Course 1 : lattices

Course 2 : lattice problems and algorithms

Course 3 : cryptographic problems

Course 4 : constructing cryptographic primitives
+ a little bit of algebraic lattices

Lattice-based Cryptography

I) Lattices :

1) Definitions :

Definition : Let $m \geq n$ and $b_1, \dots, b_n \in \mathbb{R}^m$ be linearly independent vectors. The lattice spanned by the $(b_i)_{1 \leq i \leq n}$ is

$$L(b_1, \dots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i \mid \forall i, x_i \in \mathbb{Z} \right\}$$

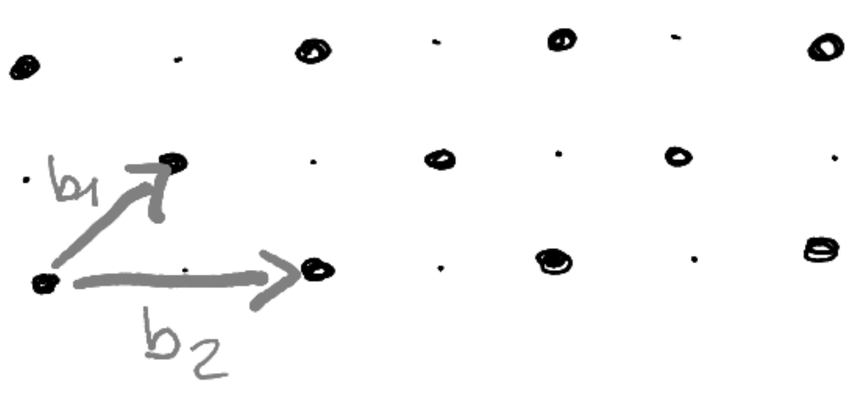
- If $m = n$, we say that the lattice has full rank.
- The vectors $(b_1, \dots, b_n)$ are called a basis of L

Remark : we use column vector notations

Example : $b_i = e_i$, the standard basis of $\mathbb{R}^n$ (one "1" at position i, and "0" everywhere else)

$\rightsquigarrow L = \mathbb{Z}^n$

- $m = n = 2$, $b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $b_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$



Alternative definition : A lattice is a subset $L \subseteq \mathbb{R}^m$ s.t. :

* L is an additive group ($v - v' \in L \quad \forall v, v' \in L$)

* L is discrete for the Euclidean metric

($\forall v \in L, \exists \varepsilon > 0$ s.t. the Euclidean ball centered at v and with radius ε contains no point of L except for v).

Proof : admitted (that both defs are equivalent)

More examples : Let $m \geq r > 0$, $q \geq 0$ integer.

Let $\boxed{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times r}$

- The image lattice associated to A is

$$\Lambda(A) = \{ y \in \mathbb{Z}^m \mid \exists s \in (\mathbb{Z}/q\mathbb{Z})^r, \; y = \boxed{A}^T s \bmod q \}$$

- The kernel lattice associated to A is

$$\Lambda^\perp(A) = \{ x \in \mathbb{Z}^m \mid x^T \boxed{A} = 0 \bmod q \}$$

Proof that these are lattices: One can check that both $\Lambda(A)$ and $\Lambda^\perp(A)$ are stable by addition and subtraction. They are also discrete because they are $\subseteq \mathbb{Z}^m$. So they are lattices using the alternative definition.

2) Properties :

Definition: Let $Span(L)$ be the real vector space spanned by the vectors of L (i.e., $Span(L) = \{ \sum_{i=1}^{s} x_i v_i \mid s \geq 1, x_i \in \mathbb{R}, v_i \in L \}$). We call $\dim(Span(L))$ the **rank** of L, and we write it $rk(L)$.

Lemma: Any basis $b_1, \dots, b_n$ of L has cardinality $n = rk(L)$.

Proof:
Let $H := Span(L)$ (so that $rk(L) = \dim(H)$).
Since $b_1, \dots, b_n \in H$, and they are linearly indep, then $\dim(H) \geq n$. Moreover, since every $v \in L$ can be written $v = \sum v_i b_i$ with $v_i \in \mathbb{Z}$, then H is actually equal to the real vector space spanned by the b_i's, so it has dimension exactly n. In other words $n = \dim(H) = rk(L)$

Exercise: Compute $rk(\Lambda(A))$ and $rk(\Lambda^{\perp}(A))$.

Solution: $rk(\Lambda(A)) = m$. Indeed, $\Lambda(A) \subseteq \mathbb{Z}^m \subseteq \mathbb{R}^m$
So $Span(\Lambda(A)) \subseteq \mathbb{R}^m$ and $rk(A) \leq m$.
But $q \times e_i = (0, \dots, 0, q, 0, \dots, 0) \in \Lambda(A)$ for all $1 \leq i \leq m$ and
position i
they span $\mathbb{R}^m$ as a vector space. So $Span(\Lambda(A)) = \mathbb{R}^m$
and $rk(\Lambda(A)) = m$.
Similarly, $Span(\Lambda^{\perp}(A)) = \mathbb{R}^m$ so $rk(\Lambda^{\perp}(A)) = m$.

From now on, we consider mostly full rank lattices, i.e., lattices $L \subseteq \mathbb{R}^n$ with $rk(L) = n$.
$\Lambda(A)$ and $\Lambda^\perp(A)$ are full rank lattices. (the columns of B and C are the basis vectors)

Lemma: Let B and $C \in \mathbb{R}^{n \times n}$ be two bases of the same (full rank) lattice L. Then $B = C \times U$ where $U \in \mathbb{Z}^{n \times n}$ is invertible and $U^{-1} \in \mathbb{Z}^{n \times n}$ too (equivalently, $U \in \mathbb{Z}^{n \times n}$ and $\det(U) = \pm 1$, we say that such U is unimodular)

Proof: exercise.
Let $b_1, \dots, b_n$ be the columns of B and $c_1, \dots, c_n$ the columns of C. Since $b_i \in L$ and C basis of L, $\exists\, x_1^{(i)}, \dots, x_n^{(i)} \in \mathbb{Z}$ s.t. $b_i = \sum_{j=1}^{n} x_j^{(i)} c_j = C \times \begin{pmatrix} x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix}$. So $B = C \times X$ for some $X \in \mathbb{Z}^{n \times n}$. Similarly, $C = B \times Y$ for some $Y \in \mathbb{Z}^{n \times n}$.
Hence, $B = CX = BYX$. Since B is invertible, $YX = I_n$.
So X in invertible and its inverse is $Y \in \mathbb{Z}^{n \times n}$, as desired.

Lemma: Let $L \subseteq \mathbb{R}^n$ be a full rank lattice. Then for any two bases B, C of L, we have $|\det(B)| = |\det(C)|$.
This quantity is called the volume of the lattice, and denoted $vol(L)$ (or $\det(L)$).

Proof: From the previous lemma, we have $B = C \times U$ with $\det(U) = \pm 1$. Then $|\det(B)| = |\det(C)| \times |\det(U)|$
$= |\det(C)|$.

Example: Let $m \geq r > 0$, $q \geq 0$ prime.
Let $A \in (\mathbb{Z}/q\mathbb{Z})^{m \times r}$ be of rank r.

Then $\det(\Lambda(A)) = q^{m-r}$.

and $\det(\Lambda^{\perp}(A)) = q^{r}$

For a proof, see exercise 3.

Definition: Let L be a lattice of rank n.
The first minimum of L is

$$\lambda_1(L) := \min\{\|v\|_2 \mid v \in L \setminus \{0\}\}$$

More generally, for $i \in \{1, \dots, n\}$, the i-th minimum of L is

$$\lambda_i(L) := \min\{r > 0 \mid \exists v_1, \dots, v_i \in L \text{ linearly independant s.t. } \|v_i\|_2 \leq r\}$$

Remark : A shortest non zero vector of a lattice L is never unique: if $v \in L$ is s.t. $\|v\|_2 = \lambda_1(L)$, then we also have $-v \in L$ and $\|-v\|_2 = \lambda_1(L)$.
In general, there may be many vectors of L reaching $\lambda_1(L)$.

Minkowski's first theorem : If L has rank n, then

$$\lambda_1(L) \leq \sqrt{n}\, \det(L)^{1/n}$$

Remark : computing the exact value of $\lambda_1(L)$ is usually a hard problem (see later).

Proof : admitted

3) Algorithmic problems :

Quizz : which ones of the following problems are hard (= no poly time algorithm known) or easy (known poly time algo): (input : a basis of some L)

(1) Testing equality of lattices ✓ (easy)
(2) Testing inclusion of lattices ✓
(3) Intersecting lattices ✓
(4) Computing a shortest vector ✗ (hard)
(5) Computing a closest vector $v \in L$ to some target $t \in \mathrm{Span}_{\mathbb{R}}(L)$ ✗

Exercise : how do we test if $L_1 \subseteq L_2$?

Solution : B_1 basis of L_1, B_2 basis of L_2.

If $L_1 \subseteq L_2$, every column vector $b_i^{(1)}$ of B_1 is in L_2,

i.e. $b_i^{(1)} = B_2 \times x_i$ with $x_i \in \mathbb{Z}^n$

So $B_1 = B_2 \times X$ with $X \in \mathbb{Z}^{n \times n}$.

Inversely, if $B_1 = B_2 X$ with $X \in \mathbb{Z}^{n \times n}$, then every column of B_1 is in L_2, and so $L_1 \subseteq L_2$.

The algorithm is then: compute $B_2^{-1} B_1$ and test if its coefficients are integers.

Let's focus now on the hard lattice problems

Definition (BDD): Let $\gamma \geq 1$. The γ-bounded distance decoding problem (BDD_γ) is as follows: given as input a basis B of a lattice L, and a vector $t \in \text{Span}(L)$ s.t. $\text{dist}(t, L) \leq \frac{1}{\gamma} \times \lambda_1(L)$, the goal is to find $v \in L$ s.t. $\|t - v\|_2 \leq 1/\gamma \cdot \lambda_1(L)$.

Remark :
- Span (L) is the real span of the vectors of L
- $\text{dist}(t, L) = \min_{v \in L} \|t - v\|_2$

Lemma: if $\gamma > 2$, then the solution to BDD_γ is unique.

Proof: Assume by contradiction that we have $v_1 \neq v_2 \in L$ s.t.

$$\|t - v_1\| \leq \frac{1}{\gamma}\lambda_1(L)$$

and $\|t - v_2\| \leq \frac{1}{\gamma}\lambda_1(L)$

Then $w = v_1 - v_2$ is a non-zero vector of L, and by triangular inequality, we have

$$\|v_1 - v_2\| \leq 2/\gamma\, \lambda_1(L) < \lambda_1(L) \quad (\text{since } \gamma > 2).$$

This is a contradiction (by definition of $\lambda_1(L)$).

Definition (SVP): Let $\gamma \geq 1$. The γ-shortest vector problem (SVP_γ) is as follows: given as input a basis B of a lattice L, find $v \in L \setminus \{0\}$ s.t. $\|v\| \leq \gamma \cdot \lambda_1(L)$.

Definition (SIVP): Let $\gamma \geq 1$. The γ-shortest independent vector problem ($SIVP_\gamma$) is as follows: given as input a basis B of a lattice L, find $v_1, \ldots, v_n \in L$, linearly independent, s.t. $\max_i \|v_i\| \leq \gamma \cdot \lambda_n(L)$.

The parameter γ is called the approximation factor. When we increase γ, the problems become no harder.

II) Cryptanalysis:

As a first approximation, it is reasonable to assume that all three problems BDD_γ, SVP_γ and $SIVP_\gamma$ are equivalent, up to a polynomial loss on the approximation factor.

(This is not completely correct: the only reductions we know are

here, $\boxed{A} \to \boxed{B}$ means that there is a polynomial time reduction from A to B: if we have a ppt algorithm for B, then we also have a ppt algo for A.

For more details, see p. 1 of Noah Stephens-Davidowitz's article "Dimension-preserving reductions between lattice problems").

The best known strategy to solve the 3 hard problems from the previous section is always the same: start with an arbitrary basis of L (made of long vectors) and reduce it to obtain a new basis B_s made of short vectors. Then use this short basis

to solve * SVP (return the 1st vector)

*SIVP (return all the basis vectors)

*BDD (use Babai algorithm, see exercise 4)

Transforming a bad basis into a good (=short) basis is called lattice reduction. In this section we will see different algorithms for performing this task.

1) Lattice reduction in dimension 2
2) The LLL algorithm
3) Sieving algorithm
4) BKZ trade-offs
5) Concrete numbers

} on slides

III) LWE

1) Definitions:

1.1 Discrete Gaussian distributions:

Definition: The Gaussian function over $\mathbb{R}^m$ with parameter $\sigma > 0$ is $\rho_\sigma : \mathbb{R}^m \longrightarrow \mathbb{R}_{>0}$

$$x \longmapsto \exp\left(-\pi \frac{\|x\|_2^2}{\sigma}\right)$$

Let $L \subseteq \mathbb{R}^m$ be a lattice, $\sigma > 0$, $c \in \mathrm{Span}_{\mathbb{R}}(L)$.
The discrete Gaussian distribution over L, with center c and parameter σ, is the distribution $D_{L,\sigma,c}$ defined over L by

$$\Pr_{x \leftarrow D_{L,\sigma,c}}(x = v) = \frac{\rho_\sigma(v-c)}{\rho_\sigma(L-c)}, \quad \forall v \in L$$

where $\rho_\sigma(L-c) = \sum_{w \in L} \rho_\sigma(w-c)$.

The discrete Gaussian distribution has many nice properties that are convenient for the proofs. We will not use them in this course though, so if you prefer, you can replace the Gaussian distrib by the uniform distrib over $[-\sigma, \sigma] \cap \mathbb{Z}$ (the results may not be completely correct anymore with the uniform distrib, but this is a good enough first approximation).

Theorem: Let B be a basis of a lattice L_1, $t \in \mathrm{span}_{\mathbb{R}}(L)$ of rk n

and $\sigma \geq \sqrt{\frac{\ln(2n+4)}{\pi}} \times \max_i \|b_i\|$,

then there is a ppt algorithm that samples from the distribution $D_{L,\sigma,t}$.

(ref: Lemma 2.3 of "Classical hardness of learning with errors" by Brakerski, Langlois, Peikert, Regev, Stehlé)

1.2 Learning with errors (LWE):

Let $s \in (\mathbb{Z}/q\mathbb{Z})^n$

Definition: Let $n \geq 1$, $q \geq 2$ and $\alpha \in (0,1)$. The LWE distribution $D^{LWE}_{n,q,\alpha}(s)$ is a distribution over $(\mathbb{Z}/q\mathbb{Z})^n \times (\mathbb{Z}/q\mathbb{Z})$ obtained with the following experiment:

* sample $a_i \leftarrow U((\mathbb{Z}/q\mathbb{Z})^n)$
* sample $e_i \leftarrow D_{\mathbb{Z},\alpha q}$ (discrete gaussian distribution over $\mathbb{Z}$ with center 0 and param $\sigma = \alpha q$)
* return $(a_i, \underbrace{\langle a_i, s \rangle + e_i \bmod q}_{=: b_i})$

The search/decision learning with errors problems are as follows:

* search $LWE_{q,n,\alpha}$: Let $s \leftarrow U((\mathbb{Z}/q\mathbb{Z})^n)$. Given an arbitrary number of samples from $D^{LWE}_{n,q,\alpha}(s)$, find s.

* decision-$LWE_{n,q,\alpha}$: Let $s \leftarrow U((\mathbb{Z}/q\mathbb{Z})^n)$. Distinguish between the distributions $D^{LWE}_{n,q,\alpha}(s)$ and $U((\mathbb{Z}/q\mathbb{Z})^n \times (\mathbb{Z}/q\mathbb{Z}))$ (given arbitrary many samples).

Remarks:

* We usually write m the number of samples from the LWE distribution (chosen by the attacker) and we use the matrix notation $\hat{A} \in \mathbb{M}_{m,n}$ (whose rows are the a_i^T) and $b = A s + e$ (where $b=(b_i)_i$ and $e=(e_i)_i$)

* if $\alpha = 0$ (which implies $e=0$), then search/dec-LWE can be solved by linear algebra

(given $b = A s$ and A, we can recover s)

* if $\alpha \gg 1$, then $e \leftarrow U(\mathbb{Z}/q\mathbb{Z})$ and so $D^{LWE}_{n,q,\alpha}(s) = U((\mathbb{Z}/q\mathbb{Z})^n \times (\mathbb{Z}/q\mathbb{Z}))$

(both search/dec-LWE are provably hard)

* We will see below that if α is small enough and m large enough, then s is uniquely defined, and so search-LWE is well-defined.

Lemma: Assume that q is prime and that

$$\alpha < \frac{1}{16\, q^{n/m} \sqrt{5m}}.$$

Then, with probability $\geq 1 - 2^{-(m-1)}$ over the choice of the $LWE_{q,n,m,\alpha}$ instance $(\boxed{A}, \boxed{b})$, one can recover Λ by solving a BDD_γ instance in $\Lambda(A)$ (with target $t = b$), where $\gamma = \frac{\alpha^{-1}}{8\, q^{n/m} \sqrt{5m}}$ (> 2).

Proof: Let $B = \alpha q \sqrt{5m}$. Properties of the discrete Gaussian distribution shows that $\Pr_{e \leftarrow D_{2^m, \alpha q}}(\|e\| \geq B) \leq 2^{-m}$. (see Banaszczyk, 1993, Lemma 1.5).

The rest of the proof is done in exercise 6.

(replacing B in exercise 5 by $\alpha q \sqrt{5m}$ gives the above result).

LWE = BDD restricted to image lattices (i.e., lattices of the form $\Lambda(A)$)

2) Hardness:

2.0) Reductions:

How do we prove that a problem is hard?
By reducing it to another problem we believe to be hard.

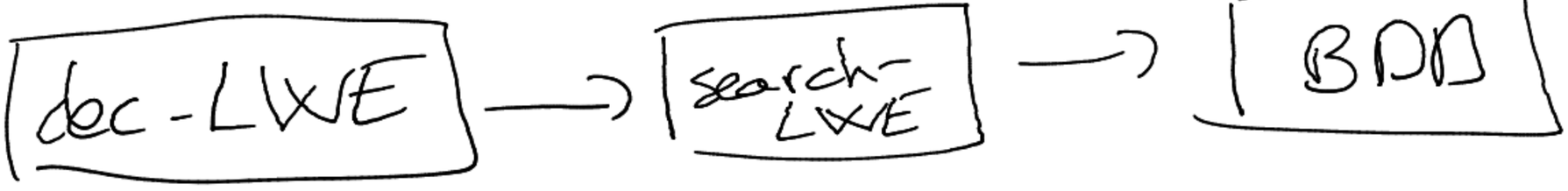
If A and B are 2 algorithmic problems, we say that A reduces (Turing) to B if there exists a polynomial time algorithm that can solve B by making (polynomially many) calls to an oracle solving A. This means that if A can be solved in polynomial time, then so does B: B is no harder than A.

We will write it

⚠ usually, drawings are quite inacurate, because for readability, we drop the parameters of the problems (e.g., we write LWE instead of $LWE_{n,q,\alpha}$)
⇝ the reductions are only valid for some parameters, not noted on the figures.

2.1) LWE hardness:

We have already seen



(The left reduction is "immediate")

In this section we will see reverse reductions.

History:

* Regev '05: quantum reduction from SIVP to LWE
* Peikert '09: classical reduction but requires $q \geq 2^{n/2}$
* Brakerski-Langlois-Peikert-Stehlé '13: classical reduction with polynomial modulus but SIVP in dim $\sqrt{n}$ (instead of n)

(Regev '05)

Theorem: Let $q > 2$ prime and $\alpha \in (0,1)$ be such that $\alpha q \geq 2\sqrt{n}$ and $q = poly(n)$.
Then there exists a worst-case to average-case quantum polynomial time reduction from $SIVP_\gamma$ in dimension n to search-$LWE_{q,n,\alpha}$ for some $\gamma = \tilde{O}(n/\alpha)$

Proof: admitted

Lemma: There is a polynomial time reduction from search-$LWE_{q,m,n,\alpha}$ to decision-$LWE_{q,m,n,\alpha}$, when $q = poly(n)$. and q is prime

Proof: Suppose that we have a poly time algorithm A that solves dec-LWE with non-negligible probability. Then, with non-negligible probability over the choice of s, A can distinguish between $D^{LWE}_{n,q,\alpha}(s)$ and $U(\mathbb{Z}_q^{mxn}, \mathbb{Z}_q^m)$ (with non-negligible advantage)

Let us use A to solve search-LWE.

Let $(a,b) = (a, \langle a, s\rangle + e)$ be an instance of search-LWE, and assume for the moment that s is among the ones for which A can distinguish between $D^{LWE}_{n,q,\alpha}(s)$ and uniform.

Let us use A to recover s_1, the first coordinate of s.

We guess s_1^*, the value of s_1 (we make at most $q = poly(n)$ guesses). Then, we ask for a $D^{LWE}_{n,q,\alpha}(s)$ sample $(a, \overbrace{\langle a, s\rangle + e}^{b})$, and we create

* $a' = a + u\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}$ where $u \leftarrow U(\mathbb{Z}_q)$

* $b' = b + u s_1^*$

We have that $(a', b') = (a', a's + e + u(s_1^* - s_1))$

If $s_1 = s_1^*$, this is a $D^{LWE}(s)$ sample.

If $s_1 \neq s_1^*$, since q is prime, then $s_1^* - s_1$ is invertible and so $u(s_1^* - s_1)$ is uniform in $\mathbb{Z}_q$, i.e., $(a', b') \sim U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

By running A on (a', b') (multiple times), we learn which guess for s_1 is correct.

We can recover the other coordinates of s similarly.

- Let us now consider the case where s is not in the set for which A can distinguish $D^{LWE}(s)$ from uniform. In this case, we can re-randomize s: we sample $r \leftarrow U(\mathbb{Z}_q^n)$.

Given a $D^{LWE}(s)$ sample $(a, b) = (a, \langle a, s \rangle + e)$, we compute $(a', b') = (a, b + \langle a, r \rangle) = (a, \langle a, s + r \rangle + e)$.

This is a sample from $D^{LWE}(s+r)$, and $s+r$ is uniform in $\mathbb{Z}_q^n$ since r is. Moreover, recovering $s+r$ allows to recover s since r is known.