge ideal.

1. Quantum decomposition output
   Apply Log
2. *Short* coset representative ?

$$(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v}$$

[5] Thanks to O. Bernard for the slide (particularly the `tikz` picture)

View of an S-unit algorithm (Twisted-PHS)[5]



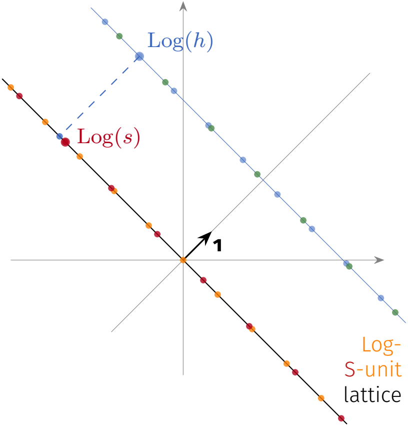
Let I be a challenge ideal.

1. Quantum decomposition output
   Apply Log
2. *Short* coset representative ?

$$(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v}$$

[5]Thanks to O. Bernard for the slide (particularly the `tikz` picture)

View of an S-unit algorithm (Twisted-PHS)[5]



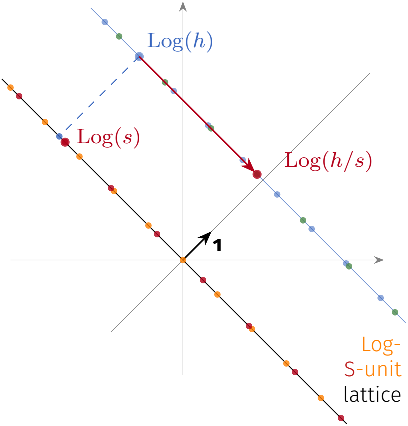
Let I be a challenge ideal.

1. Quantum decomposition output
   Apply Log
2. *Short* coset representative ?

$$(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v}$$
$$(s) = \prod_{\mathfrak{p} \in S} \mathfrak{p}^{w}$$

[5] Thanks to O. Bernard for the slide (particularly the **tikz** picture)

View of an S-unit algorithm (Twisted-PHS)[5]



Let I be a challenge ideal.

1. Quantum decomposition output
   Apply Log
2. *Short* coset representative ?
3. Hope this is *short* in I.

$$
\begin{aligned}
(h) &= I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v} \\
(s) &= \prod_{\mathfrak{p} \in S} \mathfrak{p}^{w} \\
\hline
(h/s) &= I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v-w}
\end{aligned}
$$

[5] Thanks to O. Bernard for the slide (particularly the `tikz` picture)

Can we extend these good results to higher dimensions ?

Two major obstructions for experiments :

- Decomposition $(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v_{\mathfrak{p}}}$
- Group of S-units $(s) = \prod_{S \in S} \mathfrak{p}^{e_{\mathfrak{p}}}$

Can we extend these good results to higher dimensions ?

Two major obstructions for experiments :

- Decomposition $(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v_{\mathfrak{p}}}$
- Group of S-units $(s) = \prod_{S \in S} \mathfrak{p}^{e_{\mathfrak{p}}}$

Use new results of Bernard and Kučera (2021) on Stickelberger ideal

- Obtain explicit short basis of S_m
- It is constructive : the associated generators can be computed efficiently
- Free family of short S-units

Can we extend these good results to higher dimensions ?

Two major obstructions for experiments :

- Decomposition $(h) = I \cdot \prod_{\mathfrak{p} \in S} \mathfrak{p}^{v_\mathfrak{p}}$
- Group of S-units $(s) = \prod_{S \in S} \mathfrak{p}^{e_\mathfrak{p}}$

Use new results of Bernard and Kučera (2021) on Stickelberger ideal

- Obtain explicit short basis of S_m
- It is constructive : the associated generators can be computed efficiently
- Free family of short S-units

Allows us to *approximate* $\mathrm{Log}(\mathcal{O}_{K,S}^{\times})$ with a full-rank sublattice

- Cyclotomic units
- Explicit Stickelberger generators
- Real $S \cap K_m^+$-units $\rightarrow$ only part sub-exponential ; dimension $n/2$
- 2-saturation to reduce the index

Experimental results[6]

Cyclotomic fields with almost all conductors, up to dimension 210.

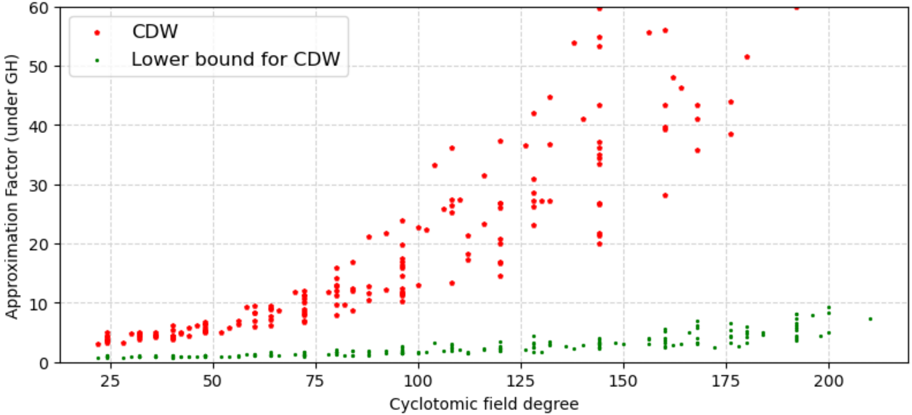
- Simulated targets in the Log-space.
- Randomised drift strategy.

[6] Code available at `https://github.com/ob3rnard/Tw-Sti`.

Experimental results[6]

Cyclotomic fields with almost all conductors, up to dimension 210.

- Simulated targets in the Log-space.
- Randomised drift strategy.

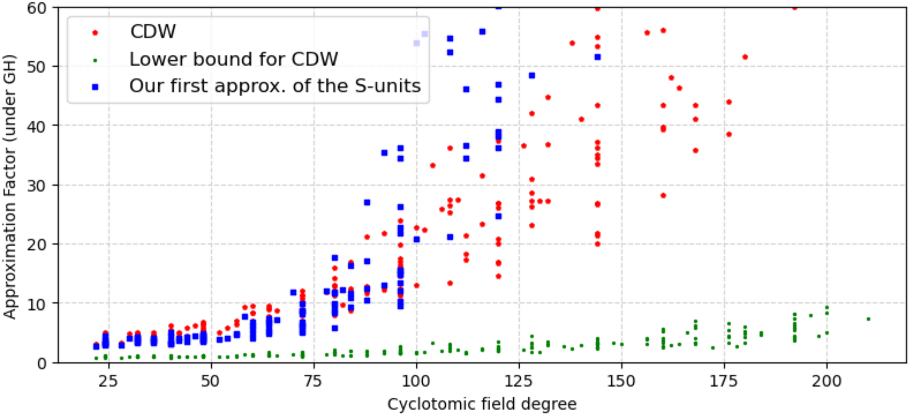


[6] Code available at `https://github.com/ob3rnard/Tw-Sti`.

Experimental results[6]

Cyclotomic fields with almost all conductors, up to dimension 210.

- Simulated targets in the Log-space.
- Randomised drift strategy.

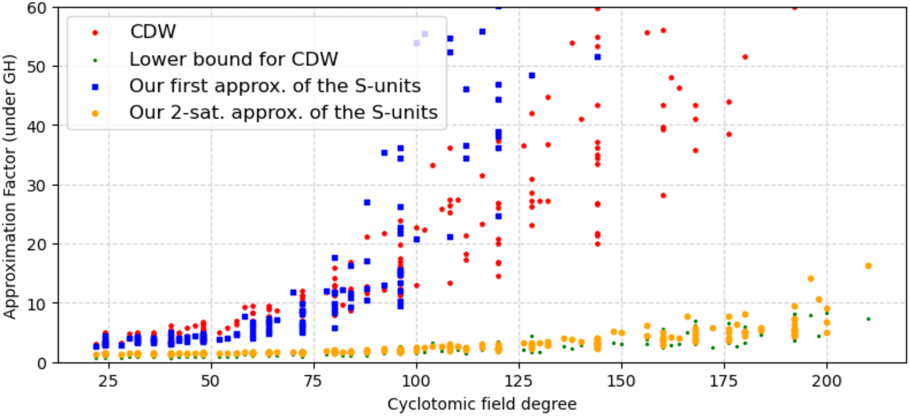


[6] Code available at `https://github.com/ob3rnard/Tw-Sti`.

Experimental results[6]

Cyclotomic fields with almost all conductors, up to dimension 210.

- Simulated targets in the Log-space.
- Randomised drift strategy.

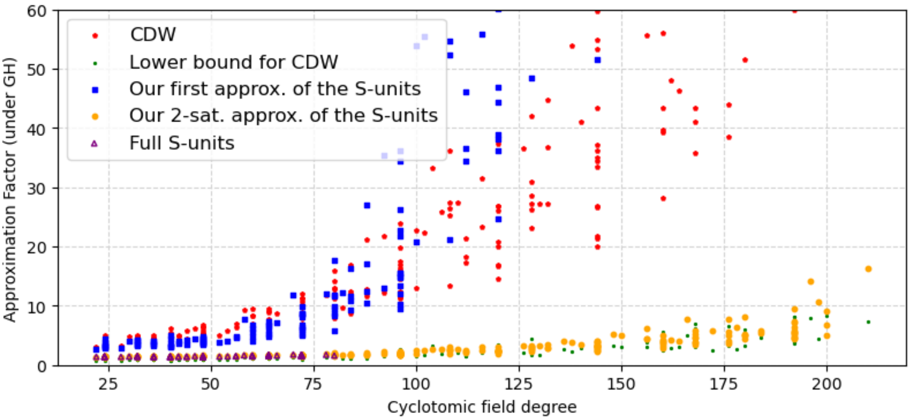


[6] Code available at `https://github.com/ob3rnard/Tw-Sti`.

Experimental results[6]

Cyclotomic fields with almost all conductors, up to dimension 210.

- Simulated targets in the Log-space.
- Randomised drift strategy.



[6] Code available at `https://github.com/ob3rnard/Tw-Sti`.

Conclusion

1. Upper-bounds on approx. factors reached by S-unit algorithms up to degree 210.
2. Twisted-PHS more efficient than CDW. (with simple CVP/BDD solver)
3. Twisted-PHS comparable to volumetic lower bound shown in [DPW19].

Conclusion

1. Upper-bounds on approx. factors reached by S-unit algorithms up to degree 210.

2. Twisted-PHS more efficient than CDW. (with simple CVP/BDD solver)

3. Twisted-PHS comparable to volumetic lower bound shown in [DPW19].

What does it mean for lattice-based cryptography ?

1. One should consider PHS / Twisted-PHS to evaluate the security of Ideal-SVP. $\rightarrow$ crossover point around $n = 7000$ in [DPW19], should be lower

2. Results not reassuring nor devastating.

3. Lattice-based crypto is safe (for now) : recall that it is based on Ring-LWE or Module-LWE.

What's next

1. Reduce the gap with Log-S-unit lattice.
   - → requires big p-saturation
   - → In the works ! (Generalisation of Couveignes' and Thomé's algorithms for square-roots [BFL23])
2. Consider other number fields (Kummer for example).
3. Study the geometrical structure of the Log-S-unit lattice.
4. Work on other specific algorithms (basis reduction, enumeration)
   - → e.g. effective Module-LLL

Thank you for your attention

References I

[Bau+17] Jens Bauch et al. "Short Generators Without Quantum Computers: The Case of Multiquadratics". In: *Advances in Cryptology – EUROCRYPT 2017*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Cham: Springer International Publishing, 2017, pp. 27–59. ISBN: 978-3-319-56620-7.

[BFL23] Olivier Bernard, Pierre-Alain Fouque, and Andrea Lesavourey. *Computing e-th roots in number fields*. 2023. arXiv: 2305.17425 [math.NT].

[BR20] Olivier Bernard and Adeline Roux-Langlois. "Twisted-PHS: Using the Product Formula to Solve Approx-SVP in Ideal Lattices". In: *Advances in Cryptology – ASIACRYPT 2020*. Ed. by Shiho Moriai and Huaxiong Wang. Cham: Springer International Publishing, 2020, pp. 349–380. ISBN: 978-3-030-64834-3.

[CDW17] R. Cramer, L. Ducas, and B. Wesolowski. "Short Stickelberger Class Relations and Application to Ideal-SVP". In: *EUROCRYPT*. 2017.

References II

[Cra+16] Ronald Cramer et al. “Recovering Short Generators of Principal Ideals in Cyclotomic Rings”. In: *Advances in Cryptology – EUROCRYPT 2016*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 559–585. ISBN: 978-3-662-49896-5.

[DPW19] Léo Ducas, Maxime Plançon, and Benjamin Wesolowski. “On the Shortness of Vectors to Be Found by the Ideal-SVP Quantum Algorithm”. In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 322–351. ISBN: 978-3-030-26948-7.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. “Trapdoors for hard lattices and new cryptographic constructions”. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. Victoria, British Columbia, Canada: Association for Computing Machinery, 2008, pp. 197–206. ISBN: 9781605580470. DOI: 10.1145/1374376.1374407. URL: https://doi.org/10.1145/1374376.1374407.

[LPS20] Andrea Lesavourey, Thomas Plantard, and Willy Susilo. "Short Principal Ideal Problem in multicubic fields". In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 359–392. DOI: https://doi.org/10.1515/jmc-2019-0028. URL: https://www.degruyter.com/view/journals/jmc/14/1/article-p359.xml.

[LPS21] Andrea Lesavourey, Thomas Plantard, and Willy Susilo. *On the Short Principal Ideal Problem over some real Kummer fields*. Cryptology ePrint Archive, Report 2021/1623. https://ia.cr/2021/1623. 2021.

[PHS19] Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. "Approx-SVP in Ideal Lattices with Pre-processing". In: *Advances in Cryptology – EUROCRYPT 2019*. Ed. by Yuval Ishai and Vincent Rijmen. Cham: Springer International Publishing, 2019, pp. 685–716. ISBN: 978-3-030-17656-3.