Réduction Recherche-Décision LWE

On rappelle la définition de la distribution LWE : $D^{LWE}_{n,q,\alpha}(\mathbf{s})$ est la distribution discrète sur $\mathbb{Z}_q^{n+1}$ obtenue par :

1. $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$
2. $e \hookleftarrow D_{\mathbb{Z}^\ell,\alpha}$, ie e est un vecteur court dans $\mathbb{Z}^\ell$.
3. Renvoyer $(\mathbf{a}, (\mathbf{a}\cdot\mathbf{s}) + e \mod q)$

On rappelle que le problème de **recherche** est de trouver $\mathbf{s}$ à partir de tirages LWE, et que le problème de **décision** est de distinguer entre des tirages LWE et $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ avec probabilité 1/2+constante (disons 3/4).

Nous allons montrer :

Lemma 1. *Lorsque q est polynomial en n, Recherche-LWE et Décision-LWE sont équivalents en termes de complexité computationnelle.*

Question 1. *Montrer la réduction de Décision à Recherche : étant donné un algorithme $\mathcal{A}$ qui résout le problème de Recherche, en déduire un algorithme pour résoudre le problème de Décision.*

Solution. *Supposons que nous avons un algorithme $\mathcal{A}$ qui résout le problème de recherche, nous l'utilisons pour résoudre le problème de décision.*

En entrée du problème de décision, nous recevons un ensemble d'échantillons : $\mathbf{a}_i, b_i$ où soit $b_i = \mathbf{a}_i \cdot \mathbf{s} + \mathbf{e}_i$, soit b_i est uniforme et indépendant de $\mathbf{a}_i$.

Notre stratégie est la suivante : nous prenons ces échantillons et les donnons à la boîte noire de recherche. La boîte noire de recherche renvoie une valeur $\mathbf{s}$. Nous vérifions si $\mathbf{b} - \mathbf{As}$ est court. Si c'est le cas, nous disons que les échantillons sont de type LWE. Sinon, nous disons qu'ils sont aléatoires.

Cette procédure réussit-elle, et avec quelle probabilité ? L'idée est la suivante. Regardons d'abord le cas LWE. Dans ce cas, l'algorithme de recherche réussira (avec grande probabilité). À son tour, notre recalcul de $\mathbf{b} - \mathbf{As}$ renverra en effet un vecteur court. Par conséquent, nous réussissons à reconnaître le cas LWE.

Ensuite, regardons le cas aléatoire. Dans ce cas, l'algorithme de recherche échouera. Il renverra un certain vecteur $\mathbf{s}$. En calculant $\mathbf{b} - \mathbf{As}$, nous obtenons un vecteur aléatoire dans $\mathbb{Z}_q^n$. Il est très peu probable que ce vecteur soit court. Donc notre procédure réussit avec une probabilité écrasante.

Question 2. *Montrer que si l'on a un algorithme pour Décision-LWE qui fonctionne sur une entrée $\mathbf{s}$ uniformément aléatoire (ce qui est l'hypothèse de départ), on peut contruire un algorithme fonctionnant sur une entrée $\mathbf{s}$ quelconque fixée.*

Solution. *L'idée de cette question est juste de voir que si s est fixe, on peut le randomiser pour donner à l'algo de décision exactement ce dont il a besoin.*

En effet, lorsque $\mathbf{s}$ est fixé, nous pouvons échantillonner $\mathbf{t} \hookleftarrow U(\mathbb{Z}_q^n)$ et transformer : $(\mathbf{a}, \mathbf{a}\cdot\mathbf{s} + e)$ en $(\mathbf{a}, \mathbf{a}\cdot(\mathbf{s}+\mathbf{t}) + e)$ pour le secret $\mathbf{s}+\mathbf{t}$, qui devient uniforme.

Question 3. *Montrer que si l'on a un algorithme pour Décision-LWE, on peut construire un algorithme qui teste si $s_0 = k$ pour un $k \in \mathbb{Z}_q$ donné, où s_0 est la première coordonnée de $\mathbf{s}$. En déduire que Recherche-LWE peut être réduit à Décision-LWE.*

Solution. *Nous essayons de récupérer la première coordonnée de* $\mathbf{s}$*, en testant si elle est égale à une valeur* k *choisie. Comme* q *est polynomial en* n*, nous pouvons le faire pour tous les* k*, puis pour toutes les coordonnées, et ainsi nous aurons trouvé* $\mathbf{s}$*.*

Une remarque importante ici est que nous avons besoin que la routine de décision soit fiable, car nous l'appellerons plusieurs fois. Pour rendre sa probabilité d'erreur très faible, une méthode typique consiste à la faire fonctionner plusieurs fois et à prendre la majorité des résultats. Cela sera suffisant pour nous, et nous pouvons supposer à partir de maintenant que la routine de décision réussit à chaque appel.

Afin d'appeler la routine de décision, nous modifions nos entrées de manière à ce que : si nous avons deviné k *correctement, elles seront de type LWE, sinon, elles seront uniformément aléatoires. Pour cela, nous devons introduire notre propre aléa dans ces échantillons.*

Pour chaque échantillon LWE $\mathbf{a_i}, b_i = \mathbf{a_i} \cdot \mathbf{s} + e_i$*, nous le modifions comme suit : nous choisissons une nouvelle valeur* r *uniformément aléatoire, nous ajoutons* $(r, 0, \ldots, 0)$ *à* $\mathbf{a_i}$ *et nous ajoutons* $(rk) \mod q$ *à* b_i*. Ensuite :*

- *si* $s_0 = k$*, l'échantillon modifié est de la forme* $\mathbf{a_i}', \mathbf{a_i}' \cdot \mathbf{s} + e_i$ *où* $\mathbf{a_i}' = \mathbf{a_i} + (r, 0, \ldots, 0)$ *est* encore uniformément aléatoire *(ce qui est important pour notre procédure de décision !)*
- *si* $s_0 \neq k$*, l'échantillon modifié est* $\mathbf{a_i}', b_i'$ *où* $b_i' = \mathbf{a_i} \cdot \mathbf{s} + e_i + (rk) = \mathbf{a_i}' \cdot \mathbf{s} + r(k - s_0) + e_i$*. Ici,* k *et* s_0 *sont des constantes, donc* $r(k - s_0)$ *est uniformément aléatoire dans* $\mathbb{Z}_q$ *et indépendant de* $\mathbf{a}_i'$*. Cela signifie que dans ces nouveaux échantillons,* $\mathbf{a}_i'$ *est uniformément aléatoire, et* b_i' *l'est également (il devient non corrélé à* $\mathbf{a_i}'$*, pour ainsi dire.)*

Ainsi, notre procédure de décision peut distinguer dans quel cas nous nous trouvons.

La dernière étape est d'utiliser la re-randomisation de la question précédente. La procédure de décision veut un s *uniforme, mais on lui donne des entrées avec un* s *fixe (potentiellement ça pourrait être un mauvais* s *sur lequel tout rate). Donc on re-randomise en* $\mathbf{s} + \mathbf{t}$ *uniforme, on retrouve* $\mathbf{s} + \mathbf{t}$*, on en déduit* $\mathbf{s}$*.*

Chiffrement de Regev

On rappelle la définition du chiffrement de Regev.

LWE PKE

KeyGen :

- Clé privée : $\mathbf{s} \in \mathbb{Z}_q^n$ aléatoire
- Clé publique : $(\mathbf{A}, \mathbf{b} := \mathbf{As} + e)$ où $\mathbf{A}$ est une matrice aléatoire $\mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$, et $\mathbf{e} \in \mathbb{Z}_q^{\ell}$ est échantillonné en utilisant la distribution d'erreurs "petites" (i.e., Gaussienne discrète)

Enc $m \in \{0, 1\}$:

- Choisir un vecteur aléatoire $\mathbf{r} \in \{0, 1\}^{\ell}$
- Retourner $\mathbf{c_1}, c_2 := \mathbf{rA}, (\mathsf{Decompress}(m) + \mathbf{r} \cdot \mathbf{b})$

Dec $c = (\mathbf{c_1}, c_2) \in \mathbb{Z}_q^{n+1}$:

- $m = \mathsf{Compress}(c_2 - \mathbf{c_1} \cdot \mathbf{s})$

Question 4. *Montrer que le chiffrement de Regev est additivement homomorph.*

Solution. *On ne dit pas pour quelle addition, c'est subtil. Étant donnés deux messages m, m', si on les chiffre en $(\mathbf{c_1}, c_2), (\mathbf{c_1}', c_2')$, alors le chiffré : $\mathbf{c_1} + \mathbf{c_1}', c_2 + c_2'$ pourra être déchiffré en $(m \ XOR \ m')$ (avec grande probabilité).*

Following the scheme, random vectors $\mathbf{r}, \mathbf{r'} \in \{0,1\}^\ell$ are selected, but remain unknown to an eavesdropper, and the ciphertexts have the expressions :

$$\begin{cases} \mathbf{c_1}, c_2 = \mathbf{rA}, m\lfloor q/2 \rfloor + \mathbf{rAs} + \mathbf{r} \cdot \mathbf{e} \\ \mathbf{c_1}', c_2' = \mathbf{r'A}, m'\lfloor q/2 \rfloor + \mathbf{r'As} + \mathbf{r'} \cdot \mathbf{e} \end{cases} \tag{1}$$

decryption of the combined ciphertext will give the following :

$$\begin{aligned} &\mathit{Compress}\Big(c_2 + c_2' - (\mathbf{c_1} + \mathbf{c_1}') \cdot \mathbf{s}\Big) \\ &\quad = \mathit{Compress}\Big((\mathbf{r} + \mathbf{r'})(\mathbf{As} + \mathbf{e}) + (m + m')\lfloor q/2 \rfloor - (\mathbf{r} + \mathbf{r'})\mathbf{As}\Big) \\ &\quad = \mathit{Compress}\Big(\underbrace{(\mathbf{r} + \mathbf{r'}) \cdot \mathbf{e}}_{short} + (m + m')\lfloor q/2 \rfloor\Big) \end{aligned}$$

It remains to show that this will decrypt to $(m \ XOR \ m')$. The reason is as follows : if $m, m' = (0,1)$ or $(1,0)$ then we obtain 1. If $m, m' = (0,0)$ we obtain 0. Otherwise $(m + m')\lfloor q/2 \rfloor = 2\lfloor q/2 \rfloor$ is very small modulo q, so we still obtain 0.

Question 5. *En déduire qu'il n'est pas IND-CCA.*

Solution. *C'est comme l'attaque sur ElGamal.*