CryptoExperts
WE INNOVATE TO SECURE YOUR BUSINESS

Télécom Paris

et

Master Parisien de Recherche en Informatique

Attaques pratiques par réduction de réseaux, assistées par canaux auxiliaires

Projet de fin d'études

Thibauld FENEUIL

Stage du 16 Mars 2020 au 28 Août 2020

Encadrants :
Matthieu RIVAIN, CryptoExperts
Sonia BELAÏD, CryptoExperts
Mélissa ROSSI, ANSSI

Rapporteur :
Sophie Laplante, IRIF

Responsable pédagogique :
Petr Kuznetsov, Télécom Paris

Attaques pratiques par réduction de réseaux, assistées par canaux auxiliaires

Le contexte général

En 2017, le NIST (*National Institute of Standards and Technology*) a lancé le projet de cryptographie post-quantique [15] qui inclut, entre autres, un processus de standardisation. Pourquoi s'y prendre dès maintenant alors que les ordinateurs quantiques ne sont actuellement pas une menace pour la cryptographie contemporaine ? Parce que certains domaines (militaire, aérospatial, automobile, ...) utilisent des composants qui ont un cycle de vie qui peut aller jusqu'à 20 ans, et donc il faut s'en occuper dès à présent pour que nous soyons prêts quand des ordinateurs quantiques suffisamment performants verront le jour... si du moins ils apparaissent un jour.

Les technologies sur lesquelles les différents candidats à la standardisation s'appuient sont variées : codes correcteurs, polynômes multivariés, isogénies, fonctions de hachage... mais parmi les différentes soumissions, une bonne proportion se fonde sur des problèmes à base de réseaux euclidiens (FrodoKEM [5], NewHope [3], Kyber [7], Dilithium [12], qTesla [4], ...), et plus spécifiquement sur le problème LWE (*Learning With Errors*). La crytographie à base de réseaux est très représentée et très prometteuse pour les futurs standards post-quantiques.

Mais qui dit cryptographie dit cryptanalyse. Il existe déjà des briques [1] qui prennent une instance d'un problème LWE et qui étudient sa sécurité. Les meilleures attaques contre ces algorithmes utilisent des techniques de réduction de réseaux, dont la dernière étape consiste en l'exécution de BKZ [10], un algorithme qui prend un réseau en entrée et qui en retourne une base constituée de courts vecteurs.

A cela s'ajoute un article [11] du début de l'année 2020 qui tente de combler le fossé entre les attaques algébriques et les attaques par canaux auxiliaires sur le problème en question. Les auteurs de ce papier ont mis au point une nouvelle brique qui, en plus de prendre une instance LWE, prend aussi des informations supplémentaires, appelées indices, sur le secret. Cette nouvelle brique, mise à disposition à travers un framework *SageMath*, permet pour la première fois dans la littérature d'exploiter des indices probabilistes dans la réduction de réseaux euclidiens.

Le problème étudié

Cette nouvelle brique nommée *Leaky-LWE-Estimator* ne prend que des indices linéaires par rapport au vecteur secret. Un tel indice n'est qu'une information sur une combinaison linéaire des coefficients du secret. Hélas, le problème est que les attaques exploitant des canaux auxiliaires fournissent généralement des informations binaires (comme le poids de Hamming), et donc cela s'intégre mal dans le framework. De manière générale, les informations binaires sont peu compatibles avec la structure de réseaux. Un des buts de mon stage était de voir comment utiliser le framework pour élaborer des attaques contre certains candidats du processus de standardisation du NIST.

Un autre but de mon stage était d'améliorer le framework, tant sur le plan de la performance que sur l'intégration de nouveaux types d'indices. Cette amélioration permettrait de pouvoir proposer à la communauté scientifique un outil le plus complet possible, ce qui pourrait être intéressant à la fois pour l'actuel processus de standardisation et pour l'étude des implémentations matérielles futures.

Ces recherches ont été réalisées dans le prolongement de l'élaboration du framework *Leaky-LWE-Estimator* ; elles ont encadrées entre autres par Mélissa Rossi, l'un des auteurs de l'article [11] qui a présenté l'outil.

Les contributions proposées

Conformément à la problématique étudiée, les contributions de ce stage se divisent en deux parties.

Une première partie concerne l'amélioration du framework *Leaky-LWE-Estimator*. Certains calculs coûteux ont pu être retirés, permettant d'améliorer significativement la vitesse de l'outil dans certains cas particuliers. De plus, de nouveaux indices ont été ajoutés au framework : une formule plus générale permet une meilleure intégration des indices probabilistes, et une nouvelle méthode pour intégrer des indices q-modulaires a été mise au point, indices déjà utilisés dans la littérature [17] mais non exploités précédemment par l'outil au maximum de leur potentiel.

Une seconde partie concerne la mise en place d'attaques pratiques sur certains cryptosystèmes présents à la standardisation du NIST. L'article introduisant *Leaky-LWE-Estimator* présentant déjà une attaque contre FrodoKEM, l'étude s'est essentiellement portée sur Crystals-Kyber. La cible privilégiée des fuites par canaux auxiliaires a été l'implémentation de la NTT (*Number Theoretic Transform*), qui est l'équivalent de la Transformée de Fourier sur $\mathbb{Z}_q$. J'ai pu établir des résultats très similaires à des résultats existants sur ce sujet [17, 16]. En particulier, j'ai réussi à établir une attaque par gabarits (*template attack*) [9] sur la génération de clé dévoilant la clé privée en une unique trace, résultat qui peut être adapté pour dévoiler le secret commun lors d'un échange de clé en attaquant la décapsulation.

Les arguments en faveur de la validité de ces contributions

Concernant l'amélioration du framework, l'intégration d'indices de nouveaux types a été validée expérimentalement en menant l'attaque jusqu'à retrouver le vecteur secret et en comparant le résultat avec la prédiction. L'amélioration des performances a été mesurée et s'est révélée très significative dans certains cas particuliers.

Afin de simuler et de faire valider les attaques élaborées, j'ai utilisé ELMO (*Emulator for power Leakage for the M0*) [14], un outil permettant de simuler la consommation de courant d'un programme exécuté sur un processeur ARM Cortex M0, représentatif des processeurs que nous pouvons trouver dans le monde de l'Internet des Objets. En raison de la crise sanitaire actuelle et par manque de temps,je n'ai pas pu tester ces attaques sur de vrais appareils.

Le bilan et les perspectives

L'amélioration de *Leaky-LWE-Estimator*, tant sur le plan de la performance que sur le plan des fonctionnalités proposées, permet de mettre à la disposition de la communauté scientifique un outil plus performant et fournissant un service de cryptanalyse plus complet, utilisable par un cryptanalyste n'ayant aucune notion théorique sur les attaques algébriques contre les réseaux.

Même si ce stage a un peu élargi le catalogue d'indices que l'outil peut exploiter, il serait intéressant de parvenir à exploiter des indices non-linéaires pour diminuer le coût de la réduction du réseau par BKZ. De plus, l'outil ne manipule que des lois gaussiennes et modélise comme telles toutes les distributions des indices probabilistes. La structure de réseaux euclidiens se comporte bien avec de telles lois, mais il serait intéressant d'exploiter d'autres types de distribution de probabilité.

Nous pourrions même pousser l'idée plus loin en imaginant un framework qui se fonderait sur une métrique différente de la métrique euclidienne. Par exemple, en exploitant des métriques binaires, nous pourrions mettre au point un framework équivalent à *Leaky-LWE-Estimator* pour les cryptosystèmes à base de codes correcteurs.

Table des matières

1 Préliminaires

1.1 Réseaux euclidiens et problème *LWE*

Un *réseau euclidien*, noté Λ, est un sous-groupe discret additif de $\mathbb{R}^m$ qui correspond à l'ensemble des combinaisons linéaires à coefficients entiers de n ($m \geq n$) vecteurs $\{\mathbf{b}_j\} \subset \mathbb{R}^m$ linéairement indépendants :

$$\Lambda := \left\{ \sum_j z_j \mathbf{b}_j : z_j \in \mathbb{Z} \right\}.$$

Nous appelons m la *dimension* de Λ et n son *rang*. Un réseau est *de rang maximal* si $n = m$. Une matrice $\mathbf{B}$ dont les colonnes sont ces vecteurs $\{\mathbf{b}_j\}$ est appelée une *base*. Le *volume* d'un réseau Λ est défini comme $\mathrm{Vol}(\Lambda) := \sqrt{\det(\mathbf{B}^T\mathbf{B})}$. Le *réseau dual* de Λ dans $\mathbb{R}^m$ est défini par

$$\Lambda^* := \{\mathbf{y} \in \mathrm{Span}(\mathbf{B}) \mid \forall \mathbf{x} \in \Lambda, \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$$

où $\mathrm{Span}(\mathbf{B})$ est le sous-espace vectoriel engendré par les colonnes de $\mathbf{B}$.

Notons que $(\Lambda^*)^* = \Lambda$ et que $\mathrm{Vol}(\Lambda^*) = 1/\mathrm{Vol}(\Lambda)$.

Définition 1 (Vecteurs primitifs)**.** *Un ensemble de vecteurs $\mathbf{y}_1, ..., \mathbf{y}_k \in \Lambda$ est dit primitif par rapport à Λ si $\Lambda \cap \mathrm{Span}(\mathbf{y}_1, ..., \mathbf{y}_k)$ est égal au réseau généré par $\mathbf{y}_1, ..., \mathbf{y}_k$. De manière équivalente, un tel ensemble est primitif s'il peut être complété en une base de Λ. Si $k = 1$, il est équivalent de dire que $\mathbf{y}_1/i \notin \Lambda$ pour tout entier $i \geq 2$.*

Il est possible de définir de nombreux problèmes sur les réseaux euclidiens. Néanmoins, les cryptosystèmes actuels utilisant des réseaux fondent leur sécurité essentiellement sur la difficulté à résoudre le problème ci-dessous.

Définition 2 (Problème LWE calculatoire avec des petits secrets)**.** *Soient n, m et q des entiers strictement positifs, et soit χ une distribution sur $\mathbb{Z}$. Le problème LWE (Learning With Errors) calculatoire avec des petits secrets défini pour les paramètres (n, m, q, χ) est le suivant.*

Soit une paire $(\mathbf{A} \in \mathbb{Z}_q^{m \times n}, \mathbf{b} = \mathbf{A}\mathbf{z} + \mathbf{e} \in \mathbb{Z}_q^m)$ où

1. *$\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ est générée aléatoirement de manière uniforme,*
2. *$\mathbf{z} \leftarrow \chi^n$ et $\mathbf{e} \leftarrow \chi^m$ sont générés aléatoirement avec des coefficients indépendants et identiquement distribués selon la distribution χ.*

Trouver $\mathbf{z}$.

Résoudre ce problème, pour certains choix de paramètres, est supposé difficile même avec l'usage de l'informatique quantique.

1.2 Attaque primale

La meilleure attaque pour résoudre le problème LWE est appelée « Attaque Primale ». Elle consiste à transformer ce problème en un problème uSVP (*unique Short Vector Problem*), où le but est de trouver le plus petit vecteur non nul du réseau, en utilisant le réseau euclidien

$$\Lambda' = \{(\mathbf{x}, \mathbf{y}, w) \in \mathbb{Z}^m \times \mathbb{Z}^n \times \mathbb{Z} : \mathbf{x} + \mathbf{A}\mathbf{y} - w\mathbf{b} = \mathbf{0} \pmod q\}.$$

Nous pouvons remarquer que le vecteur $(\mathbf{e}, \mathbf{z}, 1)$ appartient à ce réseau et que sa norme est petite (si χ est une petite distribution). Nous supposons qu'il s'agit de l'unique vecteur court du réseau Λ'. Pour déterminer numériquement les vecteurs $\mathbf{e}$ et $\mathbf{z}$, il ne nous reste plus qu'à trouver ce plus court vecteur de Λ', et ce problème peut se résoudre avec de la réduction de réseaux. Il s'agit d'un procédé qui consiste à transformer une base quelconque d'un réseau en une base du même réseau avec de plus courts vecteurs. La figure 1 résume la méthodologie de l'attaque primale.

$$\mathsf{LWE/BDD} \longrightarrow \mathsf{uSVP}_{\Lambda'} \longrightarrow \begin{matrix}\text{Réduction}\\ \text{de réseaux}\end{matrix}$$

FIGURE 1 – Attaque primale sans indice.

Il existe plusieurs algorithmes connus pour faire de la réduction de réseaux. Le plus connu d'entre eux est l'algorithme LLL. Cet algorithme est polynomial en temps (comparé aux autres), mais il n'est pas certain que le vecteur qu'il trouve soit le plus court. Nous pouvons aussi utiliser l'algorithme BKZ, une variante de LLL qui est paramétrable par une taille de bloc β. Plus β est élevée, plus la probabilité que le vecteur trouvé soit le plus court est forte, mais cela au prix d'une complexité exponentiellement plus élevée.

Comment anticiper la valeur de la taille de bloc β nécessaire pour résoudre une instance uSVP donnée (*i.e.* pour un réseau Λ donné) ? L'état de l'art pour prédire la taille de bloc β qu'il faudra utiliser pour résoudre un tel problème est donné dans [6, 2]. Pour un réseau Λ de dimension $\dim(\Lambda)$ et de volume $\mathrm{Vol}(\Lambda)$, d'après l'état de l'art, β-BKZ peut résoudre une instance uSVP avec un secret $\mathbf{s}$ quand

$$\sqrt{\beta/\dim(\Lambda)} \cdot \|\mathbf{s}\| \leq \delta_\beta^{2\beta-\dim(\Lambda)-1} \cdot \mathrm{Vol}(\Lambda)^{1/\dim(\Lambda)}$$

où δ_β est la constante appelée facteur d'Hermite de β-BKZ. Pour $\beta \geq 50$, grâce à l'Heuristique Gaussienne [10], il est possible de prédire ce facteur qui vaut approximativement

$$\delta_\beta = \left((\pi\beta)^{\frac{1}{\beta}} \cdot \frac{\beta}{2\pi e}\right)^{1/(2\beta-2)}.$$

Le coût de l'attaque augmente avec β. Cependant, il est actuellement difficile d'estimer ce coût en une sécurité en bits, car l'état de l'art à ce sujet n'est pas encore stable et il n'y a pas de clair consensus sur le modèle à utiliser pour l'estimation. Pour ces raisons, quand les auteurs du framework *LWE-Estimator* [1] ont mis en pratique leur outil pour évaluer la sécurité des différents candidats à la standardisation du NIST, il a exprimé la sécurité en bits de chaque candidats dans tous les modèles existants pour BKZ.

De leur côté, les auteurs de l'outil *Leaky-LWE-Estimator* [11] ont préféré ne pas estimer le coût de l'attaque en bits. A la place, ils utilisent simplement la taille de bloc β pour avoir une mesure de la sécurité, mesure qu'ils expriment en *bikz*. Ainsi, ils ont laissé ouverte la question de la conversion des bikz vers des bits, conversion qui n'est pas nécessairement linéaire.

2 Le framework *Leaky-LWE-Estimator*

2.1 *Distorted* BDD, une version généralisée du problème BDD

La méthodologie des auteurs de [11] pour intégrer des indices facilitant la réduction est de représenter le problème LWE sous la forme d'un problème BDD (*Bounded Distance Decoding*) généralisé. Un problème BDD consiste à chercher le point du réseau le plus proche d'un point quelconque de l'espace.

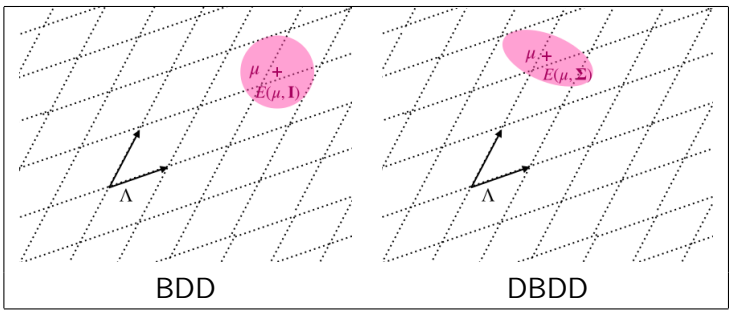


FIGURE 2 – Différence entre un problème BDD et un problème *Distorted* BDD

Leur version généralisée d'un problème BDD est de rendre l'espace anisotrope, c'est-à-dire que la notion de distance dépend de la direction. La figure 2 met graphiquement en relief la différence entre un problème BDD et un tel problème généralisé en dimension 2. Formellement, ce problème est défini de la manière suivante.

Définition 3 (Problème DBDD (*Distorted Bounded Distance Decoding*))**.** *Soient un réseau* $\Lambda \subset \mathbb{R}^d$*, une matrice symétrique* $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ *et* $\boldsymbol{\mu} \in \mathrm{Span}(\Lambda)$ *tel que*

$$\mathrm{Span}(\mathbf{\Sigma}) \subsetneq \mathrm{Span}(\mathbf{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T) = \mathrm{Span}(\Lambda).$$

Le problème $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$ *est le problème suivant.*

> *Soient* $\boldsymbol{\mu}$*,* $\mathbf{\Sigma}$ *et une base de* Λ*.*
> *Trouver l'unique vecteur* $\mathbf{x} \in \Lambda \cap E(\boldsymbol{\mu}, \mathbf{\Sigma})$

où $E(\boldsymbol{\mu}, \mathbf{\Sigma})$ *est défini comme l'ellipsoïde*

$$E(\boldsymbol{\mu}, \mathbf{\Sigma}) := \{\mathbf{x} \in \boldsymbol{\mu} + \mathrm{Span}(\mathbf{\Sigma}) \mid (\mathbf{x} - \boldsymbol{\mu}) \cdot \mathbf{\Sigma}^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu})^T \leq \mathrm{rank}(\mathbf{\Sigma})\}.$$

Par la suite, nous noterons $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \mathbf{\Sigma})$ une instance du problème $\mathsf{DBDD}_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$.

Quelques remarques pour comprendre le problème DBDD :

- Il est possible d'interpréter une instance DBDD comme la promesse que le secret suive une distribution gaussienne de moyenne $\boldsymbol{\mu}$ et de matrice de covariance $\mathbf{\Sigma}$. En réalité, c'est ce point de vue qui est utilisé pour manipuler les instances DBDD afin d'exploiter les indices.
- $\mathbf{\Sigma}$ n'a pas d'inverse, alors [11] utilise une généralisation de la notion d'inversion. Les auteurs notent $\mathbf{\Sigma}^{-1}$ la matrice vérifiant
$$\mathbf{\Sigma} \cdot \mathbf{\Sigma}^{-1} = \mathbf{\Pi}_{\mathbf{\Sigma}}$$
où $\mathbf{\Pi}_{\mathbf{\Sigma}}$ est la projection orthogonale sur $\mathrm{Span}(\mathbf{\Sigma})$. Une telle matrice peut être calculée grâce à la relation
$$\mathbf{\Sigma}^{-1} := (\mathbf{\Sigma} + \mathbf{\Pi}_{\mathbf{\Sigma}}^{\perp})^{-1} - \mathbf{\Pi}_{\mathbf{\Sigma}}^{\perp}.$$
- La condition sur $\mathbf{\Sigma}$ dans la définition est juste un détail technique. Ce n'est pas une condition restrictive : dans le cadre du framework, elle est toujours vérifiée. Sa compréhension n'est pas nécessaire pour ce rapport.

La méthodologie de [11] pour réaliser une attaque primale plus performante avec des indices consiste à transformer l'instance LWE en un instance DBDD, puis à intégrer les indices à l'aide de formulaires qui vont modifier la valeur des trois paramètres Λ, $\boldsymbol{\mu}$ et $\mathbf{\Sigma}$ de l'instance DBDD. Ensuite, il ne restera plus qu'à

transformer l'instance DBDD résultante en une instance uSVP afin de pouvoir faire, comme dans l'attaque standard, une réduction de réseaux. Schématiquement, la méthodologie peut être résumée en la figure 3.

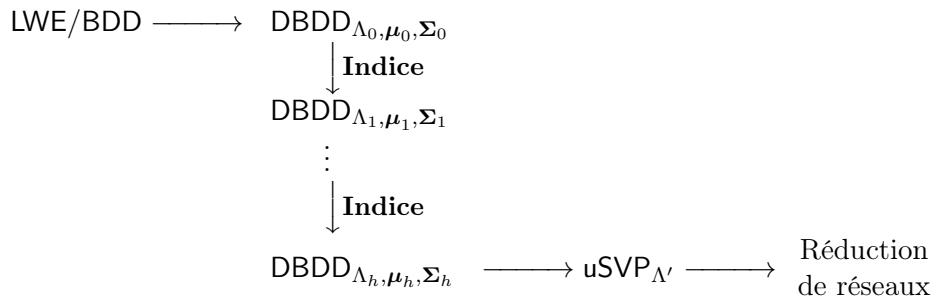


FIGURE 3 – Attaque primale avec indices (travaux de [11]).

Lorsque le framework intègre un indice, le réseau Λ est juste restreint. Ainsi, si nous trouvons la solution de la dernière instance DBDD$_{\Lambda_h,\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h}$, nous aurons directement la solution de la première instance DBDD (pas besoin de transformer la solution). Et ce sera directement la solution du problème LWE « $\mathbf{b} = \mathbf{Az}+\mathbf{e}$ », car ce sera de la forme $\bar{\mathbf{s}} := (\mathbf{e}, \mathbf{z}, 1)$.

2.2 Intégration des indices

Attention, le vocabulaire présent dans ce rapport en français, concernant les indices, **diffère** du vocabulaire présent dans l'article [11] présentant *Leaky-LWE-Estimator*. Pour les lecteurs qui ont également lu l'article d'origine, voici la correspondance du vocabulaire.

Approximate hints $\Longleftrightarrow$ Indices probabilistes
Conditioning approximate hints $\Longleftrightarrow$ Indices bruités
A posteriori approximate hints $\Longleftrightarrow$ Indices approximés

Comme expliqué précédemment, l'intégration des indices consiste en une modification d'une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Quels sont les indices que le framework *Leaky-LWE-Estimator* propose d'intégrer ? Le framework supporte l'intégration des quatre indices suivants :

- Indices parfaits : $\langle \mathbf{s}, \mathbf{v} \rangle = l$ — *Intersecter le réseau avec un hyperplan*
- Indices modulaires : $\langle \mathbf{s}, \mathbf{v} \rangle \equiv l \pmod{k}$ — *Élaguer le réseau*
- Indices bruités/approximés : $\langle \mathbf{s}, \mathbf{v} \rangle = l + \epsilon_\sigma$ — *Diminuer la covariance du secret*
- Vecteurs courts : $\mathbf{v} \in \Lambda$ — *Projeter orthogonalement à* $\mathbf{v}$

La transformation nécessaire de l'instance DBDD pour intégrer un indice est décrite dans le tableau 1. La transformation est, bien évidemment, différente selon le type d'indices.

Au premier abord, il n'est pas évident que ces différents types d'indices soient utiles dans des applications réalistes, en particulier parce que ces indices ont besoin d'être linéaires par rapport au secret. Bien entendu, le framework peut prendre en charge des indices triviaux comme la fuite d'un des coefficients du secret $\mathbf{s}_i = l$. Moins trivialement, il peut prendre en charge des fuites sur les bits de poids faible, qui seraient de la forme $\mathbf{s}_i \equiv l \pmod 2$.

Il est intéressant de noter que la plupart des calculs effectués durant un déchiffrement LWE sont linéaires : fuiter un registre intermédiaire donnerait un indice linéaire sur le secret (avec potentiellement mod q).

Fiche « Comment intégrer un indice en pratique ? »

Pour la justification des différentes formules, il faudra se référer à l'article [11]. Nous noterons $\mathbf{s} := (\mathbf{e}, \mathbf{z})$ le secret du problème LWE défini par « $\mathbf{b} = \mathbf{A}\mathbf{z} + \mathbf{e}$ » et nous noterons $\bar{\mathbf{s}} := (\mathbf{e}, \mathbf{z}, 1)$ le secret *étendu* qui correspond à la solution du problème uSVP introduit dans la section 1.2 décrivant l'attaque primale standard.

TABLEAU 1 – Formulaire de *Leaky-LWE-Estimator*

Type d'indices	**Parfait**	**Modulaire**	**Bruité**	**Approximé**	**Vecteur court**
Formalisme *	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod{k}$	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + \epsilon_{\sigma_e} = 0$	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle \sim \mathcal{D}_{ap}$	$\bar{\mathbf{v}} \in \Lambda$
$\boldsymbol{\Lambda'}$	$\Lambda \cap \{\mathbf{x} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle = 0\}$	$\Lambda \cap \{\mathbf{x} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod{k}\}$	Λ	Λ	$\Pi_{\bar{\mathbf{v}}}^{\perp} \Lambda$
$\boldsymbol{\mu'}$	$\boldsymbol{\mu} - \frac{\langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \cdot \boldsymbol{\Sigma} \bar{\mathbf{v}}$	$\boldsymbol{\mu}$ †	$\boldsymbol{\mu} - \frac{\langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2} \cdot \boldsymbol{\Sigma} \bar{\mathbf{v}}$	$\Pi_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} + \mu_{ap} \cdot \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|^2}$ ‡	$\Pi_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu}$
$\boldsymbol{\Sigma'}$	$\boldsymbol{\Sigma} - \frac{(\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}}$	$\boldsymbol{\Sigma}$ †	$\boldsymbol{\Sigma} - \frac{(\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2}$	$\Pi_{\bar{\mathbf{v}}}^{\perp} \cdot \boldsymbol{\Sigma} \cdot (\Pi_{\bar{\mathbf{v}}}^{\perp})^T + \sigma_{ap}^2 \cdot \frac{\bar{\mathbf{v}} \bar{\mathbf{v}}^T}{\|\bar{\mathbf{v}}\|^4}$ ‡	$\Pi_{\bar{\mathbf{v}}}^{\perp} \cdot \boldsymbol{\Sigma} \cdot (\Pi_{\bar{\mathbf{v}}})^T$
$\mathbf{B'}$ §	- ¶	- ¶	$\mathbf{B}$	$\mathbf{B}$	$\mathrm{LLL}(\Pi_{\bar{\mathbf{v}}}^{\perp} \mathbf{B})$
$\mathbf{D'}$ ‖	$\mathrm{LLL}(\Pi_{\bar{\mathbf{v}}}^{\perp} \mathbf{D})$	$\mathrm{LLL}(\mathbf{D} \cup \{\bar{\mathbf{v}}/k\})$	$\mathbf{D}$	$\mathbf{D}$	- ¶
$\mathbf{Vol(\Lambda')}$	$\mathrm{Vol}(\Lambda) \cdot \|\bar{\mathbf{v}}\|$ **	$\mathrm{Vol}(\Lambda) \cdot k$ **	$\mathrm{Vol}(\Lambda)$	$\mathrm{Vol}(\Lambda)$	$\mathrm{Vol}(\Lambda) \cdot \frac{1}{\|\bar{\mathbf{v}}\|}$ ††
$\mathbf{dim(\Lambda')}$	$\dim(\Lambda) - 1$	$\dim(\Lambda)$	$\dim(\Lambda)$	$\dim(\Lambda)$	$\dim(\Lambda) - 1$

Ce tableau explique comment transformer une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ en une instance DBDD $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ pour intégrer un nouvel indice.

Toutes les transformations sur le réseau Λ de l'instance DBDD peuvent être effectuées en temps polynomial. En pratique, pour garder en mémoire le réseau, nous stockons soit la base primale, soit la base duale. Et donc effectuer une transformation du réseau consiste à modifier cette base. En fonction du type d'indice, nous aurons une transformation sur la base primale ou sur la base duale. Nous pouvons, d'ores et déjà, constater que la transformation fait souvent intervenir l'algorithme LLL qui est certes de complexité polynomiale, mais rapidement très coûteux pour des grandes instances.

∗. Dans la plupart des cas, l'indice est la donnée d'un vecteur $\mathbf{v} \in \mathbb{Z}^{m+n}$ tel que nous connaissons $l \in \mathbb{Z}$ vérifiant $\langle \mathbf{s}, \mathbf{v} \rangle = l + \ldots$. Il est possible de reformuler cet indice sous la forme $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0 + \ldots$ où $\bar{\mathbf{s}}$ est le secret étendu défini précédemment et $\bar{\mathbf{v}} = (\mathbf{v}; -l)$.

†. Les formules pour $\boldsymbol{\mu'}$ et pour $\boldsymbol{\Sigma'}$ pour un indice modulaire sont pertinentes si $k^2 \ll \bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}$. Se référer à [11] si on n'est pas dans ce cas.

‡. Il est possible de remarquer que ces formules diffèrent de celles présentes dans l'article [11]. En effet, les formules du tableau sont celles qui étaient présentes dans la **première** version de [11]. Entre-temps, l'article a été corrigé avec les résultats de ma recherche décrite dans la sous-section 3.2.

§. Cette ligne consiste à calculer une base primale $\mathbf{B'}$ de $\boldsymbol{\Lambda'}$ à partir d'une base primale $\mathbf{B}$ de $\boldsymbol{\Lambda}$.

¶. Aucune moyen n'est connu pour effectuer ce calcul.

‖. Cette ligne consiste à calculer une base duale $\mathbf{D'}$ de $\boldsymbol{\Lambda'}$ à partir d'une base duale $\mathbf{D}$ de $\boldsymbol{\Lambda}$.

∗∗. Relations valides si et seulement si $\bar{\mathbf{v}}$ est primitive par rapport à Λ^*.

††. Relation valide si et seulement si $\bar{\mathbf{v}}$ est primitive par rapport à Λ.

Dans le cas où nous avons une fuite non-linéaire, il est souvent possible d'en extraire des informations linéaires. Supposons par exemple qu'une fuite nous indique que le poids de Hamming du coefficient $\mathbf{s}_0$ est 2. Alors en supposant que le support des coefficients du secret est $\{-5, ..., 5\}$, on peut en déduire que $\mathbf{s}_0 \in \{3, 5\}$. Cela nous conduit à deux indices :

- Un indice modulaire : $\mathbf{s}_0 \equiv 1 \pmod 2$;
- Un indice approximé : $\mathbf{s}_0 = 4 + \epsilon_1$ où ϵ_1 a une variance de 1.

Ce scénario reste encore relativement simple. Les auteurs de [11] nous proposent un exemple réaliste d'attaque dont les données fuitées proviennent de [8]. Un des objectifs de ce stage consistait à élaborer d'autres attaques réalistes sur des cryptosystèmes qui sont présents dans le processus de standardisation du NIST sur la cryptographie post-quantique.

3 Améliorations du framework

3.1 Réduction de dimension : vers une amélioration des performances

En utilisant cet outil pour élaborer des attaques, je me suis rapidement rendu compte qu'il présentait des problèmes de performances dès que le problème LWE « $\mathbf{b} = \mathbf{Az} + \mathbf{e}$ » à résoudre était gros, *i.e.* dès que la dimension des réseaux manipulées dépassaient environ 400. Par exemple, si nous prenons le cas de Kyber512, $\mathbf{A}$ est une matrice de $\mathbb{Z}^{512\times512}$ et lorsque nous procédons à l'attaque primale, le réseau Λ construit est de dimension $2 \times 512 + 1 = 1025$. Sachant que le processus pour intégrer des indices fait intervenir l'algorithme LLL et des inversions de matrices, il est irréalisable en pratique d'intégrer les différents indices que nous aurions à notre disposition. C'est pourquoi les auteurs du framework proposent une version *light* et une version *super-light* permettant d'estimer le coût de la réduction par BKZ sans avoir à calculer la base du réseau. Néanmoins, ces deux versions ne nous permettent pas de réaliser l'attaque en procédant à la réduction par BKZ car il nous faut justement une base de réseau à réduire.

C'est pourquoi, en pratique, quand je montais des attaques avec le framework, je faisais une étape de pré-traitement. En utilisant les indices parfaits que j'avais à ma disposition, je réduisais le problème LWE initial « $\mathbf{b} = \mathbf{Az} + \mathbf{e}$ » en un nouveau problème LWE « $\mathbf{b}' = \mathbf{A}'\mathbf{z}' + \mathbf{e}'$ » avec des dimensions raisonnables (*i.e.* avec des dimensions qui me permettaient de faire tourner la version complète du framework).

Lorsque nous intégrons un indice parfait à une instance DBDD avec le framework, nous réduisons le rang du réseau de 1 mais sa dimension reste inchangée. Et le fait que la dimension reste inchangée quel que soit le nombre d'indices que nous intégrons est ce qui nous empêche d'utiliser de grandes instances LWE avec *Leaky-LWE-Estimator*. Si nous pouvions, à moindres frais et sans perte d'information, réduire la dimension du réseau en même temps que réduire son rang, cela nous permettrait de gagner en efficacité à chaque indice que nous intégrerions car nous manipulerions des matrices de plus en plus petites. J'ai donc démontré le théorème suivant.

Théorème 1. *Soit une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Il est possible de transformer $\mathcal{I}$ en une instance $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ dont le réseau est de rang maximal.*

Si nous avons $\boldsymbol{\Gamma}$ tel que $\Lambda = \boldsymbol{\Gamma}\Lambda'$ avec Λ' de rang maximal et $\mathrm{rank}(\Lambda) = \mathrm{rank}(\Lambda')$*, alors*

$$\begin{aligned}
\mathbf{D}' &\leftarrow \boldsymbol{\Gamma}^T\mathbf{D}, \\
\boldsymbol{\mu}' &\leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T \cdot \boldsymbol{\mu}, \\
\boldsymbol{\Sigma}' &\leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1},
\end{aligned}$$

où $\mathbf{D}$ est une base duale de Λ et $\mathbf{D}'$ est la base duale correspondante de Λ'.

Démonstration. L'énoncé complet et la preuve de ce théorème se trouvent dans l'annexe B. L'énoncé complet permet d'avoir des résultats sur les volumes en jeu. Dans l'annexe, il est démontré que cette réduction n'influence absolument pas le coût de la réduction par BKZ, ce qui semble normal car ce théorème n'ajoute aucune information qui pourrait diminuer le coût de la réduction. □

L'intégration d'un indice parfait « $\langle \mathbf{s}, \mathbf{v} \rangle = l$ » réduit le rang du réseau en question sans toucher à sa dimension. Avec l'aide de ce théorème, il est possible de réduire également la dimension dès que nous connaissons une matrice $\mathbf{\Gamma}$ vérifiant les hypothèses demandées. Et c'est le cas ! En notant p l'index d'un coefficient non nul de $\bar{\mathbf{v}}$, on peut construire

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{v}_1^T & -\mathbf{v}_2^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

où $\mathbf{v}_1$ et $\mathbf{v}_2$ sont définis par la relation $\bar{\mathbf{v}}^T = \bar{\mathbf{v}}_p \cdot (\mathbf{v}_1^T, 1, \mathbf{v}_2^T)$, et cette matrice vérifie exactement ce que nous voulons. En plus, il existe une expression littérale de $(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}$ nous permettant de construire cette matrice sans coût, ce qui nous intéresse car nous souhaitons améliorer au maximum les performances de l'outil *Leaky-LWE-Estimator*.

À présent, on peut donc supposer que **les instances DBDD manipulées ont toutes leur réseau de rang maximal**.

Non seulement l'intégration des indices sera de plus en plus rapide au fur et à mesure que nous intégrerons des indices parfaits, mais en plus nous pourrons accélérer (ou même retirer) des calculs en utilisant l'hypothèse selon laquelle tous les réseaux manipulés sont de rang maximal. Par exemple, pour intégrer des indices quelconques, la version de base du framework calcule régulièrement la matrice de projection orthogonale sur le réseau. Si le réseau a pour base primale $\mathbf{B}$, cela signifie qu'elle calcule $\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$, ce qui est un calcul coûteux car il fait intervenir une inversion de matrice. Puisque nous avons supposé que le réseau est de rang maximal, cette matrice est l'identité.

Une fois que nous avons implémenté les détails techniques évoqués dans le paragraphe précédent, le coût de l'intégration d'un indice revient essentiellement au coût de l'algorithme LLL. En effet, pour chaque indice nécessitant une opération sur le réseau de l'instance DBDD, nous devons appliquer LLL pour retirer les dépendances linéaires que nous avons introduites dans la base (primale ou duale). Mais la question qui s'est alors soulevée est : pouvons-nous retirer l'exécution de LLL et travailler avec une famille génératrice du réseau au lieu d'avoir une base ?

En étudiant le fonctionnement de l'outil, j'en ai déduit qu'il n'y avait que deux circonstances où il était nécessaire d'avoir une base et pas juste une famille génératrice du réseau :

- Il faut une base quand nous voulons une estimation du volume du réseau, et nous voulons une estimation du volume quand nous voulons une estimation du coût de la réduction par BKZ.
- Il faut une base lorsque nous voulons commencer la réduction par BKZ.

La seconde circonstance n'a lieu qu'une unique fois, à la fin de l'intégration des différents indices à disposition. La première circonstance a lieu plus régulièrement. Selon la façon dont l'utilisateur utilise le framework, cela peut se produire après l'intégration de chaque indice, mais cela peut être aussi réalisé de manière moins régulière. J'ai donc décidé d'implémenter l'usage de l'algorithme LLL le plus tard possible, juste avant les circonstances où il est vraiment nécessaire d'avoir une base et pas juste une famille génératrice du réseau.

Après toutes ces améliorations de l'outil *Leaky-LWE-Estimator*, j'ai évalué expérimentalement le gain de temps sur un scénario particulier. J'ai pris l'ensemble de paramètres $\{n = 30, m = 30, q = 3301\}$ pour le problème LWE « $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{b} = \mathbf{Az} + \mathbf{e})$ » et j'ai intégré petit à petit des indices parfaits ; j'ai ensuite fait la même expérience avec des indices modulaires. J'ai obtenu les courbes de la figure 4.

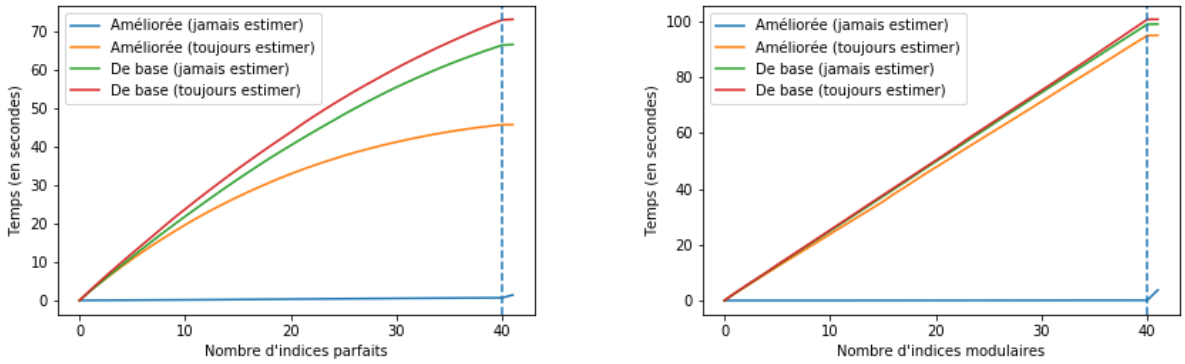


FIGURE 4 – Comparaison de la vitesse de la version de base avec celle de la version améliorée de *Leaky-LWE-Estimator*. Paramètres LWE : $n = m = 30$, $q = 3301, \sigma = 20$. Chaque point a été moyenné sur 10 échantillons. La barre verticale représente la fin de l'intégration des indices, et le dernier point correspond à la suppression des dépendances linéaires dans la famille génératrice du réseau.

Plusieurs remarques sur ces graphiques :

- Lorsque j'intègre les indices dans le pire des cas (*i.e.* quand on estime le coût de la réduction par BKZ après avoir intégré chaque indice) avec l'outil amélioré, cela reste toujours moins coûteux que dans le meilleur des cas avec l'outil de base. Cet écart est principalement justifié par les calculs qui sont simplifiés dans la version améliorée car les réseaux sont toujours de rang maximal.
- Il y a un faible écart entre le pire et le meilleur des cas dans l'outil de base car, dans les deux cas, les auteurs utilisent l'algorithme LLL lors de l'intégration de chaque indice.
- A l'opposé, il y a un fort écart entre le pire et le meilleur des cas dans l'outil amélioré car, dans le meilleur des cas, l'algorithme LLL n'est jamais utilisé.
- Les performances de l'outil amélioré dans le meilleur des cas sont particulièrement bonnes. Cela s'explique par le fait que les améliorations évoquées de cette section ont fait qu'il n'y a plus le moindre calcul coûteux dans l'intégration des indices. Il n'y a que des multiplications de matrices (il n'y a aucune inversion de matrice et l'algorithme LLL n'est pas utilisé).
- Le temps pour intégrer des indices modulaires reste constant (courbes droites) car les matrices manipulées restent toujours de la même taille. Il n'y a aucune modification du rang et de la dimension du réseau. En revanche, intégrer des indices parfaits prend de moins en moins de temps au fur et à mesure que l'intégration avance (courbes concaves). Dans la version de base, c'est parce que le rang du réseau diminue au fur et à mesure. La concavité est plus importante dans la version améliorée car la dimension du réseau diminue également (en plus du rang).
- Pour pouvoir procéder à une estimation finale ou à une attaque, il nous faut une base du réseau. Pour pouvoir comparer les différents scénarios entre eux, il me fallait donc ajouter la durée nécessaire pour retirer les dépendances linéaires ; c'est le dernier point de la courbe. Le seul moment où cette durée est non nulle, c'est quand nous ne sommes pas dans le pire des cas de la version améliorée. Dans le meilleur des cas, nous avons évité d'exécuter LLL durant toute l'intégration des indices, mais nous ne pouvons pas éviter de l'utiliser tout à la fin. Néanmoins, cela signifie que nous n'avons besoin d'exécuter cet algorithme qu'une seule fois (à l'opposé de la version de base qui l'utilise pour intégrer chaque indice) et avec, pour dimension, la dimension finale du réseau. Le réseau peut être de grande dimension au début, cela ne posera pas de problème si, à la fin, sa dimension est suffisamment réduite. Cependant, si la dimension finale reste trop grande, l'exécution de LLL sera très coûteuse.

3.2 Généralisation de la formule pour les indices approximés

L'outil *Leaky-LWE-Estimator* permet l'intégration d'indices qui sont sous la forme d'une distribution *a posteriori* de $\langle \mathbf{s}, \mathbf{v} \rangle$. L'intégration de tels indices est très intéressante car il s'agit de la forme typique des indices qu'une attaque par gabarits [9] peut fournir. La transformation de l'instance DBDD permettant l'intégration d'un tel indice dans le framework est donnée par le théorème suivant.

Théorème 2. *Considérons l'indice suivant : nous connaissons la distribution* a posteriori $\mathcal{D}_{ap}$ *de* $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle$. *Notons* μ_{ap} *la moyenne de la distribution et* σ_{ap}^2 *sa variance. Pour prendre en compte cet indice dans une instance Distorted BDD* $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, *nous devons transformer* $\mathcal{I}$ *en* $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ *avec*

$$\begin{aligned}
\Lambda' &= \Lambda, \\
\boldsymbol{\mu}' &= \Pi_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} + \mu_{ap} \cdot \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|^2}, \\
\boldsymbol{\Sigma}' &= (\Pi_{\bar{\mathbf{v}}}^{\perp}) \cdot \boldsymbol{\Sigma} \cdot (\Pi_{\bar{\mathbf{v}}}^{\perp})^T + \sigma_{ap}^2 \cdot \frac{\bar{\mathbf{v}}\bar{\mathbf{v}}^T}{\|\bar{\mathbf{v}}\|^4}.
\end{aligned}$$

Hélas, il s'est révélé par la suite que ces formules sont valides **uniquement dans le cas où le vecteur $\bar{\mathbf{v}}$ est un vecteur propre de $\boldsymbol{\Sigma}$**. C'est pourquoi j'ai prouvé le théorème suivant, dont la démonstration se trouve dans l'annexe C.

Théorème 3. *Considérons l'indice suivant : nous connaissons la distribution* a posteriori $\mathcal{D}_{ap}$ *du terme* $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + e$, *où* e *modélise du bruit qui suit une distribution gaussienne* $\mathcal{N}_1(0, \sigma_e^2)$, ***indépendante de*** $\mathbf{s}$. *Notons* μ_{ap} *la moyenne de la distribution et* σ_{ap}^2 *sa variance. Pour prendre en compte cet indice dans une instance Distorted BDD* $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, *nous devons transformer* $\mathcal{I}$ *en* $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ *avec*

$$\begin{aligned}
\Lambda' &= \Lambda, \\
\boldsymbol{\mu}' &= \hat{\Pi}_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} + \mu_{ap} \cdot \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2}, \\
\boldsymbol{\Sigma}' &= \hat{\Pi}_{\bar{\mathbf{v}}}^{\perp} \cdot \boldsymbol{\Sigma} \cdot (\hat{\Pi}_{\bar{\mathbf{v}}}^{\perp})^T + (\sigma_{ap}^2 + \sigma_e^2) \cdot \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2)^2},
\end{aligned}$$

où

$$\hat{\Pi}_{\bar{\mathbf{v}}} := \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2} \qquad \textit{et} \qquad \hat{\Pi}_{\bar{\mathbf{v}}}^{\perp} := I - \hat{\Pi}_{\bar{\mathbf{v}}}.$$

Remarque. *L'idée de la pseudo-projection* $\hat{\Pi}_{\bar{\mathbf{v}}}^{\perp}$ *(c'est une projection oblique si* $\sigma_e^2 = 0$*) est de* ***retirer la contribution de*** $\bar{\mathbf{v}}$ ***dans la moyenne*** $\boldsymbol{\mu}$, *afin d'insérer une nouvelle contribution ensuite. S'il n'y a pas de bruit (*$\sigma_e^2 = 0$*), alors*

$$\begin{aligned}
\langle \bar{\mathbf{v}}, \hat{\Pi}_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} \rangle &= \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle - \langle \bar{\mathbf{v}}, \hat{\Pi}_{\bar{\mathbf{v}}} \boldsymbol{\mu} \rangle \\
&= \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle - \frac{1}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \langle \bar{\mathbf{v}}, \boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T \boldsymbol{\mu} \rangle \\
&= \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle - \frac{1}{\langle \bar{\mathbf{v}}, \boldsymbol{\Sigma}\bar{\mathbf{v}} \rangle} \langle \bar{\mathbf{v}}, \boldsymbol{\Sigma}\bar{\mathbf{v}} \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle \rangle \\
&= \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle - \frac{1}{\langle \bar{\mathbf{v}}, \boldsymbol{\Sigma}\bar{\mathbf{v}} \rangle} \langle \bar{\mathbf{v}}, \boldsymbol{\Sigma}\bar{\mathbf{v}} \rangle \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle = 0.
\end{aligned}$$

Et il en est de même pour $\boldsymbol{\Sigma}'$. *La pseudo-projection* $\hat{\Pi}_{\bar{\mathbf{v}}}^{\perp}$ ***retire la contribution de*** $\bar{\mathbf{v}}$ ***dans la matrice de covariance*** $\boldsymbol{\Sigma}$, *afin d'insérer une nouvelle contribution ensuite.*

Ce théorème propose la version la plus générale des formules pour intégrer des indices. En effet, à partir de ce théorème, on peut (re)démontrer :

- les formules pour intégrer un indice parfait en prenant $\mu_{ap} = 0$, $\sigma_{ap}^2 = 0$ et $\sigma_e^2 = 0$;
- les formules pour intégrer un indice bruité en prenant $\mu_{ap} = 0$ et $\sigma_{ap}^2 = 0$;
- les formules pour intégrer un indice approximé en prenant $\sigma_e^2 = 0$.

A l'heure actuelle, je n'ai pas trouvé d'application concrète à la version la plus générale des formules, mais l'intérêt de ce théorème est qu'il permet d'avoir un moyen d'intégrer les indices approximés qui est valide **même si $\bar{\mathbf{v}}$ n'est pas un vecteur propre de $\boldsymbol{\Sigma}$**. Pour les indices approximés, cela donne directement ce nouveau théorème.

Théorème 4. *Considérons l'indice suivant : nous connaissons la distribution* a posteriori $\mathcal{D}_{ap}$ *de* $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle$. *Notons* μ_{ap} *la moyenne de la distribution et* σ_{ap}^2 *sa variance. Pour prendre en compte cet indice dans une instance Distorted BDD* $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$*, nous devons transformer* $\mathcal{I}$ *en* $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ *avec*

$$
\begin{aligned}
\Lambda' &= \Lambda, \\
\boldsymbol{\mu}' &= \boldsymbol{\mu} + \frac{\mu_{ap} - \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \boldsymbol{\Sigma} \bar{\mathbf{v}}, \\
\boldsymbol{\Sigma}' &= \boldsymbol{\Sigma} + \left(\frac{\sigma_{ap}^2}{(\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}})^2} - \frac{1}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \right) (\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T.
\end{aligned}
$$

J'ai implémenté ces formules corrigées dans la version améliorée de *Leaky-LWE-Estimator*. Les auteurs de [11] ont confirmé et corrigé leur article pour y inclure mes nouvelles formules. A présent, dans [11], la première version des formules n'existe plus.

3.3 Intégration des indices q-modulaires

Dans l'article [17], les auteurs mettent au point une attaque qui exploite des fuites par canaux auxiliaires pour simplifier et résoudre un problème LWE. Les fuites donnent des informations sur le secret de la forme

$$\langle \mathbf{s}, \mathbf{v} \rangle \equiv l \pmod{q}.$$

Il s'agit d'indices modulaires comme ceux que le framework peut supporter, mais le point intéressant à faire remarquer est la valeur du modulo. Les indices sont modulo q, le même q que celui défini dans le problème LWE. Avec ces indices, les auteurs de [17] réduisent l'instance LWE en une nouvelle instance plus facile qu'ils peuvent résoudre. Et pour simplifier l'instance, ils font une simple substitution.

Si nous comparons leur méthode et la méthode avec laquelle le framework *Leaky-LWE-Estimator* intègre les indices modulo q, il est aisé de constater que le framework ne parvient pas à exploiter tout le potentiel des indices modulo q. En réalité, dans l'outil, le formulaire 1 pour intégrer les indices modulaires ne permet pas de jouer sur les paramètres $\boldsymbol{\mu}$ et $\boldsymbol{\Sigma}$, simplement parce que ces deux paramètres représentent une loi gaussienne multivariée et qu'une telle loi se comporte mal avec des modulos. En résumé, une instance DBDD se comporte mal avec des indices modulaires.

Dans ce cas, si les instances se comportent mal avec des indices modulaires, ne pourrions-nous pas réduire le problème LWE avec la méthode de [17] (comme pré-traitement) puis le transformer en problème DBDD ? La réponse est non. Si nous faisions cela, il serait impossible d'intégrer des indices probabilistes (bruités ou approximés) car la méthode d'intégration de ces indices est incompatible avec une intégration préalable d'indices modulaires par « substitution ». Or l'intégration des indices probabilistes est la plus grande plus-value de *Leaky-LWE-Estimator*.

Ce qu'il faudrait donc, c'est trouver un moyen d'intégrer des indices q-modulaires après avoir intégré des indices probabilistes. Mais cela nécessite de définir une sorte de substitution pour une instance DBDD.

Après quelques semaines de recherche, j'ai pu aboutir au théorème suivant, dont la démonstration se trouve dans l'annexe D.

Théorème 5 (Intégration des indices q-modulaires)**.** *Considérons l'indice suivant : nous connaissons un vecteur $\mathbf{v} \in \mathbb{Z}^{m+n}$ et un scalaire $l \in \mathbb{Z}$ tels que $\langle \mathbf{s}, \mathbf{v} \rangle \equiv l \pmod q$. Notons que cet indice peut être réécrit en*

$$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod q$$

où $\bar{\mathbf{s}}$ est le secret étendu défini précédemment et $\bar{\mathbf{v}} := (\mathbf{v}; -l)$.

Pour prendre en compte cet indice dans une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ de rang maximal, nous devons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\begin{aligned}
\Lambda' &= \boldsymbol{\Pi} \cdot (\Lambda \cap \{\mathbf{x} \in \mathbb{Z}^d : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod q\}), \\
\boldsymbol{\mu}' &= \boldsymbol{\Pi} \cdot \boldsymbol{\mu}, \\
\boldsymbol{\Sigma}' &= \boldsymbol{\Pi} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Pi}^T,
\end{aligned}$$

où $\boldsymbol{\Pi} \in \mathbb{R}^{(\dim(\Lambda)-1) \times \dim(\Lambda)}$ est la matrice qui retire la $p^{ième}$ coordonnée (nous retirons donc une dimension) avec p un pivot possible de $\bar{\mathbf{v}}$, i.e. $\bar{\mathbf{v}}_p \neq 0$.

La réduction de réseaux nous fournira la valeur de $\bar{\mathbf{s}}' := \boldsymbol{\Pi}\bar{\mathbf{s}}$, où $\bar{\mathbf{s}}$ est le secret étendu. Pour remonter à $\bar{\mathbf{s}}$, il faut utiliser la relation

$$\bar{\mathbf{s}} = \boldsymbol{\Gamma}\bar{\mathbf{s}}' + q \cdot \left\lfloor \frac{\langle \boldsymbol{\Pi}_{\mathbf{e}_p}^{\perp} \boldsymbol{\Gamma}\bar{\mathbf{s}}', \bar{\mathbf{v}} \rangle}{q} \right\rceil \cdot \mathbf{e}_p$$

où $\boldsymbol{\Gamma}$ est la même matrice que celle définie dans la section 3.1 de la réduction de dimension.

Ce théorème est une application d'un théorème plus général D.1 que j'ai démontré. Néanmoins, je n'ai trouvé aucune utilité à cette version plus générale, c'est pourquoi je ne l'évoque que dans les annexes afin de pouvoir démontrer ce théorème pour intégrer des indices q-modulaires. La version générale m'a toutefois permis d'avoir plus de recul sur le théorème et le fait d'utiliser une version générale est une perspective ouverte intéressante pour de nouveaux types d'indices.

La démonstration nous fournit également un moyen de calculer une base duale de Λ' avec un coût faible, dans l'hypothèse où nous avons déjà une base duale de Λ.

3.4 Validation des améliorations précédentes

Durant l'intégralité de cette section, j'ai proposé diverses améliorations pour le framework *Leaky-LWE-Estimator*, améliorations que j'ai bien entendu implémentées. Néanmoins, il faut vérifier qu'il n'y a pas d'erreur dans la théorie ou dans l'implémentation. Il suffit d'utiliser le framework pour prédire la sécurité d'instances LWE aléatoires et de vérifier quelles étaient les tailles de bloc β nécessaires pour résoudre ces instances en lançant BKZ avec β de plus en plus grand.

C'est exactement ce qu'ont fait les auteurs de [**Figure 5**, 11], obtenant ainsi une comparaison entre la prédiction via leur outil et l'expérience. Avec ces résultats, ils ont pu vérifier que leur outil donnait une prédiction très proche de la réalité, avec une légère tendance à surestimer β.

J'ai donc procédé aux mêmes tests, et j'obtiens les courbes de la figure 5 qui sont très semblables aux courbes de [11]. Nous pouvons remarquer qu'ici aussi, la prédiction a tendance à surestimer la valeur de la taille de bloc β optimale.

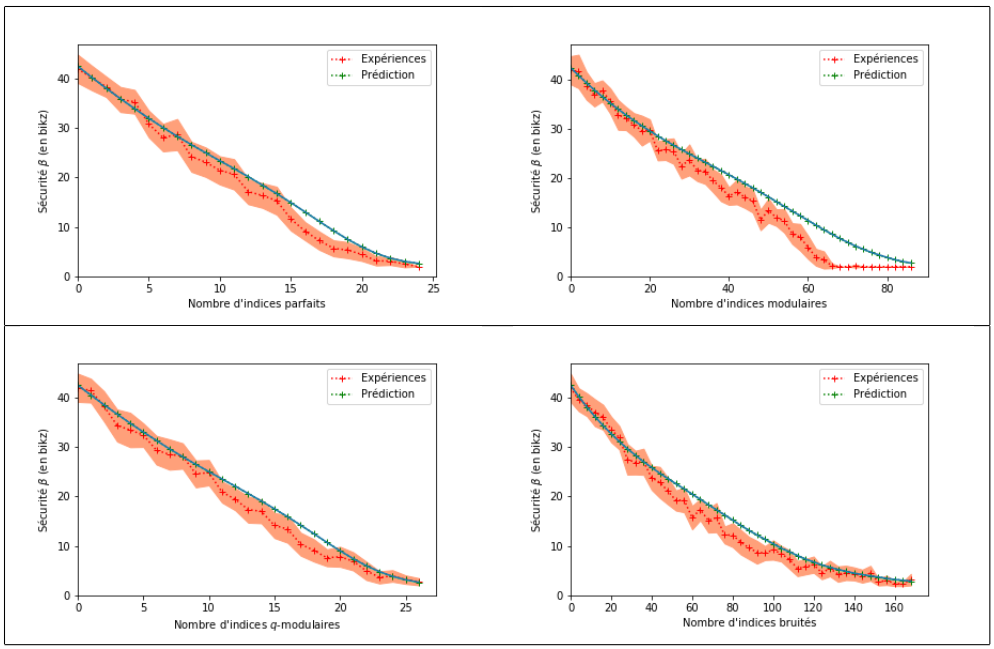


FIGURE 5 – Vérification expérimentale de la prédiction de la diminution de la sécurité pour chaque type d'indices. Paramètres LWE : $n = m = 70$, $q = 3301$, $\sigma^2 = 20$. Pour les indices modulaires, $k = 11$. Pour les indices bruités, $\sigma_e^2 = 3$. Les vecteurs indices $\mathbf{v}$ ont été choisis comme des vecteurs tertiaires aléatoires de poids 5, sauf pour les indices q-modulaires où les coefficients non nuls appartiennent à $\{0, ..., q-1\}$. Chaque point a été moyenné sur 24 échantillons. La zone orange représente l'écart-type expérimentale.

4 Attaques pratiques avec le framework

Comme décrit dans l'introduction de ce rapport, le second objectif de ce stage consistait à appliquer le framework sur différents candidats à la standardisation de la cryptographie post-quantique du NIST. Au début de ce stage, ce processus était à son stade 2, où nous pouvions trouver encore 17 candidats en lice pour les PKE/KEMs (*Public Key Encryption/Key Encapsulation Mecanism*) et 9 candidats pour les signatures. Si l'on ne considère que les cryptosystèmes à base de réseaux, il y en avait 9 pour les PKE/KEMs et 3 pour les signatures.

Le 22 Juillet 2020, le passage au stade 3 du processus de standardisation a été annoncé par le NIST. Les candidats sélectionnés pour ce nouveau stade sont

Type	**PKE/KEM**	**Signature**
A base de réseaux	• CRYSTALS-Kyber • NTRU • Saber	• CRYSTALS-Dilithium • Falcon
A base de codes	Classic McEliece	
Multivarié		Rainbow

La liste s'est considérablement restreinte. Nous pouvons le voir, la cryptographie à base de réseaux euclidiens est celle qui semble la plus prometteuse, la plupart des finalistes étant de ce type.

Tout cela nous a motivé pour tester l'outil sur différents candidats à base de réseaux euclidiens. Dans les différentes sous-sections suivantes, nous étudierons des attaques contre les trois cryptosystèmes suivants :

- FrodoKEM, un KEM qui se fonde sur un problème LWE non structuré ;

- Crystals-Kyber, un KEM qui se fonde sur un problème LWE structuré (MLWE) ;
- Crystals-Dilithium, une signature qui se fonde sur un problème LWE structuré (MLWE).

4.1 Attaque contre FrodoKEM

L'attaque sur Frodo n'est pas une contribution de ma part. En guise d'exemple d'application, les auteurs du framework ont utilisé leur outil pour monter une attaque qui exploite les fuites de données de FrodoKEM récoltées par l'attaque par gabarits de [8]. Les résultats qu'ils obtiennent sont résumés dans le tableau 2 dans lequel ils ont renseigné le coût de leurs attaques en fonction du jeu de paramètres.

	NIST1	NIST2	CCS1	CCS2	CCS3	CCS4
Attaque sans indices (bikz)	487	708	239	448	492	584
Attaque avec indices (bikz)	**330**	**423**	**128**	**123**	**219**	**230**
Attaque avec indices & estimations (bikz)	**292**	**298**	**70**	**29**	**124**	**129**
Nombre d'estimations k	100	250	200	300	250	250
Probabilité de succès	0.86	0.64	0.87	0.77	0.81	0.84

TABLEAU 2 – Coût des attaques sans/avec indices sans/avec estimations [**Tableau 3**, 11]

Dans leurs attaques, ils ont introduit la notion d' « estimation ». Il s'agit d'indices approximés qui ont été introduits dans le framework en tant qu'indices parfaits. Un indice parfait réduit nettement plus la sécurité qu'un indice approximé (voir figure 5). Néanmoins, ce remplacement implique que l'attaque ne fonctionnera pas dans 100% des cas car l'estimation pourrait être fausse. C'est pourquoi le tableau des résultats introduit une notion de « probabilité de succès ». En choisissant judicieusement les indices à estimer, on peut réduire le coût de l'attaque sans réduire trop la probabilité de succès.

Pour les jeux de paramètres CCS1 et CCS2, nous pouvons voir que le coût de l'attaque avec indices & estimations est faible. Cela implique qu'il serait possible en pratique de réaliser l'attaque. Les auteurs de *Leaky-LWE-Estimator* ont estimé le coût de la réduction, mais ne l'ont pas réalisée en pratique. En réalité, ils ne pouvaient pas la réaliser car ils avaient utilisé la version *super-light* de *Leaky-LWE-Estimator*, version qui **ne calcule pas** de base pour le réseau Λ. S'ils avaient utilisé la version complète du framework, étant donné qu'ils manipulent des réseaux de très grande dimension ($d = 1185$ pour CSS2), l'intégration des indices aurait été beaucoup trop coûteuse et donc pas réalisable en pratique.

Néanmoins, les différentes améliorations que j'ai apportées au framework pourraient changer la donne. Grâce à la réduction de dimension (sous-section 3.1) et l'optimisation de l'utilisation de l'algorithme LLL, il devient envisageable d'utiliser la version complète du framework pour effectuer la réduction pour CCS1 et CCS2.

Lorsque j'ai testé, j'obtiens un résultat mitigé. Certes, l'intégration de tous les indices est rapide, cela ne prend que quelques minutes. Néanmoins, la version améliorée de *Leaky-LWE-Estimator* manipule une famille génératrice pour le réseau, et non pas une base. Avant de pouvoir procéder à l'attaque, j'ai dû retirer les dépendances pour avoir une base, et donc j'ai dû appliquer l'algorithme LLL (*cf* dernier point de la figure 4). Hélas, le rang est encore trop élevé et cette opération dure trop longtemps. Pour l'attaque avec indices & estimations contre CCS1, le rang final du réseau est

$$\text{rank}(\Lambda) = (n + m + 1) - \text{nb\_perfect\_hints} - \text{nb\_estimations} \approx (352 + 352 + 1) - 0 - 200 = 505$$

Et c'est pire pour CCS2. L'attaque n'a pas pu aboutir. Pour réussir, il faudrait encore réduire le rang du réseau ou avoir un algorithme plus efficace pour retirer les dépendances dans une famille génératrice de réseau euclidien.

4.2 Attaque contre Kyber

L'idée de base est de s'inspirer de l'attaque contre FrodoKEM pour monter une attaque utilisant des fuites par canaux auxiliaires contre Kyber.

Dans la suite, je vais décrire l'attaque qui a été mise au point durant mon stage. Pour des raisons de simplicité et de lisibilité, je vais présenter l'idée générale sans rentrer dans les considérations techniques.

4.2.1 Définition de la cible

Contrairement à FrodoKEM, Kyber fonde sa sécurité sur un problème LWE structuré. *Qu'es aquò ?* Au lieu de travailler sur des matrices et des vecteurs avec des coefficients dans $\mathbb{Z}_q$, Kyber travaille sur des matrices et des vecteurs avec des coefficients dans $R_q = \mathbb{Z}_q[X]/(X^n+1)$. Puisqu'un élément de R_q est plus complexe qu'un élément de $\mathbb{Z}_q$, pour une sécurité fixée, les matrices du problème LWE structuré seront plus petites. Mais au lieu de manipuler des entiers, on manipule des polynômes.

Nous pourrions penser que le gain de performance lié au fait qu'on manipule des matrices plus petites est perdu par le fait que nous manipulons des objets plus complexes. Mais la structure de polynômes de R_q nous permet d'effectuer les calculs efficacement grâce à la NTT (*Number-Theoretic Transform*), qui est l'équivalent de la Transformée de Fourier sur $\mathbb{Z}_q$.

Puisque la différence fondamentale entre FrodoKEM et Kyber est l'ajout de la structure au problème LWE, nous avons décidé de prendre pour cible l'algorithme clé qui permet la manipulation efficace de cette structure ajoutée. Une question naturelle serait de savoir si l'ajout d'une telle structure, qui accélère les calculs, donne des avantages particuliers aux attaquants. Nous avons donc décidé d'attaquer l'implémentation de la NTT.

Sachant que nous voulons faire fuiter des informations sur la partie secrète $\mathbf{s} := (\mathbf{e}, \mathbf{z})$ du problème LWE « $\mathbf{b} = \mathbf{A}\mathbf{z} + \mathbf{e}$ », qui est la clé secrète du PKE/KEM, nous avons cherché où la NTT était utilisée avec, pour entrée, une donnée qui dépend du secret. Voici les endroits en question :

- Dans la génération des clés, la NTT est appliquée à la fois sur $\mathbf{z}$ et sur $\mathbf{e}$.
- Dans le déchiffrement, la NTT inverse est appliquée sur le produit de $\hat{\mathbf{z}} := \mathrm{NTT}(\mathbf{z})$ avec une partie du chiffré (qui est supposé connu).

4.2.2 Modèle de la fuite

Après avoir déterminé les endroits intéressants à étudier, il m'a fallu déterminer le modèle de fuite. Nous avons décidé de nous appuyer sur ELMO (*Emulator for power Leakage for the M0*) [14], un outil permettant de simuler la consommation de courant d'un programme exécuté sur un processeur ARM Cortex M0. Cet outil prend en entrée un code compilé pour un tel processeur et retourne en sortie une simulation de la consommation de courant liée à l'exécution du code. Plus précisément, il retourne, pour chaque instruction assembleur, une valeur décimale représentative de l'énergie consommée par celle-ci, et l'ensemble de ces valeurs nous donne la trace de la consommation.

Mais il est irréaliste d'imaginer étudier tous les points de la trace, car il y en a trop et certains ne sont pas intéressants (car ils ne dépendent pas beaucoup du secret). C'est pourquoi il faut restreindre l'analyse. Certains types d'instructions fournissent plus d'informations que d'autre. Pour pouvoir choisir quel type d'instructions exploiter, j'ai étudié le fonctionnement d'ELMO.

C'est à ce moment-là que j'ai remarqué que la multiplication est une instruction très intéressante, car il est possible d'avoir une valeur (quasiment) fixe pour la consommation de courant lors de la multiplication de deux valeurs a et b fixées. Bien entendu, dans un scénario réaliste d'attaque, nous n'avons pas accès à

cette valeur précise. En effet, dans les processeurs, il faudrait ajouter le bruit thermique gaussien qui est indépendant de cette valeur, bruit dont nous connaissons la variance σ_e^2 ($\sigma_e = 0.0045$) et qui correspond à un rapport signal sur bruit d'environ 4. L'utilitaire ELMO fournit une trace non bruitée mais, pour simuler une attaque réaliste, nous allons artificiellement ajouter ce bruit gaussien, ce qui est usuellement fait dans la littérature.

Rappelons tout de même que ELMO ne reflète pas tous les composants possibles. Cet outil simule un processeur ARM Cortex M0, l'attaque montée est donc spécifiquement mise au point contre un tel processeur.

Pour pouvoir utiliser ELMO afin de simuler l'attaque, j'ai récupéré l'implémentation de référence de Kyber (celle qui a été soumise pour le processus de standardisation du NIST). Cette implémentation était en language C. J'ai encapsulé l'implémentation de la NTT dans un petit programme C, programme que j'ai ensuite compilé avec un compilateur spécifique à ARM. Puis, j'ai encapsulé l'utilitaire ELMO dans un module Python. Il ne me restait plus qu'à appeler ce module Python pour pouvoir générer les traces qui m'intéressaient.

4.2.3 Élaboration des gabarits

J'ai choisi d'attaquer les multiplications présentes dans l'algorithme de la NTT. En étudiant l'implémentation, j'ai pu constater que les multiplications vont par trois. En effet, dans une telle série, la première multiplication fait intervenir une valeur connue ω avec une variable inconnue dépendant du secret, variable dont le support est $\{0, ..., q-1\}$; le produit de cette multiplication est ensuite multiplié deux fois par des constantes. Ainsi, pour une valeur fixée a de la variable étudiée et pour un certain ω, j'ai pu établir la consommation de courant de ces trois multiplications ou, plus exactement, la consommation moyenne (car l'attaque exploite une information bruitée selon ELMO). Cela forme le « gabarit » de la consommation pour a et ω fixés.

Maintenant, il suffit d'établir les gabarits pour tous les ω et a possibles. Comme je l'ai dit précédemment, ω est une valeur connue, elle dépend de l'état dans lequel l'exécution de la NTT se trouve. ω peut valoir 128 valeurs différentes. Et pour un ω donné, la variable inconnue contient un élément de $\{0, ..., q-1\}$. Nous devons donc établir $128 \times q \approx 500\ 000$ gabarits contenant 3 valeurs chacun.

Comment exploiter ces gabarits ? L'attaque va nous fournir une trace de la consommation. Pour chaque triplet de multiplications faisant intervenir un ω donné, nous allons comparer la consommation réelle (simulée dans notre cas) avec l'ensemble des q gabarits associés à ω. Par maximum de vraisemblance, nous allons extrapoler la valeur a. C'est le principe des *attaques par gabarits* (*template attacks*) [9].

4.2.4 Utilisation des gabarits

Dans notre cas, nous n'allons pas directement extrapoler a. Le principe de maximum de vraisemblance nous fournit une distribution sur le support de a. Nous allons combiner l'ensemble des distributions de probabilité fournies par l'attaque par gabarits afin d'obtenir une distribution pour chaque coefficient du secret s. La complexité de l'attaque se situe dans la manière de combiner les différentes distributions sur les variables intermédiaires de la NTT pour fournir une distribution sur le secret.

Voici comment j'ai procédé pour combiner ces distributions.

La NTT de Kyber travaille sur un tableau de 256 cases et, à chaque itération de la boucle principale de la NTT, chaque case du tableau est modifiée une fois et une seule. L'attaque par gabarits nous fournit des distributions de probabilité **sur le contenu de ces cases pour chaque itération**.

Ma méthodologie a donc été la suivante. Nous connaissons la distribution *a priori* de la valeur initiale

de chaque case du tableau. Nous étudions alors les distributions observées (par gabarits) sur les cases pour la première itération de la boucle principale. Nous obtenons ainsi **une distribution *a posteriori* plus raffinée** pour chaque case du tableau, et donc l'entropie globale diminue. Ensuite, la NTT combine deux à deux les cases du tableau, et cela **augmente l'entropie globale**. En effet, si nous sommons le contenu d'une case pouvant prendre n valeurs avec le contenu d'autre case pouvant prendre m valeurs, nous pouvons obtenir jusqu'à $n \times m$ valeurs différentes. Ensuite, **nous passons à l'itération suivante, et nous procédons de la même manière.** En même temps, pour chaque valeur possible d'une case du tableau à l'itération courante, nous **gardons en mémoire** les coefficients nécessaires du secret pour obtenir la valeur en question, dans l'objectif d'affiner la connaissance de la distribution sur les différents coefficients du secret lors de chaque itération.

Avec cette méthodologie, le nombre de cas à considérer **explose très rapidement**. A chaque itération, le nombre de valeurs possibles pour une case passe au carré. Si, au début, le contenu d'une case peut prendre 5 valeurs, alors

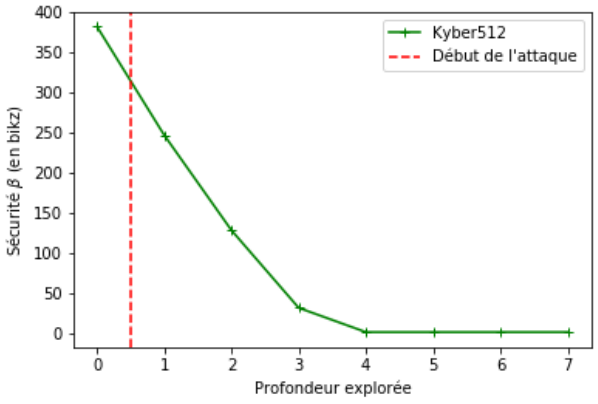
- à la fin de la première itération, il peut prendre jusqu'à 25 valeurs ;
- à la fin de la deuxième itération, il peut prendre jusqu'à 625 valeurs ;
- à la fin de la troisième itération, il peut prendre jusqu'à 390625 valeurs...

Pour avoir une chance que l'attaque aboutisse, il faut avoir des valeurs qui ont une faible probabilité d'apparition afin de pouvoir les ignorer sans trop impacter la probabilité de succès de l'attaque. **L'attaque consiste donc à trouver un juste milieu entre la divergence provoquée par la combinaison des cases deux à deux et une élimination modérée des possibilités**.

Une fois qu'on a obtenu une distribution pour chaque coefficient du secret, il ne reste plus qu'à les intégrer dans le framework *Leaky-LWE-Estimator* en tant qu'indices approximés, en faisant la modélisation que ces distributions sont gaussiennes (modélisation également réalisée dans l'attaque contre FrodoKEM).

4.2.5 Attaque contre la génération de clés

Comme décrit lors de la définition de la cible, la NTT est utilisée dans la génération des clés. Si nous appliquons l'attaque élaborée précédemment, nous obtenons le graphique suivant :



Ce graphique représente le coût de la résolution du problème LWE par BKZ (en bikz) en fonction de la profondeur explorée lors de l'exécution de la NTT. La boucle principale de la NTT fait 7 itérations. Explorer jusqu'à la profondeur i signifie que l'attaque a pris en compte les distributions des variables intermédiaires qui intervenaient dans les itérations 1, 2, ..., i.

A partir de la profondeur 4, le coût de la réduction par BKZ est 2 bikz, ce qui est le minimum. Cela signifie qu'en utilisant les variables des quatre premières itérations de la boucle de la NTT, la sécurité de Kyber est totalement cassée. Et donc, en résumé, cela signifie que nous pouvons restaurer le secret avec une unique trace de la consommation de courant.

Quelles sont les ressources requises pour l'attaque ? J'ai implémenté l'attaque en *SageMath* et je l'ai appliquée contre la génération de clé de Kyber512. Le temps d'exécution de celle-ci a été d'une quarantaine de secondes pour chaque NTT exécutée, et la NTT est exécutée 2 fois pour $\mathbf{z} \in R_q^2$ et 2 fois pour $\mathbf{e} \in R_q^2$ (pour Kyber512) en notant $\mathbf{s} := (\mathbf{e}, \mathbf{z})$ le secret.

Un résultat similaire a été trouvé dans l'article [16]. Avec une attaque par gabarits contre la NTT, les auteurs arrivent à restaurer le secret en une unique trace. Néanmoins, les deux attaques diffèrent dans les détails. Au lieu d'attaquer l'instruction de la multiplication, ils attaquent l'instruction qui charge en registre les variables ; leur attaque requiert moins de gabarits, mais nécessite plus de ressources pour être mise en oeuvre (hors réduction par BKZ) ; et ils utilisent des algorithmes de « propagation des convictions » pour combiner les distributions des variables intermédiaires.

Comme le font remarquer les auteurs de [16], il est possible d'utiliser cette attaque contre le chiffrement qui fait intervenir l'exécution de la NTT d'une clé secrète éphémère. Dans ce cas, le but de cette attaque est de restaurer le message qui a été chiffré. Et comme le chiffrement est utilisé dans la décapsulation de Kyber en tant que KEM, l'attaque permet de restaurer le secret commun établi si nous attaquons la décapsulation.

4.2.6 Attaque contre le déchiffrement

Comme décrit lors de la définition de la cible, la NTT est également utilisée dans le déchiffrement avec une entrée qui dépend du secret.

J'ai tenté de mettre en place une attaque similaire à l'attaque précédente, mais je n'y suis pas arrivé. Comme expliqué dans la description de l'attaque, celle-ci consiste à trouver un juste milieu entre la divergence provoquée par la combinaison des cases deux à deux et une élimination modérée des possibilités. Dans le cas de l'attaque contre la génération de clés, la distribution initiale des cases du tableau avait un support de 5 valeurs parce que nous appliquions la NTT directement sur la clé privée et que les coefficients de celle-ci appartiennent à $\{-2, ..., 2\}$. Mais hélas, ce n'est pas le cas dans ce nouveau contexte. Ici, nous avons une bien moins bonne connaissance de la distribution initiale. Nous devons quasiment considérer toutes les valeurs de $\{0, ..., q-1\}$ (pour Kyber, $q = 3329$) et le nombre de cas à considérer explose avant même que nous puissions essayer d'éliminer des possibilités.

Dans la littérature, il existe un article [17] qui attaque avec succès la NTT inverse utilisée dans le déchiffrement. Néanmoins, leur implémentation de la NTT était une implémentation en temps non constant, et donc l'attaque pouvait exploiter beaucoup plus de fuites d'information. Je n'ai pas trouvé d'article décrivant une attaque contre le déchiffrement avec une implémentation en temps constant de la NTT.

4.3 Attaque contre Dilithium

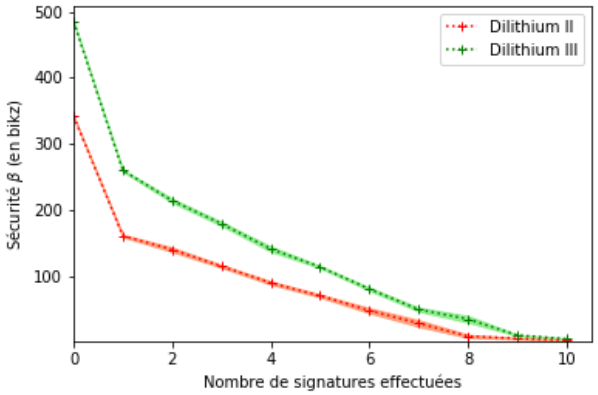
Dilithium est l'équivalent de Kyber pour les signatures. Ces deux cryptosystèmes se fondent sur le même problème LWE structuré. Alors, j'ai tenté d'adapter l'attaque contre Kyber pour avoir une attaque contre Dilithium.

La première remarque est que l'exécution de la NTT sur la clé secrète est présente **à chaque signature**. Cela signifie qu'il est facile pour l'attaquant d'avoir plusieurs traces de la même exécution de la NTT sur le secret et donc, puisque nous moyennons sur plusieurs traces, la comparaison avec les gabarits donnera

des résultats plus précis. Dans Kyber, la NTT du secret est effectuée une seule fois dans la génération des clés et est gardée en mémoire ; mais conserver la NTT du secret prend 8 fois plus de place que conserver juste le secret. C'est sûrement pour cette raison que les auteurs de Dilithium ont fait le choix de recalculer la NTT du secret à chaque fois.

D'un autre côté, le problème LWE est défini avec $q = 8\ 380\ 417$ alors que dans Kyber, $q = 3\ 329$. Cela a deux implications : le nombre de gabarits est nettement plus élevé, et combiner les distributions des variables intermédiaires est nettement plus coûteux. En pratique, il n'est possible de reproduire l'attaque **que sur les deux premières itérations** de la boucle principale de la NTT, avant que le nombre de cas explose.

Lorsque j'ai mis en pratique l'attaque, j'ai obtenu le résultat suivant :



Ce graphique montre le coût de la réduction par BKZ (en bikz) en fonction du nombre de signatures effectuées et exploitées. A la base, pour Dilithium III, la sécurité du problème LWE est de 485 bikz. Avec une signature, cela descend jusqu'à 260 bikz environ. Et au bout de 10 signatures, la sécurité du problème est proche de 2 bikz, Dilithium III est donc totalement cassé. Nous avons le même résultat pour Dilithium II : pour une sécurité de base de 342 bikz, la clé secrète peut facilement être restaurée avec une petite dizaine de signatures.

5 Conclusion

Durant ce stage, j'ai pu améliorer les performances du framework *Leaky-LWE-Estimator* à l'aide de la réduction de dimension et j'ai pu proposer de nouveaux types d'indices : les indices approximés dans un cadre général et les indices q-modulaires. Cela permet de mettre à la disposition de la communauté scientifique un outil un peu plus complet et performant.

En parallèle, les différentes attaques menées démontrent l'intérêt d'un framework complet de cryptanalyse sur les réseaux euclidiens, surtout avec la probable démocratisation des cryptosystèmes de ce type avec l'actuel processus de standardisation post-quantique du NIST.

J'invite le lecteur à revenir aux premières pages de ce rapport pour une discussion sur les potentielles perspectives des travaux de ce stage.

A Références

[1] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player et al. *Estimate all the LWE, NTRU schemes !* Cryptology ePrint Archive, Report 2018/331. https://eprint.iacr.org/2018/331. 2018.

[2] Martin R. Albrecht, Florian Göpfert, Fernando Virdia et Thomas Wunderer. *Revisiting the Expected Cost of Solving uSVP and Applications to LWE.* Cryptology ePrint Archive, Report 2017/815. https://eprint.iacr.org/2017/815. 2017.

[3] Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra et al. *NewHope.* URL : https://newhopecrypto.org/index.shtml. (accessed : 21.07.2020).

[4] Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Juliane Kramer, Patrick Longa et Jefferson E. Ricardini. *qTESLA.* URL : https://qtesla.org/. (accessed : 21.07.2020).

[5] Erdem Alkim, Joppe W. Bos, Léo Ducas, Patrick Longa, Ilya Mironov et al. *FrodoKEM.* URL : https://frodokem.org/. (accessed : 21.07.2020).

[6] Erdem Alkim, Léo Ducas, Thomas Pöppelmann et Peter Schwabe. *Post-quantum key exchange - a new hope.* Cryptology ePrint Archive, Report 2015/1092. https://eprint.iacr.org/2015/1092. 2015.

[7] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint et al. *Crystals-Kyber.* URL : https://pq-crystals.org/kyber/index.shtml. (accessed : 21.07.2020).

[8] Joppe W. Bos, Simon Friedberger, Marco Martinoli, Elisabeth Oswald et Martijn Stam. *Assessing the Feasibility of Single Trace Power Analysis of Frodo.* Cryptology ePrint Archive, Report 2018/687. https://eprint.iacr.org/2018/687. 2018.

[9] Suresh Chari, Josyula R. Rao et Pankaj Rohatgi. "Template Attacks". Dans : *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers.* T. 2523. Lecture Notes in Computer Science. Springer, 2002, p. 13-28. DOI : 10.1007/3-540-36400-5_3.

[10] Yuanmi Chen et Phong Q. Nguyen. "BKZ 2.0 : Better Lattice Security Estimates". Dans : *Advances in Cryptology – ASIACRYPT 2011.* Sous la dir. de Dong Hoon Lee et Xiaoyun Wang. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011, p. 1-20. ISBN : 978-3-642-25385-0.

[11] Dana Dachman-Soled, Léo Ducas, Huijing Gong et Mélissa Rossi. *LWE with Side Information : Attacks and Concrete Security Estimation.* Cryptology ePrint Archive, Report 2020/292. https://eprint.iacr.org/2020/292. 2020.

[12] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe et al. *Crystals-Dilithium.* URL : https://pq-crystals.org/dilithium/index.shtml. (accessed : 21.07.2020).

[13] Li-Ping Liu. *Linear Transformation of Multivariate Normal Distribution : Marginal, Joint and Posterior.* URL : http://www.cs.columbia.edu/~liulp/pdf/linear_normal_dist.pdf. (accessed : 21.07.2020).

[14] David McCann, Elisabeth Oswald et Carolyn Whitnall. "Towards Practical Tools for Side Channel Aware Software Engineering : 'Grey Box' Modelling for Instruction Leakages". Dans : *26th USENIX Security Symposium (USENIX Security 17).* Vancouver, BC : USENIX Association, août 2017, p. 199-216. ISBN : 978-1-931971-40-9. URL : https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/mccann.

[15] NIST. *Post-Quantum cryptography standardization.* URL : https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/call-for-proposals. (accessed : 21.07.2020).

[16] Peter Pessl et Robert Primas. *More Practical Single-Trace Attacks on the Number Theoretic Transform*. Cryptology ePrint Archive, Report 2019/795. https://eprint.iacr.org/2019/795. 2019.

[17] Robert Primas, Peter Pessl et Stefan Mangard. *Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption*. Cryptology ePrint Archive, Report 2017/594. https://eprint.iacr.org/2017/594. 2017.

B Réduction de dimension

Théorème 6. *Soit une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Il est possible de transformer $\mathcal{I}$ en une instance $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ dont le réseau est de rang maximal.*

Si nous avons $\boldsymbol{\Gamma}$ tel que $\Lambda = \boldsymbol{\Gamma}\Lambda'$ avec Λ' de rang maximal et $\mathrm{rank}(\Lambda) = \mathrm{rank}(\Lambda')$, alors

$$\begin{aligned}\mathbf{D}' &\leftarrow \boldsymbol{\Gamma}^T\mathbf{D},\\ \boldsymbol{\mu}' &\leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\mu},\\ \boldsymbol{\Sigma}' &\leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1},\end{aligned}$$

où $\mathbf{D}$ est une base duale de Λ et $\mathbf{D}'$ et la base duale correspondante de Λ'.

Et alors,

$$\begin{aligned}\mathrm{Vol}(\Lambda') &= \det(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1/2}\cdot\mathrm{Vol}(\Lambda),\\ \mathrm{rdet}(\boldsymbol{\Sigma}' + \boldsymbol{\mu}'\cdot\boldsymbol{\mu}'^T) &= \det(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\cdot\mathrm{rdet}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\cdot\boldsymbol{\mu}^T),\\ \mathrm{Vol}(\mathcal{I}') &= \mathrm{Vol}(\mathcal{I}).\end{aligned}$$

De plus, voici quelques propriétés qu'un tel $\boldsymbol{\Gamma}$ vérifie :

$$\begin{aligned}&\forall \text{ vecteur propre } \mathbf{v} \text{ associé à } 0 \text{ de } \sqrt{\boldsymbol{\Sigma} + \boldsymbol{\mu}\cdot\boldsymbol{\mu}^T}, \boldsymbol{\Gamma}^T\mathbf{v} = 0.\\ &\Lambda^* \subset \mathrm{Span}(\Lambda) \subset E_1[\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T],\\ &\qquad \text{donc, } \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\mathbf{D} = \mathbf{D} \text{ et } \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\mathbf{B} = \mathbf{B},\end{aligned}$$

en notant $E_1[\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T]$ le sous-espace propre associé à 0 de $\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T$.

Démonstration.

Supposons que nous avons $\boldsymbol{\Gamma}$ tel que $\Lambda = \boldsymbol{\Gamma}\Lambda'$ avec Λ' de rang maximal et $\mathrm{rank}(\Lambda) = \mathrm{rank}(\Lambda')$.

- **Calcul de $\mathbf{D}'$.**
  Si $\mathbf{B}'$ est une base (carrée) de Λ', alors $\boldsymbol{\Gamma}\mathbf{B}'$ est une base de Λ.
  Nous avons

$$\begin{aligned}\mathbf{D} &= \mathbf{B}\cdot(\mathbf{B}^T\mathbf{B})^{-1}\\ &= (\boldsymbol{\Gamma}\mathbf{B}')\cdot((\boldsymbol{\Gamma}\mathbf{B}')^T(\boldsymbol{\Gamma}\mathbf{B}'))^{-1}\\ &= \boldsymbol{\Gamma}\mathbf{B}'\cdot(\mathbf{B}'^T\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\mathbf{B}')^{-1}\\ &= \boldsymbol{\Gamma}\mathbf{B}'\mathbf{B}'^{-1}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\mathbf{B}'^{-T}\\ &= \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\mathbf{D}'.\end{aligned}$$

  Et donc,

$$\mathbf{D}' = \boldsymbol{\Gamma}^T\cdot\mathbf{D}.$$

La dernière implication peut ne pas être réciproque. Pour avoir une équivalence, nous avons besoin de rajouter un détail.

$$\Lambda = \mathbf{\Gamma}\Lambda' \text{ ssi } ((\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\Lambda = \Lambda' \text{ et } \mathrm{Span}(\Lambda) \subset E_1[\mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T]).$$

Pour le sens direct, l'égalité est triviale (nous ajoutons juste $\mathbf{\Gamma}^T$ et nous inversons la matrice devant Λ'). Donc parlons de l'inclusion. Prenons un $\mathbf{y} \in \mathrm{Span}(\Lambda)$, alors il existe $\mathbf{x} \in \mathrm{Span}(\Lambda')$ tel que $\mathbf{y} = \mathbf{\Gamma}\mathbf{x}$. Nous en déduisons

$$\mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{y} = \mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}(\mathbf{\Gamma}^T\mathbf{\Gamma})\mathbf{x} = \mathbf{\Gamma}\mathbf{x} = \mathbf{y}.$$

Donc, $\mathbf{y} \in E_1[\mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T]$.
Pour le sens indirect, nous avons

$$\begin{aligned} \mathbf{\Gamma}\Lambda' &= \mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\Lambda && \text{(avec l'égalité)} \\ &= \Lambda. && \text{(avec l'inclusion)} \end{aligned}$$

- **Calcul de $\boldsymbol{\mu}'$.**
  Le secret $\mathbf{s}$ dans le réseau Λ correspond au secret $\mathbf{s}'$ dans le réseau Λ' avec $\mathbf{s} = \mathbf{\Gamma}\mathbf{s}'$. En passant à l'espérance, nous obtenons $\boldsymbol{\mu} = \mathbf{\Gamma}\boldsymbol{\mu}'$, et donc

$$\mathbf{\Gamma}^T\boldsymbol{\mu} = \mathbf{\Gamma}^T\mathbf{\Gamma}\boldsymbol{\mu}'$$
$$\boldsymbol{\mu}' = (\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\boldsymbol{\mu}.$$

- **Calcul de $\mathbf{\Sigma}'$.**
  Raisonnons comme pour la moyenne.

$$\begin{aligned} \mathbf{\Sigma}_{i,j} &= \mathrm{Cov}(\mathbf{s}_i, \mathbf{s}_j) = \mathrm{Cov}((\mathbf{\Gamma}\mathbf{s}')_i, (\mathbf{\Gamma}\mathbf{s}')_j) \\ &= \mathrm{Cov}(\sum_k \mathbf{\Gamma}_{i,k}\mathbf{s}'_k, \sum_l \mathbf{\Gamma}_{j,l}\mathbf{s}'_l) \\ &= \sum_k \mathbf{\Gamma}_{i,k}(\sum_l \mathrm{Cov}(\mathbf{s}'_k, \mathbf{s}'_l)\mathbf{\Gamma}_{j,l}) \\ &= \sum_k \mathbf{\Gamma}_{i,k}(\sum_l \mathbf{\Sigma}'_{k,l}(\mathbf{\Gamma}^T)_{l,j}) \\ &= \sum_k \mathbf{\Gamma}_{i,k}(\mathbf{\Sigma}'\mathbf{\Gamma}^T)_{k,j} \\ &= (\mathbf{\Gamma}\mathbf{\Sigma}'\mathbf{\Gamma}^T)_{i,j} \end{aligned}$$

  Donc, nous avons $\mathbf{\Sigma} = \mathbf{\Gamma}\mathbf{\Sigma}'\mathbf{\Gamma}^T$, et alors

$$\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{\Sigma}'\mathbf{\Gamma}^T\mathbf{\Gamma} = \mathbf{\Gamma}^T\mathbf{\Sigma}\mathbf{\Gamma}$$
$$\mathbf{\Sigma}' = (\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{\Sigma}\mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}.$$

- **Calcul de $\mathrm{Vol}(\Lambda')$.**
  Notons $\mathbf{B}'$ une base de Λ'. Alors, $\mathbf{\Gamma}\mathbf{B}'$ est une base de Λ.

$$\begin{aligned} \mathrm{Vol}(\Lambda) &= \sqrt{\det((\mathbf{\Gamma}\mathbf{B}')^T(\mathbf{\Gamma}\mathbf{B}'))} \\ &= \sqrt{\det(\mathbf{B}'^T\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{B}')} \\ &= \sqrt{\det(\mathbf{B}')^2 \cdot \det(\mathbf{\Gamma}^T\mathbf{\Gamma})} \\ &= \mathrm{Vol}(\Lambda') \cdot \sqrt{\det(\mathbf{\Gamma}^T\mathbf{\Gamma})} \end{aligned}$$

- **Calcul de** $\mathrm{rdet}(\boldsymbol{\Sigma}' + \boldsymbol{\mu}' \cdot \boldsymbol{\mu}'^T)$

  Commençons par un petit calcul préliminaire. Prenons un vecteur propre $\mathbf{v}$ de $\sqrt{\boldsymbol{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T}$ associé à la valeur propre 0.

$$\mathbf{v}^T(\boldsymbol{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T)\mathbf{v} = (\sqrt{\boldsymbol{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T}\mathbf{v})^T(\sqrt{\boldsymbol{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T}\mathbf{v}) = 0$$

$$\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} + \langle \mathbf{v}, \boldsymbol{\mu} \rangle^2 = 0$$

$$0 \leq \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} = -\langle \mathbf{v}, \boldsymbol{\mu} \rangle^2 \leq 0$$

  Donc, $\mathbf{v}^T\boldsymbol{\Gamma}\mathbf{v} = 0$ et $\langle \mathbf{v}, \boldsymbol{\mu} \rangle = 0$. Puisque $\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} = 0$,

$$\boldsymbol{\Gamma}\Lambda' = \Lambda \subset \{\mathbf{x} \in \mathrm{Span}(\Lambda) : \langle \mathbf{x}, \mathbf{v} \rangle = \langle \boldsymbol{\mu}, \mathbf{v} \rangle = 0\}.$$

  Alors,

$$\forall \mathbf{x}' \in \Lambda', \langle \mathbf{x}', \boldsymbol{\Gamma}^T\mathbf{v} \rangle = \langle \boldsymbol{\Gamma}\mathbf{x}', \mathbf{v} \rangle = 0.$$

  Puisque Λ' est de rang maximal (par hypothèse), nous avons

$$\forall \mathbf{x}' \in \mathbb{R}^{\dim(\Lambda')}, \langle \mathbf{x}', \boldsymbol{\Gamma}^T\mathbf{v} \rangle = 0.$$

  Donc,

$$\boldsymbol{\Gamma}^T\mathbf{v} = 0.$$

  Conclusion du calcul préliminaire :

$$\forall \text{ vecteur propre } \mathbf{v} \text{ associé à } 0 \text{ de } \sqrt{\boldsymbol{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T},\ \boldsymbol{\Gamma}^T\mathbf{v} = 0.$$

  Notons $\mathbf{M}$ pour $\boldsymbol{\Sigma} + \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T$ et $\mathbf{M}'$ pour $\boldsymbol{\Sigma}' + \boldsymbol{\mu}' \cdot \boldsymbol{\mu}'^T$. De plus, notons $\mathbf{V}$ une base orthogonale du sous-espace de $\sqrt{\mathbf{M}}$ associé à la valeur propre 0. Grâce au calcul préliminaire, nous savons que

$$\boldsymbol{\Gamma}^T\mathbf{V} = 0.$$

  Construisons une base **B orthogonale** de vecteurs propres en ajoutant d'autres vecteurs propres $\mathbf{W}$ à $\mathbf{V}$, $\mathbf{B} = (\mathbf{V}|\mathbf{W})$, pour avoir (théorème spectral) :

$$\sqrt{\mathbf{M}} = (\mathbf{V}|\mathbf{W}) \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \lambda_1 & & \\ & & & & \ddots & \\ & & & & & \lambda_d \end{pmatrix} \begin{pmatrix} \mathbf{V}^T \\ \hline \mathbf{W}^T \end{pmatrix}.$$

  Et donc,

$$\mathbf{M} = (\mathbf{V}|\mathbf{W}) \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \lambda_1^2 & & \\ & & & & \ddots & \\ & & & & & \lambda_d^2 \end{pmatrix} \begin{pmatrix} \mathbf{V}^T \\ \hline \mathbf{W}^T \end{pmatrix}.$$

Et si nous ajoutons $\mathbf{\Gamma}$,

$$\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma} = (\mathbf{0}|\mathbf{\Gamma}^T\mathbf{W}) \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \lambda_1^2 & & \\ & & & & \ddots & \\ & & & & & \lambda_d^2 \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \hline \mathbf{W}^T\mathbf{\Gamma} \end{pmatrix} = \mathbf{\Gamma}^T\mathbf{W} \begin{pmatrix} \lambda_1^2 & & \\ & \ddots & \\ & & \lambda_d^2 \end{pmatrix} \mathbf{W}^T\mathbf{\Gamma}.$$

Notons $\mathbf{S} := \mathbf{\Gamma}^T\mathbf{\Gamma}$, alors

$$\sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} = \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{W} \begin{pmatrix} \lambda_1^2 & & \\ & \ddots & \\ & & \lambda_d^2 \end{pmatrix} \mathbf{W}^T\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1}.$$

Si nous notons $\mathbf{O} := \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{W}$, nous avons

$$\begin{aligned} \mathbf{O}\mathbf{O}^T &= \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{W}\mathbf{W}^T\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} \\ &= \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} \\ &= \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} - \sqrt{\mathbf{S}}^{-T}(\mathbf{\Gamma}^T\mathbf{V})(\mathbf{V}^T\mathbf{\Gamma})\sqrt{\mathbf{S}}^{-1} \\ &= \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} \\ &= \sqrt{\mathbf{S}}^{-T}\mathbf{S}\sqrt{\mathbf{S}}^{-1} \\ &= \mathbf{I}. \end{aligned}$$

Donc, $\mathbf{O} = \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{W}$ est une base orthogonale. Plus précisément, c'est la matrice de transition orthogonale pour diagonaliser $\sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1}$.
Donc,

$$\mathrm{rdet}(\sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1}) = \Pi_{i=1}^d\lambda_i^2 = \mathrm{rdet}(\mathbf{M}).$$

Mais,

$$\begin{aligned} \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} &= \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T(\mathbf{\Sigma} + \boldsymbol{\mu}\cdot\boldsymbol{\mu}^T)\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} \\ &= \sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\mathbf{\Sigma}\mathbf{\Gamma}\sqrt{\mathbf{S}}^{-1} + (\sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\boldsymbol{\mu})\cdot(\sqrt{\mathbf{S}}^{-T}\mathbf{\Gamma}^T\boldsymbol{\mu})^T \\ &= \sqrt{\mathbf{S}}^T\mathbf{\Sigma}'\sqrt{\mathbf{S}} + (\sqrt{\mathbf{S}}^T\boldsymbol{\mu}')\cdot(\sqrt{\mathbf{S}}^T\boldsymbol{\mu}')^T \\ &= \sqrt{\mathbf{S}}^T\mathbf{M}'\sqrt{\mathbf{S}}. \end{aligned}$$

Donc,

$$\mathrm{rdet}(\sqrt{\mathbf{S}}^T\mathbf{M}'\sqrt{\mathbf{S}}) = \mathrm{rdet}(\mathbf{M}).$$

Le rang de $\mathbf{M}'$ est maximal, parce que nous avons (propriété d'un DBDD)

$$\mathrm{rank}(\mathbf{M}') = \mathrm{rank}(\Lambda')$$

et que Λ' est de rang maximal. Alors, pour une matrice de rang maximal / inversible, le déterminant restreint est équivalent au déterminant standard, donc

$$\begin{aligned}\mathrm{rdet}(\sqrt{\mathbf{S}}^T\mathbf{M}'\sqrt{\mathbf{S}}) &= \det(\sqrt{\mathbf{S}}^T\mathbf{M}'\sqrt{\mathbf{S}}) \\ &= \det(\sqrt{\mathbf{S}}^T)\cdot\det(\mathbf{M}')\cdot\det(\sqrt{\mathbf{S}}) \\ &= \det(\mathbf{S})\cdot\det(\mathbf{M}') \\ &= \det(\mathbf{S})\cdot\mathrm{rdet}(\mathbf{M}').\end{aligned}$$

Et par conséquent,

$$\mathrm{rdet}(\mathbf{M}') = \frac{1}{\det(\mathbf{S})}\cdot\mathrm{rdet}(\mathbf{M})$$

i.e.

$$\mathrm{rdet}(\mathbf{\Sigma}' + \boldsymbol{\mu}'\cdot\boldsymbol{\mu}'^T) = \frac{1}{\det(\mathbf{\Gamma}^T\mathbf{\Gamma})}\cdot\mathrm{rdet}(\mathbf{\Sigma}+\boldsymbol{\mu}\cdot\boldsymbol{\mu}^T).$$

- **Calcul de** $\mathrm{Vol}(\mathcal{I}')$**.**

$$\begin{aligned}\mathrm{Vol}(\mathcal{I}') &= \mathrm{rdet}(\mathbf{\Sigma}' + \boldsymbol{\mu}'\cdot\boldsymbol{\mu}'^T)^{-1/2}\cdot\mathrm{Vol}(\Lambda') \\ &= (\det(\mathbf{\Gamma}^T\mathbf{\Gamma})^{1/2}\cdot\mathrm{rdet}(\mathbf{\Sigma}+\boldsymbol{\mu}\cdot\boldsymbol{\mu}^T)^{-1/2})\cdot(\det(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1/2}\cdot\mathrm{Vol}(\Lambda)) \\ &= (\det(\mathbf{\Gamma}^T\mathbf{\Gamma})^{1/2}\cdot\det(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1/2})\cdot(\mathrm{rdet}(\mathbf{\Sigma}+\boldsymbol{\mu}\cdot\boldsymbol{\mu}^T)^{-1/2}\cdot\mathrm{Vol}(\Lambda)) \\ &= \mathrm{Vol}(\mathcal{I})\end{aligned}$$

Voici ainsi démontrés tous les éléments du théorème. □

B.1 La matrice de substitution standard, un choix intéressant pour $\mathbf{\Gamma}$

Si nous devions réaliser une réduction sans aide pour trouver un $\mathbf{\Gamma}$ qui fonctionne, nous devrions réaliser une mise sous forme d'Hermite, un calcul de déterminant et une inversion de matrice (pour avoir le déterminant et l'inverse de $\mathbf{\Gamma}^T\mathbf{\Gamma}$), et ce serait réellement coûteux.

Pour éviter ce problème, nous pouvons utiliser le vecteur $\bar{\mathbf{v}}$ d'un indice parfait. Avec celui-ci, il est facile de construire un $\mathbf{\Gamma}$ qui fonctionne, de la forme suivante

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{v}_1^T & -\mathbf{v}_2^T \\ \mathbf{0} & \mathbf{I}\end{pmatrix}$$

où $\mathbf{v}_1$ et $\mathbf{v}_2$ sont définis par la relation $\bar{\mathbf{v}} = \bar{\mathbf{v}}_p\cdot(\mathbf{v}_1^T, 1, \mathbf{v}_2^T)$, p étant l'index d'un coefficient non nul de $\bar{\mathbf{v}}$. En construisant directement cette matrice, nous évitons la mise sous forme d'Hermite, mais nous devons toujours calculer le déterminant et l'inverse de $\mathbf{\Gamma}^T\mathbf{\Gamma}$... sauf qu'il existe heureusement une expression littérale de $\det(\mathbf{\Gamma}^T\mathbf{\Gamma})$ et de $(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}$.

Nous avons

$$\mathbf{\Gamma}^T\mathbf{\Gamma} = \begin{pmatrix} \mathbf{I}+\mathbf{v}_1\mathbf{v}_1^T & \mathbf{v}_1\mathbf{v}_2^T \\ \mathbf{v}_2\mathbf{v}_1^T & \mathbf{I}+\mathbf{v}_2\mathbf{v}_2^T\end{pmatrix}.$$

Théorème 7. *Prenons $\mathbf{\Gamma}$ de la forme décrite précédemment. Notons $\mathbf{v}^T = (\mathbf{v}_1^T, 1, \mathbf{v}_2^T)$. Alors, le déterminant de $\mathbf{\Gamma}^T\mathbf{\Gamma}$ est*

$$\det(\mathbf{\Gamma}^T\mathbf{\Gamma}) = \|\mathbf{v}\|^2 = 1 + \|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2.$$

Et son inverse est

$$(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1} = \begin{pmatrix} \mathbf{I} - \frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_1^T & -\frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_2^T \\ -\frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_2\mathbf{v}_1^T & I - \frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_2\mathbf{v}_2^T\end{pmatrix}.$$

Lemme 1 (Lemme préliminaire)**.** *Soit* $\mathbf{v}$ *un vecteur. Alors,* $\mathbf{I} + \mathbf{v}\mathbf{v}^T$ *est inversible avec*

$$\det(\mathbf{I} + \mathbf{v}\mathbf{v}^T) = 1 + \|\mathbf{v}\|^2,$$

$$(\mathbf{I} + \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{I} - \frac{1}{1 + \|\mathbf{v}\|^2}\mathbf{v}\mathbf{v}^T.$$

Démonstration du lemme. Prenons un vecteur propre $\mathbf{u} \neq \mathbf{0}$ de $\mathbf{I} + \mathbf{v}\mathbf{v}^T$,

$$(\mathbf{I} + \mathbf{v}\mathbf{v}^T)\mathbf{u} = \lambda\mathbf{u}.$$

Donc,

$$(\lambda - 1)\mathbf{u} = \langle \mathbf{u}, \mathbf{v} \rangle \mathbf{v}.$$

Et alors, si $\lambda \neq 1$ (la valeur propre 1 n'a pas d'impact sur le déterminant), alors $(\lambda - 1) \neq 0$. Donc,

$$\mathbf{u} = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\lambda - 1}\mathbf{v}.$$

$\mathbf{u}$ est colinéaire avec $\mathbf{v}$, et $(\lambda - 1)\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle \langle \mathbf{v}, \mathbf{v} \rangle$, *i.e.*

$$\lambda = 1 + \|\mathbf{v}\|^2.$$

Donc, le sous-espace propre associé à la valeur propre 1 est un hyperplan si $\mathbf{v} \neq 0$, et la dernière valeur propre est $1 + \|\mathbf{v}\|^2$. Donc,

$$\det(\mathbf{I} + \mathbf{v}\mathbf{v}^T) = 1 + \|\mathbf{v}\|^2.$$

Pour l'inverse de $\mathbf{I} + \mathbf{v}\mathbf{v}^T$,

$$\begin{aligned}(\mathbf{I} + \mathbf{v}\mathbf{v}^T)(\mathbf{I} - \frac{1}{1 + \|\mathbf{v}\|^2}\mathbf{v}\mathbf{v}^T) &= \mathbf{I} + \mathbf{v}\mathbf{v}^T - \frac{1}{1 + \|\mathbf{v}\|^2}\mathbf{v}\mathbf{v}^T - \frac{\|\mathbf{v}\|^2}{1 + \|\mathbf{v}\|^2}\mathbf{v}\mathbf{v}^T \\ &= \mathbf{I} + (1 - \frac{1 + \|\mathbf{v}\|^2}{1 + \|\mathbf{v}\|^2})\mathbf{v} = \mathbf{I}.\end{aligned}$$

Ici s'achève la démonstration du lemme. □

Démonstration du théorème. Calculons $(\mathbf{\Gamma}^T\mathbf{\Gamma})(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}$.

$$(\mathbf{\Gamma}^T\mathbf{\Gamma})(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$$

avec

$$\begin{aligned}\mathbf{A} &= (\mathbf{I} + \mathbf{v}_1\mathbf{v}_1^T)(\mathbf{I} - \frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_1^T) + (\mathbf{v}_1\mathbf{v}_2^T)(-\frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_2\mathbf{v}_1^T) \\ &= \mathbf{I} + \mathbf{v}_1\mathbf{v}_1^T - \frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_1^T - \frac{\|\mathbf{v}_1\|^2}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_1^T - \frac{\|\mathbf{v}_2\|^2}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_1^T \\ &= \mathbf{I} + (1 - \frac{1 + \|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2}{\|\mathbf{v}\|^2})\mathbf{v}_1\mathbf{v}_1^T = \mathbf{I}, \\ \mathbf{D} &= \mathbf{I}, \qquad \text{(même calcul que pour } \mathbf{A} \text{ en échangeant } \mathbf{v}_1 \text{ et } \mathbf{v}_2)\end{aligned}$$

$$\begin{aligned}
\mathbf{B} &= (\mathbf{I} + \mathbf{v}_1\mathbf{v}_1^T)(-\frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_2^T) + (\mathbf{v}_1\mathbf{v}_2^T)(\mathbf{I} - \frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_2\mathbf{v}_2^T) \\
&= -\frac{1}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_2^T - \frac{\|\mathbf{v}_1\|^2}{\|\mathbf{v}\|^2}\mathbf{v}_1\mathbf{v}_2^T + \mathbf{v}_1\mathbf{v}_2^T - \frac{\|\mathbf{v}_2\|^2}{\|\mathbf{v}\|^2}\mathbf{v}_2\mathbf{v}_2^T \\
&= (1 - \frac{1 + \|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2}{\|\mathbf{v}\|^2})\mathbf{v}_1\mathbf{v}_2^T = \mathbf{0}, \\
\mathbf{C} &= \mathbf{0}, \qquad \text{(même calcul que pour } \mathbf{B} \text{ en échangeant } \mathbf{v}_1 \text{ et } \mathbf{v}_2)
\end{aligned}$$

Et, pour le déterminant, nous avons la relation suivante :

$$(\mathbf{\Gamma}^T\mathbf{\Gamma}) \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\frac{1}{1+\|\mathbf{v}_2\|^2}\mathbf{v}_2\mathbf{v}_1^T & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{I} + \frac{1}{1+\|\mathbf{v}_2\|^2}\mathbf{v}_1\mathbf{v}_1^T & * \\ 0 & \mathbf{I} + \mathbf{v}_2\mathbf{v}_2^T \end{pmatrix}.$$

Donc,

$$\begin{aligned}
\det(\mathbf{\Gamma}^T\mathbf{\Gamma}) &= \det(\mathbf{I} + \frac{1}{1 + \|\mathbf{v}_2\|^2}\mathbf{v}_1\mathbf{v}_1^T) \cdot \det(\mathbf{I} + \mathbf{v}_2\mathbf{v}_2^T) \\
&= (1 + \frac{1}{1 + \|\mathbf{v}_2\|^2}\|\mathbf{v}_1\|^2)(1 + \|\mathbf{v}_2\|^2) \\
&= 1 + \|\mathbf{v}_2\|^2 + \|\mathbf{v}_1\|^2 = \|\mathbf{v}\|^2.
\end{aligned}$$

Ici s'achève la démonstration du théorème. □

C Intégration des indices approximés

Preuve de la formule généralisée des indices approximés. Nous connaissons la distribution *a posteriori* $\mathcal{D}_{ap}$ de $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + e$, où e représente du bruit qui suit une distribution gaussienne $\mathcal{N}_1(0, \sigma_e^2)$, indépendante de $\mathbf{s}$. Notons μ_{ap} la moyenne de la distribution et σ_{ap}^2 sa variance.

De plus, notons f_{ap} sa fonction de densité de probabilité.

$$\begin{aligned}
\mathbb{E}[\bar{\mathbf{s}}] &= \mathbb{E}[\mathbb{E}[\bar{\mathbf{s}} | \langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + e]] \\
&= \int_{y \in \mathbb{R}} \mathbb{E}[\bar{\mathbf{s}} | \langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + e = y] \cdot f_{ap}(y) dy
\end{aligned}$$

D'après [13], nous avons

$$\begin{aligned}
\mathbb{E}[\bar{\mathbf{s}} | \langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + e = y] &= \boldsymbol{\mu} + \frac{y - \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2}\mathbf{\Sigma}\bar{\mathbf{v}} \\
&= \boldsymbol{\mu} - \frac{\mathbf{\Sigma}\bar{\mathbf{v}}\langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2} + y \cdot \frac{\mathbf{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2} \\
&= \left(\mathbf{I} - \frac{\mathbf{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2}\right)\boldsymbol{\mu} + y \cdot \frac{\mathbf{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2}.
\end{aligned}$$

Définissons la pseudo-projection $\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}} := \frac{\mathbf{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2}$.

$$\begin{aligned}
\mathbb{E}[\bar{\mathbf{s}} | \langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + e = y] &= \left(\mathbf{I} - \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}\right)\boldsymbol{\mu} + y \cdot \frac{\mathbf{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2} \\
&= \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu} + y \cdot \frac{\mathbf{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\mathbf{\Sigma}\bar{\mathbf{v}} + \sigma_e^2}
\end{aligned}$$

Donc,

$$\begin{aligned}\mathbb{E}[\bar{\mathbf{s}}] &= \int_{y\in\mathbb{R}} \left(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu} + y\cdot\frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right)\cdot f_{ap}(y)dy \\ &= \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu} + \left(\int_{y\in\mathbb{R}} y f_{ap}(y)dy\right)\cdot\frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} \\ &= \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu} + \mu_{ap}\cdot\frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}.\end{aligned}$$

Nous avons

$$\boldsymbol{\mu}' = \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu} + \mu_{ap}\cdot\frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}$$

où $\boldsymbol{\mu}'$ est défini dans l'énoncé du théorème.

Et maintenant, qu'avons-nous pour la matrice de covariance ?
Prenons $(i,j)\in\{1,...,d\}^2$.

$$\begin{aligned}\mathrm{Cov}(\bar{\mathbf{s}}_i,\bar{\mathbf{s}}_j) &= \mathbb{E}[\bar{\mathbf{s}}_i\bar{\mathbf{s}}_j] - \mathbb{E}[\bar{\mathbf{s}}_i]\cdot E[\bar{\mathbf{s}}_j] \\ &= \mathbb{E}[\mathbb{E}[\bar{\mathbf{s}}_i\bar{\mathbf{s}}_j|\langle\bar{\mathbf{s}},\bar{\mathbf{v}}\rangle+e]] - \mathbb{E}[\bar{\mathbf{s}}_i]\cdot E[\bar{\mathbf{s}}_j]\end{aligned}$$

D'après le calcul précédent,

$$\mathbb{E}[\bar{\mathbf{s}}_i] = (\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i + \mu_{ap}\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}.$$

Même relation pour $\mathbb{E}[\bar{\mathbf{s}}_j]$. Et alors, nous avons la relation suivante :

$$\hat{\boldsymbol{\Sigma}}_{i,j} = \mathbb{E}[\bar{\mathbf{s}}_i\bar{\mathbf{s}}_j|\langle\bar{\mathbf{s}},\bar{\mathbf{v}}\rangle+e=y] - \mathbb{E}[\bar{\mathbf{s}}_i|\langle\bar{\mathbf{s}},\bar{\mathbf{v}}\rangle+e=y]\cdot\mathbb{E}[\bar{\mathbf{s}}_j|\langle\bar{\mathbf{s}},\bar{\mathbf{v}}\rangle+e=y]$$

où $\hat{\boldsymbol{\Sigma}}$ est la matrice de covariance définie dans [13]. Donc,

$$\mathbb{E}[\bar{\mathbf{s}}_i\bar{\mathbf{s}}_j|\langle\bar{\mathbf{s}},\bar{\mathbf{v}}\rangle+e=y] = \hat{\boldsymbol{\Sigma}}_{i,j} + \left((\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i + y\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right)\left((\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j + y\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right).$$

Et donc ($\hat{\boldsymbol{\Sigma}}_{i,j}$ indépendante de y),

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\bar{\mathbf{s}}_i\bar{\mathbf{s}}_j|\langle\bar{\mathbf{s}},\bar{\mathbf{v}}\rangle+e]] &= \hat{\boldsymbol{\Sigma}}_{i,j} + \int_{y\in\mathbb{R}}\left((\hat{\Pi}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i + y\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right)\left((\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j + y\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right)f_{ap}(y)dy \\ &= \hat{\boldsymbol{\Sigma}}_{i,j} + (\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j + \mu_{ap}\cdot\frac{(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + \mu_{ap}\cdot\frac{(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} \\ &\quad + \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\int_{y\in\mathbb{R}} y^2 f_{ap}(y)dy \\ &= \hat{\boldsymbol{\Sigma}}_{i,j} + (\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j + \mu_{ap}\cdot\frac{(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + \mu_{ap}\cdot\frac{(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} \\ &\quad + \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}(\sigma_{ap}^2+\mu_{ap}^2).\end{aligned}$$

Et

$$\begin{aligned}\mathbb{E}[\bar{\mathbf{s}}_i]\cdot\mathbb{E}[\bar{\mathbf{s}}_j] &= \left((\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i + \mu_{ap}\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right)\left((\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j + \mu_{ap}\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\right)\\ &= (\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j + \mu_{ap}\cdot\frac{(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + \mu_{ap}\cdot\frac{(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\boldsymbol{\mu})_j(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2}\\ &\quad + \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}(\mu_{ap}^2).\end{aligned}$$

Fort heureusement, beaucoup de termes disparaissent lorsque nous combinons les deux expressions !

$$\mathrm{Cov}(\bar{\mathbf{s}}_i,\bar{\mathbf{s}}_j) = \hat{\boldsymbol{\Sigma}}_{i,j} + \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\sigma_{ap}^2 = \boldsymbol{\Sigma}_{i,j} - \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{v^T\boldsymbol{\Sigma}v+\sigma_e^2} + \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})_i(\boldsymbol{\Sigma}\bar{\mathbf{v}})_j}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\sigma_{ap}^2$$

Et donc, la matrice de covariance de la nouvelle instance *Distorted BDD* est

$$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + \sigma_{ap}^2\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2} = \boldsymbol{\Sigma} + (\sigma_{ap}^2-\sigma_e^2-\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}})\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}.$$

A présent, calculons $\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\cdot\boldsymbol{\Sigma}\cdot(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp})^T$.

$$\begin{aligned}\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\cdot\boldsymbol{\Sigma}\cdot(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp})^T &= \boldsymbol{\Sigma} - \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}\boldsymbol{\Sigma} - \boldsymbol{\Sigma}(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}})^T + \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}\boldsymbol{\Sigma}(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}})^T\\ &= \boldsymbol{\Sigma} - \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} - \frac{\boldsymbol{\Sigma}(\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T)^T}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}(\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T)^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\\ &= \boldsymbol{\Sigma} - \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} - \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + \frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\\ &= \boldsymbol{\Sigma} - 2\frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}}{\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2} + (\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}})\frac{\boldsymbol{\Sigma}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\boldsymbol{\Sigma}}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\\ &= \boldsymbol{\Sigma} + \left(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}} - 2(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)\right)\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\\ &= \boldsymbol{\Sigma} - \left(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}} + 2\sigma_e^2\right)\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\end{aligned}$$

Et donc,

$$\begin{aligned}\boldsymbol{\Sigma}' &= \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\cdot\boldsymbol{\Sigma}\cdot(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp})^T + \left((\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+2\sigma_e^2) + (\sigma_{ap}^2-\sigma_e^2-\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}})\right)\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}\\ &= \hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp}\cdot\boldsymbol{\Sigma}\cdot(\hat{\mathbf{\Pi}}_{\bar{\mathbf{v}}}^{\perp})^T + (\sigma_{ap}^2+\sigma_e^2)\cdot\frac{(\boldsymbol{\Sigma}\bar{\mathbf{v}})(\boldsymbol{\Sigma}\bar{\mathbf{v}})^T}{(\bar{\mathbf{v}}^T\boldsymbol{\Sigma}\bar{\mathbf{v}}+\sigma_e^2)^2}.\end{aligned}$$

Le théorème est ainsi démontré. □

D Intégration des indices q-modulaires

D.1 Un théorème plus général, la réduction forcée de dimension

Théorème 8 (Réduction forcée de dimension)**.** *Soit une instance Distorted BDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ de rang maximal. Notons d sa dimension. Soit une matrice $\mathbf{\Pi} \in \mathbb{R}^{(d-\Delta)\times d}$ de rang maximal avec $\Delta \geq 0$.*

Soit $\mathbf{W} \in \mathbb{R}^{d\times\Delta}$ *un ensemble de* Δ *vecteurs, primitif vis à vis de* Λ*. Il existe* $\mathbf{\Gamma} \in \mathbb{R}^{d\times(d-\Delta)}$ *tel que*

$$\forall \mathbf{z} \in \Lambda, \exists! \mathbf{y} \in \mathbb{Z}^{\Delta} : \mathbf{z} = \mathbf{\Gamma\Pi z} + \mathbf{W y}$$

si et seulement si $\mathbf{W} \in \operatorname{Ker} \mathbf{\Pi}$.

Prenons donc un $\mathbf{W}$ *tel que* $\mathbf{\Pi W} = \mathbf{0}$.

I. *Un tel* $\mathbf{W}$*, ensemble de vecteurs inclus dans le noyau de* $\mathbf{\Pi}$*, primitif vis à vis de* Λ*, est unique à une matrice unimodulaire près. Donc, en particulier,* $\det(\mathbf{W}^T\mathbf{W})$ *est fixé.*

II. *Il existe* $\mathbf{R} \in \mathbb{R}^{d\times(d-\Delta)}$ *tel que* $\mathbf{\Gamma} = \mathbf{R}(\mathbf{\Pi R})^{-1}$ *et que* $(\mathbf{R}|\mathbf{W})$ *est une base de* Λ.

III. $(\mathbf{\Gamma}|\mathbf{W})$ *est inversible, et*

$$(\mathbf{\Gamma}|\mathbf{W})^{-1} = \begin{pmatrix} \mathbf{\Pi} \\ \mathbf{H}^T \end{pmatrix}$$

où $\mathbf{H} := (\mathbf{I}_d - \mathbf{\Pi}^T\mathbf{\Gamma}^T)\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1} \in \mathbb{R}^{d\times\Delta}$*. Donc, nous avons*

$$\begin{aligned} \mathbf{\Pi\Gamma} &= \mathbf{I}_{d-\Delta}, & \mathbf{\Pi W} &= \mathbf{0}, \\ \mathbf{H}^T\mathbf{\Gamma} &= \mathbf{0}, & \mathbf{H}^T\mathbf{W} &= \mathbf{I}_{\Delta}. \end{aligned}$$

De manière similaire à $\mathbf{W}$*,* $\mathbf{H}$ *est un ensemble de* Δ *vecteurs, primitif vis à vis de* Λ^*.

IV. $\mathbf{\Gamma\Pi}$ *est une projection oblique qui projette les vecteurs sur* $\operatorname{Span}(\mathbf{\Gamma}) = \operatorname{Span}(\mathbf{H})^{\perp}$ *en suivant la direction de* $\operatorname{Span}(\mathbf{W})$.

V. *Le réseau* $\Lambda' := \Pi\Lambda$ *est de rang maximal et vérifie les relations suivantes :*

$$\operatorname{Vol}(\Lambda') = \sqrt{\frac{\det(\mathbf{\Pi\Pi}^T)}{\det(\mathbf{W}^T\mathbf{W})}} \cdot \operatorname{Vol}(\Lambda),$$

$$\Lambda'^* = \mathbf{\Gamma}^T \cdot \Lambda^*.$$

La formulation de la réduction forcée de dimension pour Λ^* *donne*

$$\forall \mathbf{z} \in \Lambda^*, \exists! \mathbf{y} \in \mathbb{Z}^{\Delta} : \mathbf{z} = \mathbf{\Pi}^T\mathbf{\Gamma}^T\mathbf{z} + \mathbf{H y}.$$

Définissons l'instance Distorted BDD $\mathcal{I}' = \mathbf{\Pi} \cdot \mathcal{I}$*. Alors* $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \mathbf{\Sigma}')$ *avec*

$$\begin{aligned} \boldsymbol{\mu}' &\leftarrow \mathbf{\Pi} \cdot \boldsymbol{\mu}, \\ \mathbf{\Sigma}' &\leftarrow \mathbf{\Pi} \cdot \mathbf{\Sigma} \cdot \mathbf{\Pi}^T. \end{aligned}$$

Pour être capables de restaurer le secret $\mathbf{s} \in \Lambda$ *d'une réduction forcée de dimension avec* Π*, nous devons connaître (à partir d'une source extérieure) le* $\mathbf{y_s}$ *dans* $\mathbf{s} = \mathbf{\Gamma\Pi s} + \mathbf{W y_s}$*, i.e.*

$$\mathbf{y_s} = \mathbf{H}^T\mathbf{s} \in \mathbb{Z}^{\Delta}.$$

Si $\Delta = 1$*, l'information dont nous avons besoin peut être reformulée en*

$$\langle \mathbf{s}, \mathbf{H} \rangle \in \mathbb{Z}.$$

Démonstration. Commençons par démontrer l'équivalence du début du théorème, *i.e.* l'équivalence entre ces deux affirmations suivantes :

❁. Il existe $\mathbf{\Gamma} \in \mathbb{R}^{d\times(d-\Delta)}$ tel que $\forall \mathbf{z} \in \Lambda, \exists! \mathbf{y} \in \mathbb{Z}^{\Delta} : \mathbf{z} = \mathbf{\Gamma\Pi z} + \mathbf{W y}$;

❀. $\mathbf{W} \in \operatorname{Ker} \mathbf{\Pi}$.

Procédons par double implication.

[❄ ⇒ ❀] **Supposons qu'il existe un $\mathbf{\Gamma} \in \mathbb{R}^{d\times(d-\Delta)}$ tel que**

$$\forall \mathbf{z} \in \Lambda, \exists \mathbf{y} \in \mathbb{Z}^{\Delta} : \mathbf{z} = \mathbf{\Gamma\Pi z} + \mathbf{W y}.$$

Prenons un $\mathbf{z} \in \Lambda$. Il existe $\mathbf{y}$ tel que $\mathbf{z} = \mathbf{\Gamma\Pi z} + \mathbf{W y}$, donc

$$\mathbf{y} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T(\mathbf{I_d} - \mathbf{\Gamma\Pi})\mathbf{z}.$$

Définissons $\mathbf{H} := (\mathbf{I_d} - \mathbf{\Pi}^T\mathbf{\Gamma}^T)\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1} \in \mathbb{R}^{d\times\Delta}$. Alors, $\mathbf{y} = \mathbf{H}^T\mathbf{z}$ et donc

$$\mathbf{z} = (\mathbf{\Gamma}|\mathbf{W}) \begin{pmatrix} \mathbf{\Pi z} \\ \mathbf{y} \end{pmatrix} = (\mathbf{\Gamma}|\mathbf{W}) \begin{pmatrix} \mathbf{\Pi} \\ \mathbf{H}^T \end{pmatrix} \mathbf{z}.$$

Puisque cette relation est valide pour tout $\mathbf{z}$ de Λ et comme Λ est de rang maximal, nous avons

$$(\mathbf{\Gamma}|\mathbf{W}) \begin{pmatrix} \mathbf{\Pi} \\ \mathbf{H}^T \end{pmatrix} = \mathbf{I_d}.$$

Par conséquent, $(\mathbf{\Gamma}|\mathbf{W})$ est inversible et

$$(\mathbf{\Gamma}|\mathbf{W})^{-1} = \begin{pmatrix} \mathbf{\Pi} \\ \mathbf{H}^T \end{pmatrix}.$$

En particulier, nous obtenons les relations suivantes :

$$\begin{array}{ll} \mathbf{\Pi\Gamma} = \mathbf{I}_{d-\Delta}, & \mathbf{\Pi W} = \mathbf{0}, \\ \mathbf{H}^T\mathbf{\Gamma} = 0, & \mathbf{H}^T\mathbf{W} = \mathbf{I}_{\Delta}. \end{array}$$

Nous avons donc... $\mathbf{\Pi W} = \mathbf{0}$.

[❀ ⇒ ❄] **A présent, supposons que $\mathbf{W} \in \operatorname{Ker}\mathbf{\Pi}$.** Nous pouvons compléter $\mathbf{W}$ en une base $\mathbf{B}$ de Λ.

$$\mathbf{B} = (\mathbf{R}|\mathbf{W})$$

Nous avons $\mathbf{\Pi R} \in \mathbb{R}^{(d-\Delta)\times(d-\Delta)}$. Si $\mathbf{\Pi R}$ n'était pas de rang maximal, il existerait $\mathbf{r} \in \operatorname{Span}(\mathbf{R})$ tel que $\mathbf{\Pi r} = \mathbf{0}$. On aurait par conséquent $\mathbf{W} + \mathbb{R}\mathbf{r} \subset \operatorname{Ker}\mathbf{\Pi}$. Mais $\mathbf{r} \notin \operatorname{Span}(\mathbf{W})$ parce que $(\mathbf{R}|\mathbf{W})$ est une base. Donc,

$$\dim(\mathbf{W} + \mathbb{R}\mathbf{r}) = \dim(\mathbf{W}) + 1 = \Delta + 1.$$

Ce qui est absurde car $\dim \operatorname{Ker}\mathbf{\Pi} = \dim(\mathbb{R}^d) - \operatorname{rank}(\mathbf{\Pi}) = \Delta$, donc $\mathbf{\Pi R}$ est inversible.

Définissons $\mathbf{\Gamma} = \mathbf{R} \cdot (\mathbf{\Pi R})^{-1}$.

Prenons un $\mathbf{z} \in \Lambda$. Il existe $\mathbf{x} \in \mathbb{Z}^d$ tel que $\mathbf{z} = (\mathbf{R}|\mathbf{W})\mathbf{x}$. Notons $\mathbf{y} := (\mathbf{0}|\mathbf{I}_{\Delta})\mathbf{x} \in \mathbb{Z}^{\Delta}$.

$$\begin{aligned} \mathbf{\Gamma\Pi z} + \mathbf{W y} &= \mathbf{R}(\mathbf{\Pi R})^{-1}\mathbf{\Pi}(\mathbf{R}|\mathbf{W})\mathbf{x} + \mathbf{W}(\mathbf{0}|\mathbf{I}_{\Delta})\mathbf{x} \\ &= \mathbf{R}(\mathbf{\Pi R})^{-1}(\mathbf{\Pi R}|\mathbf{0})\mathbf{x} + (\mathbf{0}|\mathbf{W})\mathbf{x} \\ &= \left(\mathbf{R}(\mathbf{\Pi R})^{-1}\mathbf{\Pi R}|\mathbf{0}\right)\mathbf{x} + (\mathbf{0}|\mathbf{W})\mathbf{x} \\ &= (\mathbf{R}|\mathbf{0})\mathbf{x} + (\mathbf{0}|\mathbf{W})\mathbf{x} \\ &= (\mathbf{R}|\mathbf{W})\mathbf{x} = \mathbf{z} \end{aligned}$$

L'unicité de $\mathbf{y}$ provient de l'unicité de la décomposition de $(\mathbf{z} - \mathbf{\Gamma\Pi z})$ dans la base $\mathbf{W}$ du réseau $\mathbf{W}\mathbb{Z}^{\Delta}$.

A présent, démontrons tous les points du théorème, les uns après les autres. **Dans ce qui va suivre, nous prendrons un $\mathbf{W}$ tel que $\mathbf{\Pi W} = \mathbf{0}$.**

I. Soient $\mathbf{W_1}$ et $\mathbf{W_2}$ deux ensembles de vecteurs inclus dans le noyau de $\mathbf{\Pi}$, primitifs vis à vis de Λ. Nous pouvons compléter $\mathbf{W_1}$ et $\mathbf{W_2}$ en deux bases $\mathbf{B_1}$ et $\mathbf{B_2}$ de Λ.

$$\mathbf{B_1} = (\mathbf{R_1}|\mathbf{W_1}) \text{ et } \mathbf{B_2} = (\mathbf{R_2}|\mathbf{W_2})$$

Il existe donc une matrice unimodulaire $\mathbf{U} = \begin{pmatrix} \mathbf{U_1} & \mathbf{U_2} \\ \mathbf{U_3} & \mathbf{U_4} \end{pmatrix}$ telle que $\mathbf{B_1} = \mathbf{B_2 U}$. Appliquons $\mathbf{\Pi}$ à cette égalité, nous obtenons

$$(\mathbf{\Pi R_1}|\mathbf{0}) = (\mathbf{\Pi R_2}|\mathbf{0}) \begin{pmatrix} \mathbf{U_1} & \mathbf{U_2} \\ \mathbf{U_3} & \mathbf{U_4} \end{pmatrix} = (\mathbf{\Pi R_2 U_1}|\mathbf{\Pi R_2 U_2}).$$

Nous savons que $\mathbf{\Pi R_2}$ est inversible, donc $\mathbf{U_2} = \mathbf{0}$. Alors $1 = \det(\mathbf{U}) = \det(\mathbf{U_1})\det(\mathbf{U_4})$. Comme $\mathbf{U_1}$ et $\mathbf{U_4}$ sont des matrices d'entiers, nous en déduisons que $\det(\mathbf{U_4}) = 1$ et donc $\mathbf{U_4}$ est une matrice unimodulaire. Comme $\mathbf{B_1} = \mathbf{B_2 U}$ et $\mathbf{U_2} = \mathbf{0}$, nous avons

$$\mathbf{W_1} = \mathbf{W_2 U_4}$$

donc,

$$\det(\mathbf{W_1}^T\mathbf{W_1}) = \det(\mathbf{U_4}^T\mathbf{W_2}^T\mathbf{W_2 U_4}) = \det(\mathbf{U_4})\det(\mathbf{W_2}^T\mathbf{W_2})\det(\mathbf{U_4}) = \det(\mathbf{W_2}^T\mathbf{W_2}).$$

II. Complétons $\mathbf{W}$ en une base $\mathbf{B} = (\mathbf{R}|\mathbf{W})$ de Λ. Comme $\mathbf{W}\mathbb{Z}^\Delta \subset \Lambda$, nous avons

$$\mathbf{\Gamma\Pi}(\mathbf{R}|\mathbf{W})\mathbb{Z}^d = \mathbf{\Gamma\Pi}\Lambda \subset \Lambda$$

donc, $\mathbf{\Gamma\Pi R}\mathbb{Z}^{d-\Delta}$ est un sous-réseau de Λ. Puisque

$$\mathbf{\Gamma\Pi R}\mathbb{Z}^{d-\Delta} + \mathbf{W}\mathbb{Z}^\Delta = \mathbf{\Gamma\Pi}\Lambda + \mathbf{W}\mathbb{Z}^\Delta = \Lambda,$$

nous déduisons que $(\mathbf{\Gamma\Pi R}|\mathbf{W})$ est une base de Λ. Définissons $\mathbf{R'} := \mathbf{\Gamma\Pi R}$.

$$\begin{aligned} \mathbf{R'}(\mathbf{\Pi R'})^{-1} &= \mathbf{\Gamma\Pi R}(\mathbf{\Pi\Gamma\Pi R})^{-1} \\ &= \mathbf{\Gamma\Pi R}(\mathbf{\Pi R})^{-1} \\ &= \mathbf{\Gamma} \end{aligned}$$

parce que $\mathbf{\Pi\Gamma} = \mathbf{I}_{d-\Delta}$, et nous avons $(\mathbf{R'}|\mathbf{W})$ qui est une base de Λ.

III. Nous avons déjà prouvé que $(\mathbf{\Gamma}|\mathbf{W})$ est inversible et que

$$(\mathbf{\Gamma}|\mathbf{W})^{-1} = \begin{pmatrix} \mathbf{\Pi} \\ \mathbf{H}^T \end{pmatrix}$$

où $\mathbf{H} := (\mathbf{I}_d - \mathbf{\Pi}^T\mathbf{\Gamma}^T)\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1} \in \mathbb{R}^{d\times\Delta}$. Reste à montrer que $\mathbf{H}$ est primitif. Nous pouvons compléter $\mathbf{W}$ en une base $\mathbf{B} = (\mathbf{R}|\mathbf{W})$ de Λ telle que $\mathbf{\Gamma} = \mathbf{R}(\mathbf{\Pi R})^{-1}$.

$$(\mathbf{\Gamma}|\mathbf{W}) = (\mathbf{R}|\mathbf{W}) \begin{pmatrix} (\mathbf{\Pi R})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_\Delta \end{pmatrix}$$

Alors,

$$(\mathbf{\Pi}^T|\mathbf{H}) = (\mathbf{\Gamma}|\mathbf{W})^{-T} = (\mathbf{R}|\mathbf{W})^{-T} \begin{pmatrix} \mathbf{R}^T\mathbf{\Pi}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_\Delta \end{pmatrix}.$$

Comme $(\mathbf{R}|\mathbf{W})$ est une base primale de Λ, alors $(\mathbf{R}|\mathbf{W})^{-T}$ est une base duale de Λ. Donc $\mathbf{H}$ est égal aux Δ dernières colonnes de $(\mathbf{R}|\mathbf{W})^{-T}$, qui sont des vecteurs de Λ^*.

IV. $\mathbf{\Gamma\Pi}$ est une projection. En effet,

$$(\mathbf{\Gamma\Pi})^2 = \mathbf{\Gamma}(\mathbf{\Pi\Gamma})\mathbf{\Pi} = \mathbf{\Gamma\Pi}.$$

Le rang de $\mathbf{\Gamma\Pi}$ est $d - \Delta$, donc le noyau est de dimension Δ. $\mathbf{W}$ est dans le noyau de $\mathbf{\Pi}$, et donc $\operatorname{Ker}\mathbf{\Pi} = \operatorname{Span}(\mathbf{W})$. Quelle est l'image ? Un vecteur projeté par $\mathbf{\Gamma\Pi}$ est une combinaison linéaire des colonnes de $\mathbf{\Gamma}$, donc l'image de la projection est incluse dans $\operatorname{Span}(\mathbf{\Gamma})$. Comme $\operatorname{rank}(\mathbf{\Gamma\Pi}) = d - \Delta$, nous avons $\operatorname{Im}(\mathbf{\Gamma\Pi}) = \operatorname{Span}(\mathbf{\Gamma})$. Et comme que $\mathbf{H}^T\mathbf{\Gamma} = \mathbf{0}$, $\operatorname{Span}(\mathbf{\Gamma}) = \operatorname{Span}(\mathbf{H})^\perp$.

V. Soit une base $\mathbf{B} = (\mathbf{R}|\mathbf{W})$ de Λ telle que $\mathbf{\Gamma} = \mathbf{R}(\mathbf{\Pi R})^{-1}$. Puisque Λ est de rang maximal, $\mathbf{B}$ est également de rang maximal. $\mathbf{B'} = \mathbf{\Pi B}$ est une famille génératrice de Λ'.

$$\operatorname{rank}(\Lambda') = \operatorname{rank}(\mathbf{B'}) = \operatorname{rank}(\mathbf{\Pi B}) = \operatorname{rank}(\mathbf{\Pi}) = d - \Delta$$

$$\dim(\Lambda') = \text{nb\_rows}(\mathbf{B'}) = \text{nb\_rows}(\mathbf{\Pi B}) = \text{nb\_rows}(\mathbf{\Pi}) = d - \Delta$$

Par conséquent, Λ' est de rang maximal.
Trouvons une relation entre les volumes des deux réseaux. $\mathbf{\Pi B} = \mathbf{\Pi}(\mathbf{R}|\mathbf{W}) = (\mathbf{\Pi R}|\mathbf{0})$ est une famille génératrice de $\Lambda' = \mathbf{\Pi}\Lambda$. Et donc, $\mathbf{B'} := \mathbf{\Pi R}$ est une base de Λ'.

$$\mathbf{B} = (\mathbf{R}|\mathbf{W}) = (\mathbf{R}|\mathbf{W}) \begin{pmatrix} (\mathbf{\Pi R})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{\Pi R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = (\mathbf{\Gamma}|\mathbf{W}) \begin{pmatrix} \mathbf{B'} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

donc, $\det(\mathbf{B}) = \det(\mathbf{\Gamma}|\mathbf{W}) \det(\mathbf{B'})$. De plus,

$$\begin{pmatrix} \mathbf{\Pi} \\ \mathbf{W}^T \end{pmatrix} (\mathbf{\Gamma}|\mathbf{W}) = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ * & \mathbf{W}^T\mathbf{W} \end{pmatrix}$$

donc, $\det(\mathbf{\Pi}^T|\mathbf{W}) \cdot \det(\mathbf{\Gamma}|\mathbf{W}) = \det(\mathbf{W}^T\mathbf{W})$. Et alors,

$$\begin{pmatrix} \mathbf{\Pi} \\ \mathbf{W}^T \end{pmatrix} (\mathbf{\Pi}^T|\mathbf{W}) = \begin{pmatrix} \mathbf{\Pi\Pi}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^T\mathbf{W} \end{pmatrix}$$

donc, $\det(\mathbf{\Pi}^T|\mathbf{W})^2 = \det(\mathbf{\Pi\Pi}^T) \det(\mathbf{W}^T\mathbf{W})$. Si nous réunissons ces relations, nous obtenons

$$\begin{aligned}
\operatorname{Vol}(\Lambda') &= |\det(\mathbf{B'})| \\
&= \left|\frac{1}{\det(\mathbf{\Gamma}|\mathbf{W})}\right| \cdot |\det(\mathbf{B})| \\
&= \left|\frac{\det(\mathbf{\Pi}^T|\mathbf{W})}{\det(\mathbf{W}^T\mathbf{W})}\right| \cdot |\det(\mathbf{B})| \\
&= \left|\frac{\sqrt{\det(\mathbf{\Pi\Pi}^T) \cdot \det(\mathbf{W}^T\mathbf{W})}}{\det(\mathbf{W}^T\mathbf{W})}\right| \cdot |\det(\mathbf{B})| \\
&= \left|\sqrt{\frac{\det(\mathbf{\Pi\Pi}^T)}{\det(\mathbf{W}^T\mathbf{W})}}\right| \cdot \operatorname{Vol}(\Lambda).
\end{aligned}$$

A présent, prouvons la relation $\Lambda'^* = \mathbf{\Gamma}^T \cdot \Lambda^*$,

$$\begin{aligned}
\Lambda'^* &= \{\mathbf{v} \in \operatorname{Span}(\Lambda') : \forall \mathbf{z} \in \Lambda', \langle \mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}\} \\
&= \{\mathbf{v} \in \operatorname{Span}(\Lambda') : \forall \mathbf{z} \in \Lambda, \langle \mathbf{v}, \mathbf{\Pi z} \rangle \in \mathbb{Z}\} \\
&= \{\mathbf{v} \in \operatorname{Span}(\Lambda') : \forall \mathbf{z} \in \Lambda, \langle \mathbf{\Pi}^T\mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}\} \\
&= \mathbf{\Gamma}^T \cdot \{\mathbf{v} \in \mathbf{\Pi}^T \operatorname{Span}(\Lambda') : \forall \mathbf{z} \in \Lambda, \langle \mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}\} \\
&= \mathbf{\Gamma}^T \cdot \{\mathbf{v} \in \operatorname{Span}(\Lambda) : \forall \mathbf{z} \in \Lambda, \langle \mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}\}.
\end{aligned}$$

Le sens direct de la dernière égalité est trivial, parce que $\mathrm{Span}(\Lambda) = \mathbb{R}^d$ (Λ est de rang maximal). Donc, prouvons le sens indirect. Prenons $(\mathbf{\Pi}^T|\mathbf{H})$ comme base de $\mathbb{R}^d$.
Prenons un élément $\mathbf{x}$ de $\mathbf{\Gamma}^T \cdot \{\mathbf{v} \in \mathrm{Span}(\Lambda) : \forall \mathbf{z} \in \Lambda, \langle \mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}\}$. Il existe un $\mathbf{v} \in \mathrm{Span}(\Lambda)$ tel que $\mathbf{x} = \mathbf{\Gamma}^T\mathbf{v}$ et $\forall \mathbf{z} \in \Lambda, \langle \mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}$. De plus, il existe $\mathbf{y} \in \mathbb{R}^d$ tel que $\mathbf{v} = (\mathbf{\Pi}^T|\mathbf{H})\mathbf{y}$.

$$\mathbf{x} = \mathbf{\Gamma}^T\mathbf{v} = \mathbf{\Gamma}^T(\mathbf{\Pi}^T|\mathbf{H})\mathbf{y} = (\mathbf{I}_{d-\Delta}|\mathbf{0})\mathbf{y}$$

Définissons $\mathbf{v}' := \mathbf{v} - (\mathbf{0}|\mathbf{H})\mathbf{y} = (\mathbf{\Pi}^T|\mathbf{0})\mathbf{y} = \mathbf{\Pi}^T(\mathbf{I}_{d-\Delta}|\mathbf{0})\mathbf{y} \in \mathbf{\Pi}^T \cdot \mathbb{R}^{d-\Delta} = \mathbf{\Pi}^T \cdot \mathrm{Span}(\Lambda')$.

$$(\mathbf{0}|\mathbf{H})\mathbf{y} = (\mathbf{0}|\mathbf{H})(\mathbf{\Pi}^T|\mathbf{H})^{-1}\mathbf{v} = (\mathbf{0}|\mathbf{H}) \begin{pmatrix} \mathbf{\Gamma}^T \\ \mathbf{W}^T \end{pmatrix} \mathbf{v} = \mathbf{H}\mathbf{W}^T\mathbf{v}$$

Choisissons arbitrairement un $\mathbf{z} \in \Lambda$. Il existe $\mathbf{y}' \in \mathbb{Z}^\Delta$ tel que $\mathbf{z} = \mathbf{\Gamma}\mathbf{\Pi}\mathbf{z} + \mathbf{W}\mathbf{y}'$.

$$\begin{aligned}
\langle \mathbf{v}', \mathbf{z} \rangle &= \langle \mathbf{v}, \mathbf{z} \rangle - \langle (\mathbf{0}|\mathbf{H})\mathbf{y}, \mathbf{z} \rangle \\
&= \langle \mathbf{v}, \mathbf{z} \rangle - \langle \mathbf{H}\mathbf{W}^T\mathbf{v}, \mathbf{z} \rangle \\
&= \langle \mathbf{v}, \mathbf{z} \rangle - \langle \mathbf{v}, \mathbf{W}\mathbf{H}^T\mathbf{z} \rangle \\
&= \langle \mathbf{v}, \mathbf{z} \rangle - \langle \mathbf{v}, \mathbf{W}\mathbf{H}^T(\mathbf{\Gamma}\mathbf{\Pi}\mathbf{z} + \mathbf{W}\mathbf{y}') \rangle \\
&= \langle \mathbf{v}, \mathbf{z} \rangle - \langle \mathbf{v}, \mathbf{W}(\mathbf{H}^T\mathbf{\Gamma})\mathbf{\Pi}\mathbf{z} + \mathbf{W}(\mathbf{H}^T\mathbf{W})\mathbf{y}' \rangle \\
&= \langle \mathbf{v}, \mathbf{z} \rangle - \langle \mathbf{v}, \mathbf{W}\mathbf{y}' \rangle \in \mathbb{Z}
\end{aligned}$$

Comme $\mathbf{z}$ et $\mathbf{W}\mathbf{y}'$ sont dans le réseau Λ, $\langle \mathbf{v}, \mathbf{z} \rangle$ et $\langle \mathbf{v}, \mathbf{W}\mathbf{y}' \rangle$ sont des entiers. Donc,

$$\mathbf{v}' \in \{\mathbf{v} \in \mathbf{\Pi}^T \mathrm{Span}(\Lambda') : \forall \mathbf{z} \in \Lambda, \langle \mathbf{v}, \mathbf{z} \rangle \in \mathbb{Z}\}$$

et

$$\mathbf{\Gamma}\mathbf{v}' = \mathbf{x} - \mathbf{\Gamma}^T(\mathbf{0}|\mathbf{H})\mathbf{y} = \mathbf{x} - (\mathbf{0}|\mathbf{\Gamma}^T\mathbf{H})\mathbf{y} = \mathbf{x}.$$

Cela conclut le sens indirect de la dernière égalité, et donc

$$\Lambda'^* = \mathbf{\Gamma}^T \cdot \Lambda^*.$$

La formulation d'une réduction forcée de dimension pour Λ^* donne

$$\forall \mathbf{z} \in \Lambda^*, \exists! \mathbf{y} \in \mathbb{Z}^\Delta : \mathbf{z} = \hat{\mathbf{\Gamma}}\mathbf{\Gamma}^T\mathbf{z} + \mathbf{H}\mathbf{y}$$

parce que $\mathbf{H} \in \mathrm{Ker}\, \mathbf{\Gamma}^T$. Prenons une base $(\mathbf{R}|\mathbf{W})$ de Λ telle que $\mathbf{\Gamma} = \mathbf{R}(\mathbf{\Pi}\mathbf{R})^{-1}$. Alors

$$(\mathbf{R}|\mathbf{W})^{-T} = (\mathbf{\Pi}^T(\mathbf{R}^T\mathbf{\Pi}^T)^{-1}|\mathbf{H})$$

est une base de Λ^*. Donc, si nous considérons le $\hat{\mathbf{\Gamma}}$ associé à $\mathbf{\Pi}^T(\mathbf{R}^T\mathbf{\Pi}^T)^{-1}$,

$$\begin{aligned}
\hat{\mathbf{\Gamma}} &= (\mathbf{\Pi}^T(\mathbf{R}^T\mathbf{\Pi}^T)^{-1})(\mathbf{\Gamma}^T\mathbf{\Pi}^T(\mathbf{R}^T\mathbf{\Pi}^T)^{-1})^{-1} \\
&= (\mathbf{\Pi}^T(\mathbf{R}^T\mathbf{\Pi}^T)^{-1})((\mathbf{R}^T\mathbf{\Pi}^T)^{-1}\mathbf{R}^T\mathbf{\Pi}^T(\mathbf{R}^T\mathbf{\Pi}^T)^{-1})^{-1} \\
&= \mathbf{\Pi}^T \cdot (\mathbf{R}^T\mathbf{\Pi}^T)^{-1} \cdot (\mathbf{R}^T\mathbf{\Pi}^T) \cdot (\mathbf{R}^T\mathbf{\Pi}^T)^{-1} \cdot (\mathbf{R}^T\mathbf{\Pi}^T) \\
&= \mathbf{\Pi}^T.
\end{aligned}$$

Définissons l'instance *Distorted BDD* $\mathcal{I}' = \mathbf{\Pi} \cdot \mathcal{I}$. Alors $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \mathbf{\Sigma}')$ avec

$$\begin{aligned}
\boldsymbol{\mu}' &\leftarrow \mathbf{\Pi} \cdot \boldsymbol{\mu}, \\
\mathbf{\Sigma}' &\leftarrow \mathbf{\Pi} \cdot \mathbf{\Sigma} \cdot \mathbf{\Pi}^T.
\end{aligned}$$

En effet,

$$\begin{aligned}
\boldsymbol{\mu}' = \mathbb{E}[\mathbf{s}'] = \mathbb{E}[\boldsymbol{\Pi}\mathbf{s}] = \boldsymbol{\Pi}, \mathbb{E}[\mathbf{s}] = \boldsymbol{\Pi}\boldsymbol{\mu} \\
\boldsymbol{\Sigma}'_{i,j} = \mathrm{Cov}(\mathbf{s}'_i, \mathbf{s}'_j) = \mathrm{Cov}((\boldsymbol{\Pi}\mathbf{s})_i, (\boldsymbol{\Pi}\mathbf{s})_j) \\
= \mathrm{Cov}(\sum_k \boldsymbol{\Pi}_{i,k}\mathbf{s}_k, \sum_l \boldsymbol{\Pi}_{j,l}\mathbf{s}_l) \\
= \sum_k \boldsymbol{\Pi}_{i,k}(\sum_l \mathrm{Cov}(\mathbf{s}_k, \mathbf{s}_l)\boldsymbol{\Pi}_{j,l}) \\
= \sum_k \boldsymbol{\Pi}_{i,k}(\sum_l \boldsymbol{\Sigma}_{k,l}(\boldsymbol{\Pi}^T)_{l,j}) \\
= \sum_k \boldsymbol{\Pi}_{i,k}(\boldsymbol{\Sigma}\boldsymbol{\Pi}^T)_{k,j} \\
= (\boldsymbol{\Pi}\boldsymbol{\Sigma}\boldsymbol{\Pi}^T)_{i,j}.
\end{aligned}$$

Comme le petit secret $\mathbf{s}$ appartient à Λ, il existe $\mathbf{y} \in \mathbb{Z}^\Delta$ tel que

$$\mathbf{s} = \boldsymbol{\Gamma}\mathbf{s}' + \mathbf{W}\mathbf{y}_{\bar{\mathbf{s}}} \text{ avec } \mathbf{s}' := \boldsymbol{\Pi}\mathbf{s}.$$

Pour restaurer $\mathbf{s}$ à partir de $\mathbf{s}'$, nous devons connaître la valeur de $\mathbf{y}_\mathbf{s}$. Comme $\mathbf{s} = (\boldsymbol{\Gamma}|\mathbf{W}) \begin{pmatrix} \mathbf{s}' \\ \mathbf{y}_{\bar{\mathbf{s}}} \end{pmatrix}$, nous avons

$$\mathbf{y}_{\bar{\mathbf{s}}} = (\mathbf{0}|\mathbf{I}_\Delta)(\boldsymbol{\Gamma}|\mathbf{W})^{-1}\mathbf{s} = \mathbf{H}^T\mathbf{s}.$$

Voici ainsi démontrés tous les éléments du théorème. □

D.2 Un autre point de vue sur la réduction, la réduction assistée de dimension

Théorème 9 (Réduction assistée de dimension)**.** *Soit une instance Distorted BDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ de rang maximal. Notons d sa dimension. Soit un ensemble $\mathbf{H} \in \mathbb{R}^{d\times\Delta}$ de vecteurs, primitif vis à vis de Λ^*. Si nous connaissons un ensemble $\mathbf{W} \in \mathbb{R}^{d\times\Delta}$ de vecteurs, primitif vis à vis de Λ, et une fonction $f_{\mathbf{H},\mathbf{W}}$ tels que*

$$\mathbf{H}^T\mathbf{W} = \mathbf{I}_\Delta \qquad \textit{et} \qquad \mathbf{H}^T\mathbf{s} = f_{\mathbf{H},\mathbf{W}}(\boldsymbol{\Pi}^\perp_\mathbf{W}\mathbf{s})$$

avec $\mathbf{s}$ le petit secret de $\mathcal{I}$, alors il est possible de procéder à une réduction forcée de dimension sans perte d'information sur $\mathbf{s}$.

Démonstration. Il existe une matrice $\boldsymbol{\Gamma} \in \mathbb{R}^{d\times(d-\Delta)}$ de rang maximal telle que $\mathbf{H}^T\boldsymbol{\Gamma} = \mathbf{0}$ (nous pouvons par exemple la construire avec le procédé de Gram-Schmidt). Donc, par unicité de l'inverse, il existe $\boldsymbol{\Pi}$ telle que

$$(\boldsymbol{\Gamma}|\mathbf{W})^{-1} = \begin{pmatrix} \boldsymbol{\Pi} \\ \mathbf{H}^T \end{pmatrix}.$$

Par conséquent, nous sommes dans les conditions requises pour appliquer une réduction forcée de dimension avec $\boldsymbol{\Pi}$, $\boldsymbol{\Gamma}$, $\mathbf{W}$ et $\mathbf{H}$. Pour appliquer la réduction, nous avons juste besoin de $\boldsymbol{\Pi}$. La nouvelle instance *Distorted BDD* est $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\begin{aligned}
\Lambda' &\leftarrow \boldsymbol{\Pi} \cdot \Lambda, \\
\boldsymbol{\mu}' &\leftarrow \boldsymbol{\Pi} \cdot \boldsymbol{\mu}, \\
\boldsymbol{\Sigma}' &\leftarrow \boldsymbol{\Pi} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Pi}^T.
\end{aligned}$$

Pour restaurer le petit secret $\mathbf{s} \in \Lambda$, la réduction de réseaux (par BKZ) nous donnera la valeur de $\mathbf{s}' := \mathbf{\Pi s}$. Calculons alors la quantité $\boldsymbol{\gamma}$ suivante (définie uniquement avec des variables connues) :

$$\boldsymbol{\gamma} := \mathbf{\Gamma s}' + \mathbf{W} \cdot f_{\mathbf{H},\mathbf{W}} \left(\mathbf{\Pi}^{\perp}_{\mathbf{W}} \mathbf{\Gamma s}' \right).$$

Puisque $\mathrm{Span}(\mathbf{W})^{\perp}$ est égal à $\mathrm{Span}(\mathbf{\Pi}^T)$ et $\mathbf{\Pi}^T$ est de rang maximal, nous avons $\mathbf{\Pi}^{\perp}_{\mathbf{W}} = \mathbf{\Pi}^T(\mathbf{\Pi\Pi}^T)^{-1}\mathbf{\Pi}$.

$$\begin{aligned}
\boldsymbol{\gamma} &= \mathbf{\Gamma s}' + \mathbf{W} \cdot f_{\mathbf{H},\mathbf{W}} \left(\mathbf{\Pi}^{\perp}_{\mathbf{W}} \mathbf{\Gamma s}' \right) \\
&= \mathbf{\Gamma\Pi} s + \mathbf{W} \cdot f_{\mathbf{H},\mathbf{W}} \left(\mathbf{\Pi}^T(\mathbf{\Pi\Pi}^T)^{-1}\mathbf{\Pi} \cdot \mathbf{\Gamma\Pi s} \right) \\
&= \mathbf{\Gamma\Pi s} + \mathbf{W} \cdot f_{\mathbf{H},\mathbf{W}} \left(\mathbf{\Pi}^T(\mathbf{\Pi\Pi}^T)^{-1}(\mathbf{\Pi\Gamma})\mathbf{\Pi s} \right) \\
&= \mathbf{\Gamma\Pi s} + \mathbf{W} \cdot f_{\mathbf{H},\mathbf{W}} \left(\mathbf{\Pi}^T(\mathbf{\Pi\Pi}^T)^{-1}\mathbf{\Pi} s \right) \\
&= \mathbf{\Gamma\Pi s} + \mathbf{W} \cdot f_{\mathbf{H},\mathbf{W}} \left(\mathbf{\Pi}^{\perp}_{\mathbf{W}} \mathbf{s} \right) \\
&= \mathbf{\Gamma\Pi s} + \mathbf{W} \cdot \mathbf{H}^T \mathbf{s} \\
&= \mathbf{s}
\end{aligned}$$

En calculant $\boldsymbol{\gamma}$, nous restaurons donc bien le secret $\mathbf{s}$. □

D.3 Application pour les indices q-modulaires

Supposons que nous connaissons

$$\langle \mathbf{s}, \mathbf{v} \rangle \equiv 0 \pmod{q}.$$

Supposons qu'il existe un index i tel que $\mathbf{v}_i \neq 0$. Comme nous sommes dans $\mathbb{F}_q$, nous pouvons définir

$$\mathbf{v}' \leftarrow (\mathbf{v}_i)^{-1}\mathbf{v}$$

et donc, nous avons

$$\langle \mathbf{s}, \mathbf{v}' \rangle \equiv 0 \pmod{q} \text{ avec } \mathbf{v}'_i = 1.$$

Nous pouvons donc élaguer le réseau Λ en utilisant la formule de [11].

$$\Lambda \leftarrow \Lambda \cap \{\mathbf{x} \in \mathbb{Z}^d : \langle \mathbf{x}, \mathbf{v}' \rangle \equiv 0 \pmod{q}\}$$

En pratique, pour obtenir ce réseau élagué, nous injectons juste le vecteur $\mathbf{v}'/q$ dans une base duale. Donc, $\mathbf{h} = \mathbf{v}'/q$ est un vecteur dual du nouveau réseau, et il a de fortes chances d'être primitif vis à vis du réseau dual. De plus, $\mathbf{w} = q \cdot \mathbf{e}_i$ est un vecteur primitif du réseau primal et

$$\langle \mathbf{h}, \mathbf{w} \rangle = \langle \mathbf{v}', \mathbf{e}_i \rangle = \mathbf{v}'_i = 1.$$

Notons $\mathbf{y_s} := \langle \mathbf{s}, \frac{\mathbf{v}'}{q} \rangle \in \mathbb{Z}$. Pour appliquer le théorème de la réduction assistée de dimension, nous avons besoin d'une fonction $f_{\mathbf{h},\mathbf{w}}$ telle que

$$\mathbf{y_s} = f_{\mathbf{h},\mathbf{w}}(\mathbf{\Pi}^{\perp}_{q \cdot \mathbf{e_i}} \mathbf{s}).$$

Comment trouver une telle fonction ? Nous savons que $\langle \mathbf{s}, \mathbf{v}' \rangle = q \cdot \mathbf{y_s}$. Donc, nous avons

$$\begin{aligned}
q \cdot \mathbf{y_s} - \langle \mathbf{\Pi}^{\perp}_{q \cdot \mathbf{e}_i} \mathbf{s}, \mathbf{v}' \rangle &= q \cdot \mathbf{y_s} - \langle \mathbf{s}, \mathbf{\Pi}^{\perp}_{q \cdot \mathbf{e}_i} \mathbf{v}' \rangle \\
&= q \cdot \mathbf{y_s} - \langle \mathbf{s}, \mathbf{v}' - \mathbf{e}_i \rangle \\
&= q \cdot \mathbf{y_s} - \langle \mathbf{s}, \mathbf{v}' \rangle + \langle \mathbf{s}, \mathbf{e}_i \rangle \\
&= \langle \mathbf{s}, \mathbf{e}_i \rangle = \mathbf{s}_i \in \{-\alpha, ..., \alpha\}.
\end{aligned}$$

Par conséquent, $\mathbf{y_s}$ est l'unique entier appartenant à l'intervalle

$$\left[\frac{-\alpha + \langle \mathbf{\Pi}^{\perp}_{q\cdot\mathbf{e}_i}\mathbf{s}, \mathbf{v}'\rangle}{q}, \frac{\alpha + \langle \mathbf{\Pi}^{\perp}_{q\cdot\mathbf{e}_i}\mathbf{s}, \mathbf{v}'\rangle}{q}\right]$$

avec α la valeur maximale qu'un coefficient du secret peut avoir (dans Kyber, $\alpha = 2$). Donc, construisons la fonction $f_{\mathbf{h},\mathbf{w}}$ comme

$$f_{\mathbf{h},\mathbf{w}} : \mathbf{x} \mapsto \left\lfloor \frac{\langle \mathbf{x}, \mathbf{v}'\rangle}{q} \right\rceil.$$

Il est donc possible de procéder à une réduction assistée de dimension : il est possible de réduire le réseau et de restaurer le secret. Mais qu'en est-il de l'adaptation des indices suivants dans le réseau réduit ? Supposons que nous avons un indice quelconque

$$\langle \mathbf{s}, \hat{\mathbf{v}}\rangle = \mathcal{D}$$

et que nous avons appliqué la transformation précédente. Avec les notations précédentes, nous avons

$$\begin{aligned}\langle \mathbf{s}, \hat{\mathbf{v}}\rangle &= \langle \mathbf{\Gamma}\mathbf{s}' + \mathbf{y_s}\cdot\mathbf{w}, \hat{\mathbf{v}}\rangle \\ &= \langle \mathbf{\Gamma}\mathbf{s}', \hat{\mathbf{v}}\rangle + \langle \mathbf{y_s}\cdot\mathbf{w}, \hat{\mathbf{v}}\rangle \\ &= \langle \mathbf{s}', \mathbf{\Gamma}^T\hat{\mathbf{v}}\rangle + \langle \langle \mathbf{s}, \mathbf{h}\rangle \cdot (q\cdot\mathbf{e}_i), \hat{\mathbf{v}}\rangle \\ &= \langle \mathbf{s}', \mathbf{\Gamma}^T\hat{\mathbf{v}}\rangle + q\cdot\langle \mathbf{s}, \mathbf{h}\rangle\cdot\hat{\mathbf{v}}_i\end{aligned}$$

et donc, l'indice pour le secret réduit est...

$$\langle \mathbf{s}', \mathbf{\Gamma}^T\hat{\mathbf{v}}\rangle = \mathcal{D} - q\langle \mathbf{s}, \mathbf{h}\rangle\cdot\hat{\mathbf{v}}_i.$$

A ce stade (durant l'intégration des indices), nous ne connaissons pas $\langle \mathbf{s}, \mathbf{h}\rangle$. Donc, si $\hat{\mathbf{v}}_i$ est nulle, nous avons $\langle \mathbf{s}', \mathbf{\Gamma}^T\hat{\mathbf{v}}\rangle = \mathcal{D}$ et nous pouvons intégrer l'indice sans problème. Mais si $\hat{\mathbf{v}}_i \neq 0$, nous pouvons juste avoir l'indice

$$\langle \mathbf{s}, \mathbf{\Gamma}^T\hat{\mathbf{v}}\rangle \equiv \mathcal{D} \pmod q.$$

Donc, les types d'indices que nous pouvons intégrer après cette réduction assistée de dimension sont

- les indices avec $\hat{\mathbf{v}}_i$ égal à 0 ;
- les indices qui ne sont pas impactés par le modulo q.

Pour résumer tout ce qui vient d'être dit, nous obtenons le théorème suivant :

Théorème 10. *Soit l'indice suivant :*

$$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}}\rangle \equiv 0 \pmod q.$$

Pour prendre cet indice en compte dans une instance Distorted DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \mathbf{\Sigma})$ de rang maximal, nous pouvons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \mathbf{\Sigma}')$ avec

$$\begin{aligned}\Lambda' &= \mathbf{\Pi}\cdot(\Lambda \cap \{\mathbf{x} \in \mathbb{Z}^d : \langle \mathbf{x}, \bar{\mathbf{v}}\rangle \equiv 0 \pmod q\}), \\ \boldsymbol{\mu}' &= \mathbf{\Pi}\cdot\boldsymbol{\mu}, \\ \mathbf{\Sigma}' &= \mathbf{\Pi}\cdot\mathbf{\Sigma}\cdot\mathbf{\Pi}^T,\end{aligned}$$

où

$$\mathbf{\Pi} = \begin{pmatrix} \mathbf{I}_{p\times p} & \mathbf{0}_{p\times 1} & \mathbf{0}_{p\times(d-p-1)} \\ \mathbf{0}_{(d-p-1)\times p} & \mathbf{0}_{(d-p-1)\times 1} & \mathbf{I}_{(d-p-1)\times(d-p-1)} \end{pmatrix}$$

avec p un pivot possible de $\bar{\mathbf{v}}$, i.e. $\bar{\mathbf{v}}_p \neq 0$.

Nous avons alors

$$\begin{aligned}\mathrm{Vol}(\Lambda') &= \sqrt{\frac{\det(\mathbf{\Pi}\mathbf{\Pi}^T)}{\det(\mathbf{w}^T\mathbf{w})}} \cdot \mathrm{Vol}(\Lambda \cap \{\mathbf{x} \in \mathbb{Z}^d : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle = 0 \pmod q\}) \\ &= \sqrt{\frac{1}{q^2}} \cdot q \cdot \mathrm{Vol}(\Lambda) \\ &= \mathrm{Vol}(\Lambda).\end{aligned}$$

De plus,

$$(\mathbf{\Pi}^T|\mathbf{h})^{-1} = (\mathbf{\Gamma}_{\bar{\mathbf{v}}}|\mathbf{w})^T$$

avec

$$\mathbf{\Gamma}_{\bar{\mathbf{v}}} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{v_1}^T & -\mathbf{v_2}^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

où $\mathbf{v_1}$ et $\mathbf{v_2}$ sont définis par $\bar{\mathbf{v}}^T = \bar{\mathbf{v}}_p \cdot (\mathbf{v_1}^T, 1, \mathbf{v_2}^T)$.

En pratique, pour calculer une base de Λ', nous devrons injecter $\bar{\mathbf{v}}/q$ dans une base duale et appliquer dessus $\mathbf{\Gamma}_{\bar{\mathbf{v}}}^T$, *i.e.*

$$\mathbf{D}' \leftarrow \mathbf{\Gamma}_v^T \cdot \left(\mathbf{D} \cup \left\{\frac{\bar{\mathbf{v}}}{q}\right\}\right).$$