M2 CRYPTO
RÉSEAUX EUCLIDIENS EN CRYPTOGRAPHIE - TD 2

Exercice 1. On se propose de montrer le résultat suivant :

Pour tout réseau $(\mathcal{L}, \langle , \rangle)$ *de rang* 2*, il existe une base* $(\mathbf{b}_1, \mathbf{b}_2)$ *telle que* $\mathbf{b}_1$ *est un plus court vecteur de* $\mathcal{L}$ *et* $|\langle \mathbf{b}_1, \mathbf{b}_2 \rangle| \leq \frac{1}{2}\|\mathbf{b}_1\|^2$.

Pour cela, on considère l'algorithme suivant, attribué à Gauss et à Lagrange.

Algorithm 1: Algorithme de Gauss-Lagrange

input : Une base $(\mathbf{b}_1, \mathbf{b}_2)$ d'un réseau $\mathcal{L}$, avec $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$
output: Une base $(\mathbf{b}, \mathbf{b}')$ satisfaisant les hypothèses du lemme.
repeat
  $x \leftarrow \lfloor \frac{\langle \mathbf{b}_1, \mathbf{b}_2 \rangle}{\|\mathbf{b}_1\|^2} \rceil$
  $\mathbf{t} \leftarrow \mathbf{b}_2 - x\mathbf{b}_1$
  $\mathbf{b}_2 \leftarrow \mathbf{b}_1$
  $\mathbf{b}_1 \leftarrow \mathbf{t}$
until $\|\mathbf{b}_1\| \geq \|\mathbf{b}_2\|$;
return $(\mathbf{b}_2, \mathbf{b}_1)$

On s'occupe d'abord de la correction de l'algorithme.

(1) Montrer qu'à chaque itération, l'algorithme construit une base de $\mathcal{L}$.

(2) Notons $(\mathbf{b}'_1, \mathbf{b}'_2)$ une base obtenue après une itération de la boucle. Montrer que pour tout $z \in \mathbb{Z}$ on a $\|\mathbf{b}'_1 + z\mathbf{b}'_2\| \geq \|\mathbf{b}'_1\|$. En déduire qu'en sortie de l'algorithme, on a $|\langle \mathbf{b}, \mathbf{b}' \rangle| \leq \|\mathbf{b}\|^2/2$.

Le second minimum d'un réseau $\mathcal{L}$ est $\lambda_2(\mathcal{L}) = \min\{r > 0 \ : \ \dim \mathrm{Vect}_{\mathbb{R}}(\mathcal{B}(0, r) \cap \mathcal{L}) \geq 2\}$.

(3) Montrer que $\|\mathbf{b}\| = \lambda_1(\mathcal{L})$ et que $\|\mathbf{b}'\| = \lambda_2(\mathcal{L})$.

Si l'algorithme termine, il renvoie donc une base qui satisfait le lemme. On s'intéresse maintenant au temps d'exécution. Le but est de montrer qu'à chaque passage dans la boucle, la quantité $(\|\mathbf{b}_1\|\|\mathbf{b}_2\|)^2$ est diminué d'un facteur au moins 3.

(4) Montrer que si $x = 0$ alors la boucle est terminée.

(5) Montrer que $|x| = 1$ n'est possible qu'à la première ou à la dernière itération de l'algorithme (on raisonnera par l'absurde en montrant que r n'est pas alors le choix minimal).

On suppose maintenant qu'on a une base $(\mathbf{b}_1, \mathbf{b}_2)$ donnant $|x| \geq 2$. Notons $\widetilde{\mathbf{b}}_2$ la projection de $\mathbf{b}_2$ sur $(\mathbb{R}\mathbf{b}_1)^{\perp}$.

(6) Montrer que $\|\mathbf{b}_2\|^2 \geq \|\widetilde{\mathbf{b}}_2\|^2 + \frac{9}{4}\|\mathbf{b}_1\|^2$ et que $\|\mathbf{t}\|^2 \leq \|\widetilde{\mathbf{b}}_2\|^2 + \frac{1}{4}\|\mathbf{b}_1\|^2$. En déduire que $\|\mathbf{b}_2\|^2 \leq 3\|\mathbf{t}\|^2$, si l'on n'est pas à la dernière itération.

(7) Si $(\mathbf{b}_0, \mathbf{b}'_0)$ est la base en entrée de l'algorithme, conclure qu'on passe au plus $4\log_3(\|\mathbf{b}'_0\|^2)$ fois dans la boucle.

Exercice 2. Le problème CVP (pour *Closest Vector Problem*) consiste à trouver dans un réseau donné le point le plus proche d'une cible fixée dans l'espace ambiant. C'est un problème difficile en général, et cet exercice propose d'étudier des algorithmes résolvant des variantes de ce problème.

(1) Echauffement : montrer qu' on sait très bien résoudre CVP en rang 1.

(2) Soit maintenant une base $(\mathbf{b}_i)_{1\leq i\leq n}$ de $\mathcal{L}\subset\mathbb{R}^n$, de matrice $\mathbf{B}$ dans la base canonique. Pour une cible $\mathbf{t}$, on considère le *round-off* $\mathbf{s}=\mathbf{B}\lceil\mathbf{B}^{-1}\mathbf{t}\rfloor$, où l'entier le plus proche est pris coordonnée par coordonnée. Montrer $\|\mathbf{t}-\mathbf{s}\|\leq\frac{n}{2}\cdot\max\|\mathbf{b}_i\|$.

On considère maintenant la version relaxée du problème, importante en pratique. Pour $\gamma>0$, le problème $\mathrm{CVP}_\gamma(\mathbf{t})$ est :
Etant donné une base d'un réseau $\mathcal{L}\subset\mathbb{R}^n$, trouver $\mathbf{s}\in\mathcal{L}$ tel que $\|\mathbf{t}-\mathbf{s}\|\leq\gamma\cdot d(\mathbf{t},\mathcal{L})$.

L'algorithme *Nearest Plane*, attribué à Babai, résout ce problème dans un régime de paramètres. C'est ce qu'on va étudier dans la suite de l'exercice. À gauche, on décrit géométriquement les étapes de l'algorithmes, et à droite on fournit un pseudo-code équivalent.

Nearest Plane :

Entrées : $(\mathbf{b}_i)$ une base de $\mathcal{L}$, $\mathbf{t}\in\mathbb{R}^n$
Sortie : $\mathbf{s}\in\mathcal{L}$ "proche" de $\mathbf{t}$.

(1) Projeter $\mathbf{t}$ sur $\mathbf{t}'$ dans l'espace ambiant de $\mathcal{L}$;
(2) Si on est en dimension 1, renvoyer le plus proche vecteur de $\mathbf{t}'$;
Sinon, trouver $\mathrm{Vect}_{\mathbb{R}}(\mathbf{b}_1,\dots,\mathbf{b}_{n-1})+\mathbf{y}$, avec $\mathbf{y}\in\mathcal{L}$, le plus proche de $\mathbf{t}'$;
(3) Se ramener "dans un réseau" : $\mathbf{t}'\leftarrow\mathbf{t}'-\mathbf{y}$;
(4) Rappeler sur $\mathbf{t}'$ et $\mathbf{b}_1,\dots,\mathbf{b}_{n-1}$, ceci donne $\mathbf{s}'$;
(5) On corrige la translation : renvoyer $\mathbf{s}'+\mathbf{y}$;

(1) $\mathbf{s}\leftarrow 0,\mathbf{e}_{n+1}\leftarrow\mathbf{t}$;
(2) Pour $i=n$ à 1, faire :
— $c_i\leftarrow\lceil\frac{\langle\mathbf{e}_{i+1},\widetilde{\mathbf{b}_i}\rangle}{\|\widetilde{\mathbf{b}_i}\|^2}\rfloor$;
— $\mathbf{s}\leftarrow\mathbf{s}+c_i\mathbf{b}_i$;
— $\mathbf{e}_i\leftarrow\mathbf{e}_{i+1}-c_i\mathbf{b}_i$;
(3) Renvoyer $\mathbf{s}$

(3) Monter que Nearest Plane renvoie $\mathbf{s}$ tel que $\|\mathbf{t}-\mathbf{s}\|^2\leq\frac{1}{4}\sum_i\|\widetilde{\mathbf{b}_i}\|^2$.
Indication : montrer que $\mathbf{e}=\mathbf{e}_1\in\{\sum_i x_i\widetilde{\mathbf{b}_i}\ :\ |x_i|\leq\frac{1}{2}\}$.

(4) Si $(\mathbf{b}_i)_i$ est $\frac{3}{4}$-LLL-réduite, montrer qu'on a $\|\mathbf{t}-\mathbf{s}\|\leq 2^{n/2}\|\widetilde{\mathbf{b}_n}\|$.

On va maintenant formaliser le comportement de l'algorithme et montrer comment résoudre $\mathrm{CVP}_{2^{n/2}}(\mathbf{t})$ en temps polynomial.

(5) Que peut-il se passer à chaque itération ? *Indication : Raisonner sur l'emplacement du plus proche vecteur $\mathbf{u}\in\mathcal{L}$ de $\mathbf{t}$, c'est-à-dire $\|\mathbf{t}-\mathbf{u}\|=d(\mathbf{t},\mathcal{L})$.*

(6) Pour tout $1\leq i\leq n$, soit $V_i=\mathrm{Vect}_{\mathbb{R}}(\mathbf{b}_1,\dots,\mathbf{b}_i)$, et supposons $\mathbf{t}\in V_{i+1}$. Si $\mathbf{y}=\lceil t_{i+1}\rfloor\mathbf{b}_{i+1}$, montrer que $d(\mathbf{t},V_i+\mathbf{y})=\min_{\mathbf{x}\in\mathcal{L}}d(\mathbf{t},V_i+\mathbf{x})$. Autrement dit, l'algorithme trouve l'hyperplan affine le plus proche de $\mathbf{t}$ parmi les translatés de V_i par le réseau.

(7) Supposons que $\mathbf{u}\notin V_i+\mathbf{y}$ et montrer qu'à cette étape, l'algorithme renvoie alors $\mathbf{s}'$ à moins de $2^{n/2}d(\mathbf{t},\mathcal{L})$.

(8) On suppose maintenant $\mathbf{u}\in V_i+\mathbf{y}$. Montrer qu'on peut se ramener à une instance d'un problème CVP en dimension i.

(9) Conclure en procédant par récurrence.