Chiffrement avancé à partir du problème *Learning With Errors*

Version modifiée d'un chapitre de l'ouvrage "Informatique Mathématique, une photographie en 2014", publié aux Presses Universitaires de Perpignan

19 Mars 2014

Fabien Laguillaumie[1,3], Adeline Langlois[2,3], and Damien Stehlé[2,3]

[1] Université Claude Bernard Lyon 1
[2] École Normale Supérieure de Lyon
[3] LIP (U. Lyon, CNRS, ENS Lyon, INRIA, UCBL),
46 Allée d'Italie, 69364 Lyon Cedex 07, France.
{fabien.laguillaumie,adeline.langlois,damien.stehle}@ens-lyon.fr

Résumé Le problème *Learning With Errors* (LWE) est algorithmiquement difficile pour des instances aléatoires. Il a été introduit par Oded Regev en 2005 et, depuis lors, il s'est avéré très utile pour construire des primitives cryptographiques, pour assurer la confidentialité de l'information. Dans ce chapitre, nous présenterons le problème LWE et illustrerons sa richesse, en décrivant des schémas de chiffrement avancés pouvant être prouvés au moins aussi sûrs que LWE est difficile. Nous rappellerons le concept fondamental de chiffrement, puis nous nous focaliserons sur les notions de chiffrement fondé sur l'identité et de chiffrement par attributs.

1 Introduction

La cryptographie, dont on attend en général qu'elle assure la confidentialité et l'authenticité des communications, subit actuellement une mutation significative, qui se traduit par des changements dans les paradigmes mêmes qui la fondent. Jusqu'à présent, la construction des primitives cryptographiques était pensée *de point à point*, entre deux entités (Alice *et* Bob), qui souhaitent échanger des données de façon sécurisée.

Cette cryptographie classique est très mal adaptée aux nouveaux usages, liés en particulier au développement du *cloud*. Le *cloud* permet de stocker des quantités de données extrêmement importantes, dans des fermes de serveurs sur lesquelles les usagers n'ont en réalité aucune garantie de sécurité. Parmi les fonctionnalités cryptographiques qui seraient nécessaires dans un environnement de ce type, on peut citer l'accès anonyme aux données, les calculs sécurisés sur celles-ci, et le contrôle de leur accès. Ce dernier point est crucial pour maîtriser la diffusion d'informations personnelles. Les politiques d'accès des données personnelles stockées sur le *cloud* peuvent être à la fois très sensibles (notamment pour les applications médicales), évoluées et dynamiques (comme par exemple sur les réseaux sociaux). Actuellement, les services de gestion de données de type *cloud* ne mettent en œuvre aucun protocole cryptographique adapté.

Pour concevoir des systèmes cryptographiques avancés permettant d'obtenir ces fonctionnalités avancées, un objet mathématique semble particulièrement prometteur : il s'agit des réseaux euclidiens. Ce sont des grilles de points régulièrement espacés dans l'espace euclidien $\mathbb{R}^n$. Ils étaient déjà implicitement utilisés dans les cryptosystèmes de type sac-à-dos [57], ainsi qu'en cryptanalyse [55]. Mais leur utilisation cryptographique a connu un renouveau récent, grâce aux travaux d'Ajtai [6], puis de Regev [63,64]. En plein essor, ils sont aujourd'hui considérés comme l'alternative la plus crédible à la cryptographie traditionnelle, reposant sur des variantes du problème de la factorisation de grands entiers et du problème du logarithme discret dans des groupes algébriques. Les réseaux euclidiens offrent une grande flexibilité pour la conception de cryptosystèmes : ils sont au cœur du premier protocole de chiffrement complètement homomorphe [29], qui permet d'effectuer n'importe quel calcul (efficace) sur des messages, en ne manipulant que leurs chiffrés. D'autres propriétés les rendent particulièrement attractifs : la sécurité des systèmes fondés sur les réseaux peut souvent être réduite à la difficulté des instances *pires cas* de problèmes portant sur les réseaux euclidiens ; leur efficacité est parfois asymptotiquement quasi-optimale (par exemple, chiffrer k bits de données en temps quasi-linéaire en k) ; et ils semblent résister aux attaques utilisant d'hypothétiques ordinateurs quantiques, à l'opposé des cryptosystèmes qui reposent sur les problèmes de la factorisation d'entiers (comme le célèbre RSA) et du logarithme discret.

Les primitives de la cryptographie fondée sur les réseaux reposent, pour la plupart, sur deux problèmes intermédiaires. Les instances aléatoires de ceux-ci sont au moins aussi difficiles à résoudre que les instances les plus difficiles de problèmes portant sur les réseaux (on parle de réductions pires cas moyens cas), et ils sont plus faciles d'utilisation lorsqu'il s'agit de construire des primitives cryptographiques. Le problème *Short Integer Solution* (SIS), introduit par Ajtai [6], permet, entre autres, de construire des fonctions de hachage résistantes aux collisions et des signatures numériques [30,43]. Le problème *Learning With Errors* (LWE), introduit par Regev [63,64], a été utilisé pour élaborer des primitives cryptographiques, en particulier autour du chiffrement.

Dans ce chapitre, nous nous focaliserons sur LWE. Informellement, LWE consiste à résoudre un système linéaire *surdéterminé mais bruité*, modulo un entier q. Nous le décrirons en détails à la Section 3. Nous illustrerons la flexibilité de la cryptographie fondée sur les réseaux en décrivant des schémas de chiffrement dont la sécurité peut être prouvée, sous l'hypothèse que LWE est difficile à résoudre. Nous commencerons par des schémas de chiffrement asymétriques élémentaires (Section 4). Ensuite, dans la Section 5, nous décrirons ce qu'est un schéma de chiffrement fondé sur l'identité (IBE, pour *identity-based encryption*) et montrerons comment en construire un à partir de LWE. Enfin, dans la Section 6, nous introduirons une primitive plus avancée encore, le chiffrements par attributs avec attributs publics [17], que l'on appellera plus simplement chiffrement par attributs. Ce dernier est une sous-classe d'une nouvelle primitive, appelée *chiffrement fonctionnel*, qui marque un changement de paradigme fondamental en cryptographie à clé publique. En effet, il permet de décider d'une politique de déchiffrement, s'adressant non plus à un destinataire unique,

mais à tout un ensemble. Parmi celui-ci, les utilisateurs satisfaisant cette politique auront l'accès aux données en clair. La définition de cette primitive n'est pas triviale, elle l'a été de manières légèrement différentes dans [42,9,58], avant d'être récemment formalisée par [17]. Nous décrirons donc un schéma tiré de la sous-classe des chiffrements par attributs, dont la sécurité repose sur la difficulté de LWE.

Les schémas de chiffrement fondé sur l'identité permettent de chiffrer pour une identité plutôt que pour une clé publique : étant donnée une identité, une autorité peut générer une clé de déchiffrement associée. Les premiers schémas de chiffrement fondé sur l'identité datent d'une dizaine d'années [24,15]. Ils utilisent des résidus quadratiques et des couplages sur des courbes elliptiques, respectivement. La première construction à partir d'objets mathématiques différents date de 2008 : Gentry, Peikert et Vaikuntanathan [30] en élaborent un en utilisant le problème LWE. Cette construction a été une étape majeure pour montrer que les réseaux permettent d'obtenir des primitives avancées. Nous décrirons ici l'IBE de [22], dont la sécurité est plus simple à mettre en évidence.

Les schémas de chiffrement par attributs permettent de mettre en place des politiques de contrôle d'accès sur des données, y compris dans le *cloud*, avec une sécurité cryptographique : un attaquant ne pourra rien faire des données même s'il pénètre dans le système sur lequel elles sont stockées. Le chiffrement fondé sur l'identité est un cas particulier de chiffrement par attributs, pour une politique d'accès très simple (le récepteur peut déchiffrer un message chiffré pour son identité). Les premiers protocoles permettant d'implémenter des politiques d'accès un peu plus élaborées sont récents : on peut citer par exemple l'article [66], qui a formalisé la notion de chiffrement par attributs et proposé un chiffrement fondé sur des identités bruitées (le récepteur peut déchiffrer un message chiffré pour une identité qui lui est proche), et [33], qui permet des politiques d'accès qui peuvent être codées à partir de formules booléennes. Nous décrirons ici le récent protocole de Gorbunov, Vaikuntanathan et Wee [32], qui permet, à l'aide de LWE, d'implémenter des politiques d'accès pouvant être décrites à l'aide de n'importe quel circuit de taille polynomiale. À ce jour, seuls les réseaux euclidiens permettent de construire des chiffrements par attributs aussi puissants.

Ce chapitre est une double introduction à des primitives de chiffrement avancées, et au problème LWE. Il ne s'agit pas d'un article de survol : les primitives que nous avons choisies l'ont été pour leur valeur pédagogique, et de nombreuses variantes et améliorations ont été mises de côté. Aussi, nous ne donnons que les idées des preuves, afin de ne pas noyer les concepts sous la technique. Les preuves complètes peuvent être trouvées dans les références que nous donnons.

Notations. Pour un entier $q \geq 2$, on définit $\mathbb{Z}_q$ comme l'ensemble des entiers compris entre 0 et $q-1$, muni de l'addition et de la multiplication modulo q, l'équipant ainsi d'une structure d'anneau. Si q est premier, ce qui sera notre cas tout au long de ce chapitre, alors $\mathbb{Z}_q$ est un corps. Les vecteurs seront toujours en gras, et les matrices en lettres capitales

et en gras. Nos vecteurs sont des vecteurs colonnes. La notation $\{0,1\}^*$ désigne l'ensemble des chaînes finies de bits.

Si $f : \mathbb{R}^n \mapsto [0, +\infty[$ et $A \subseteq \mathbb{R}^n$ est dénombrable, on définit $f(A) = \sum_{x \in A} f(x) \in [0, +\infty]$.

Nous utiliserons les notations de Landau classiques, ainsi que la notation $\widetilde{O}(\cdot)$, définie de la façon suivante : on écrira $f(n) = \widetilde{O}(g(n))$ s'il existe une constante $c > 0$ telle que $f(n) = O(g(n) \cdot \log^c g(n))$. Une fonction $f(n)$ est dite négligeable si $f(n) = O(1/n^c)$ pour toute constante $c > 0$. La notation $poly(n)$ désigne une fonction arbitraire $f(n)$ bornée supérieurement par un polynôme en n.

Si D est une loi de probabilité, la notation $x \hookleftarrow D$ signifiera que l'on échantillonne x selon D. Pour X un ensemble fini, la loi uniforme sur X sera notée $U(X)$.

Nous dirons qu'un problème algorithmique est difficile quand aucun algorithme probabiliste de complexité polynomiale (en la taille de l'instance) ne peut le résoudre avec probabilité non-négligeable (en la taille de l'instance). La probabilité porte sur l'aléa interne de l'algorithme et sur celui de l'instance si celle-ci est choisie aléatoirement.

2 Préliminaires

Dans cette section, nous décrivons succintement quelques objets mathématiques et résultats portant sur ceux-ci, que nous utiliserons par la suite.

Indistinguabilité entre distributions. Soient D_0 et D_1 deux distributions sur un même support dénombrable X. La *distance statistique* entre D_0 et D_1 est définie par $\Delta(D_0, D_1) = \sum_{x \in X} |D_0(x) - D_1(x)|/2$. Il s'agit bien d'une distance. De plus, les trois propriétés suivantes sont satisfaites :

- si D_0 et D_1 ont un support dénombrable commun et D'_0 et D'_1 également, alors $\Delta((D_0, D'_0), (D_1, D'_1)) \leq \Delta(D_0, D_1) + \Delta(D'_0, D'_1)$;
- si f est une fonction éventuellement randomisée, alors $\Delta(f(D_0), f(D_1)) \leq \Delta(D_0, D_1)$;
- pour tout A sous-ensemble du support, on a $|D_0(A) - D_1(A)| \leq \Delta(D_0, D_1)$.

Les preuves de ces propriétés sont relativement élémentaires, et peuvent être trouvées, par exemple, dans [51, Chap. 8].

Soient deux suites de distributions $D_0^{(n)}$ et $D_1^{(n)}$ telles que pour tout n, les distributions $D_0^{(n)}$ et $D_1^{(n)}$ aient un support dénombrable commun. Un *distingueur* entre $D_0^{(n)}$ et $D_1^{(n)}$ est un algorithme $\mathcal{A}$ probabiliste polynomial en n qui a accès à un oracle $\mathcal{O}_{D_b^{(n)}}$ produisant des échantillons indépendants distribués selon $D_b^{(n)}$ pour un $b \in \{0,1\}$ fixé (noté alors $\mathcal{A}^{\mathcal{O}_{D_b^{(\cdot)}}}$) ; le distingueur est tel que la quantité

$$|\Pr[\mathcal{A}^{\mathcal{O}_{D_0^{(n)}}}(x) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{D_1^{(n)}}}(x) = 1]|,$$

appelée *avantage* de $\mathcal{A}$, est une fonction non-négligeable de n. Les probabilités sont prises vis-à-vis de l'aléa de l'oracle $\mathcal{O}_{D_b^{(n)}}$ et de l'aléa interne de $\mathcal{A}$. Si aucun tel algorithme $\mathcal{A}$

n'existe, on dit que les suites de distributions D_0 et D_1 sont *calculatoirement indistinguables*.

Si la distance statistique $\Delta(D_0^{(n)}, D_1^{(n)})$ est une fonction négligeable de n, on dit que ces suites de distributions sont *statistiquement indistinguables*. Dans ce cas, on peut utiliser les propriétés listées ci-dessus pour montrer qu'il n'existe pas de distingueur entre ces suites de distributions : l'indistinguabilité statistique est une propriété plus forte que l'indistinguabilité calculatoire. Tant que l'on se restreint à des algorithmes probabilistes polynomiaux, on peut sans heurt raisonner comme si deux distributions statistiquement indistinguables étaient identiques.

Réseaux euclidiens. Un réseau euclidien est un ensemble de la forme $L = \sum \mathbb{Z}\mathbf{b}_i \subseteq \mathbb{R}^n$ avec des vecteurs $\mathbf{b}_i$ linéairement indépendants. Ces derniers forment une base du réseau. Celui-ci en possède une infinité (quand la dimension est supérieure à 2), qui sont liées entre elles par des transformations unimodulaires (c'est-à-dire linéaires, inversibles et à coefficients entiers). La Figure 1 donne un exemple de réseau de dimension 2, avec deux de ses bases. Le problème algorithmique central portant sur les réseaux, le problème du vecteur le plus court, SVP (pour *Shortest Vector Problem*), consiste à trouver un vecteur de longueur $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \{\mathbf{0}\})$, à partir d'une base arbitraire. Nous définissons ci-dessous sa version décisionnelle.

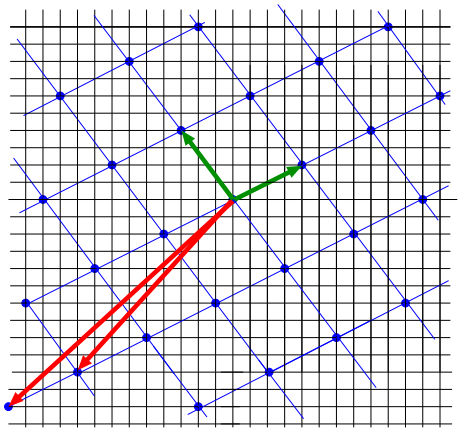

Figure 1. Un réseau de dimension 2 et deux de ses bases.

Définition 1 *Soient $n \geq 2$ un entier, et $\gamma \geq 1$ un réel. La version décisionnelle relâchée du problème du vecteur le plus court $GapSVP_\gamma$ est la suivante : étant donnée une base d'un réseau L, déterminer si $\lambda(L) \leq 1$ ou si $\lambda(L) \geq \gamma$.*

Un algorithme résolvant GapSVP_γ n'a pas à se prononcer, ou peut répondre n'importe quoi, lorsque $\lambda(L) \in]1, \gamma[$. Ainsi, la difficulté algorithmique de GapSVP_γ décroît avec γ. Ce problème est NP-difficile sous des réductions randomisées pour de petites valeurs de γ (voir [6,49,34]), et le plus petit γ pour lequel on sait le résoudre en temps polynomial est $\gamma = 2^{\Theta(n\frac{\log\log n}{\log n})}$ (cet algorithme s'obtient en combinant [67] et [54]). Ce problème est conjecturé exponentiellement difficile à résoudre vis-à-vis de la dimension n, pour tout γ polynomial en n. C'est sur cette conjecture que reposent la sécurité de la plupart des primitives cryptographiques reposant sur les réseaux euclidiens.

Le lecteur désireux d'approfondir le sujet pourra trouver une introduction détaillée aux réseaux euclidiens et à leurs aspects calculatoires dans les notes de cours de Regev [62].

Gaussiennes discrètes. Nous rappelons que la loi normale, ou gaussienne, de paramètre s et de centre $c \in \mathbb{R}$ est la loi de densité

$$D_{s,c}(x) = \frac{1}{s}\exp(-\pi(x-c)^2/s^2)$$

(pour $x \in \mathbb{R}$). La loi gaussienne de dimension n, de paramètre s et de centre $\mathbf{c} \in \mathbb{R}^n$ est définie par

$$D_{s,\mathbf{c}}(\mathbf{x}) = \prod_i D_{s,c_i}(x_i) = \frac{1}{s^n}\exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/s^2).$$

Dans ce chapitre, nous utiliserons plusieurs fois des variantes discrètes de la loi gaussienne, dont le support est un réseau euclidien. Pour un réseau support Λ, un paramètre $s > 0$ et un centre $\mathbf{c} \in \mathbb{R}^n$, nous définissons la loi de probabilité suivante :

$$\forall \mathbf{x} \in \Lambda : D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = D_{s,\mathbf{c}}(\mathbf{x})/D_{s,\mathbf{c}}(\Lambda).$$

La distribution est la même que celle d'une gaussienne continue mais tous les points échantillonnés appartiennent au support Λ. La Figure 2 fournit un exemple pour $\Lambda = \mathbb{Z}$. Quand le centre est $\mathbf{0}$, nous omettrons l'indice $\mathbf{c}$.

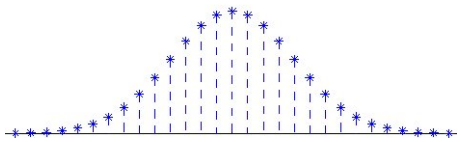

Figure 2. Gaussienne discrète sur $\mathbb{Z}$.

Les propriétés satisfaites par les gaussiennes continues le sont très souvent aussi par leurs variantes discrètes. En particulier, la queue de la distribution a un poids très faible.

Lemme 1 ([10, Le. 1.5]) *Pour tout réseau Λ et paramètre $s > 0$, on a* $\Pr_{\mathbf{b} \hookleftarrow D_{\Lambda,s}}[\|\mathbf{b}\| \leq \sqrt{n}s] \geq 1 - 2^{-n}$.

D'autres propriétés des gaussiennes continues s'adaptent aux gaussiennes discrètes quand le paramètre s est suffisamment grand. Ceci est quantifié à l'aide du *paramètre de lissage* du réseau Λ, qui correspond à la plus petite quantité de bruit gaussien à déposer sur chaque point de Λ pour le faire « disparaître ». Nous ne détaillerons pas plus ce sujet ici, car nous n'en avons pas besoin par la suite. Une définition précise est donnée dans [53], ainsi qu'un certain nombre de propriétés élémentaires.

Par ailleurs, quand le paramètre s est suffisamment grand, on peut échantillonner selon $D_{\Lambda,s,\mathbf{c}}$ de manière efficace. À chaque fois que nous échantillonnerons selon des lois gaussiennes discrètes, implicitement, nous utiliserons l'algorithme mentionné dans le résultat suivant. Cet algorithme avait auparavant été introduit par Klein [40], dans un autre contexte, avant que Gentry *et al.* [30] l'utilisent pour échantillonner selon des lois gaussiennes discrètes.

Lemme 2 ([30, Th. 4.1] et [19, Le. 2.3]) *Il existe un algorithme probabiliste polynomial qui, étant donnés $(\mathbf{b}_i)_{i \leq n}$ une base d'un réseau Λ, $s \geq \max_i \|\mathbf{b}_i\| \cdot \Omega(\sqrt{\log n})$ et $\mathbf{c} \in \mathbb{R}^n$, renvoie un échantillon de $D_{\Lambda,s,\mathbf{c}}$.*

Les bases courtes d'un réseau sont plus difficiles à obtenir : elles apportent notamment des indications pour résoudre le problème GapSVP. Comme l'indiquent les Lemmes 2 et 1 précédents, elles permettent, entre autres, d'échantillonner de vecteurs de petites normes dans le réseau.

3 Le problème *Learning With Errors* (LWE)

Le problème LWE fait intervenir trois paramètres : la dimension n, le module q, et le facteur d'erreur α. Les paramètres q et α sont souvent choisis comme des fonctions de n, qui croît vers $+\infty$ dans les résultats asymptotiques. La version calculatoire de LWE consiste à retrouver un vecteur $\mathbf{s} \in \mathbb{Z}_q^n$ à partir d'un nombre arbitraire de produits scalaires bruités entre $\mathbf{s}$ et des vecteurs connus $\mathbf{a}_i$ choisis uniformément dans $\mathbb{Z}_q^n$. Sa version décisionnelle consiste à distinguer ces produits scalaires bruités de la distribution uniforme.

Définition 2 *Soient $n \geq 1$, $q \geq 2$ deux entiers, et $\alpha \in]0,1[$. La distribution $D_{n,q,\alpha}^{LWE}(\mathbf{s})$, pour $\mathbf{s} \in \mathbb{Z}_q^n$, est la distribution sur $\mathbb{Z}_q^{n+1}$ obtenue par l'expérience suivante :*

$$\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n);\ e \hookleftarrow D_{\mathbb{Z},\alpha q};\ \text{renvoyer}\ \ (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^{n+1}.$$

La version calculatoire de LWE, de paramètres n, q et α, est comme suit : soit $\mathbf{s} \in \mathbb{Z}_q^n$; étant donné un nombre arbitraire d'échantillons de $D_{n,q,\alpha}^{LWE}(\mathbf{s})$, trouver $\mathbf{s}$.
La version décisionnelle de LWE, de paramètres n, q et α, est comme suit : soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$; distinguer entre les distributions $D_{n,q,\alpha}^{LWE}(\mathbf{s})$ et $U(\mathbb{Z}_q^{n+1})$.

Il est aussi possible d'écrire ces deux problèmes en version matricielle, lorsqu'on connaît le nombre d'échantillons m. Comme le montre la Figure 3, les m échantillons de la forme $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$ sont représentés par une matrice $\mathbf{A}$ dont les lignes sont les $\mathbf{a}_i$ pour $1 \leq i \leq m$ et par un produit entre $\mathbf{A}$ et $\mathbf{s}$ auquel on ajoute un vecteur $\mathbf{e}$ de taille m dont les coordonnées sont les e_i. Ce vecteur $\mathbf{e}$ peut être directement échantillonné selon $D_{\mathbb{Z}^m, \alpha q}$. La version calculatoire de LWE est alors définie comme suit : Soit $\mathbf{s} \in \mathbb{Z}_q^n$; étant donné $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ où $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ et $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$, trouver $\mathbf{s}$. La version décisionnelle peut s'écrire de manière équivalente.

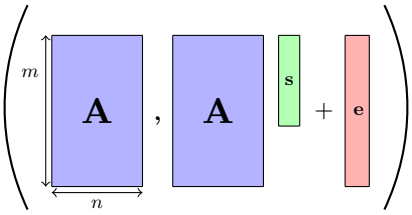

Figure 3. Représentation matricielle de m échantillons de $D_{n,q,\alpha}^{\mathrm{LWE}}(\mathbf{s})$.

Dans la plupart des cas, on choisit q polynomial en n (c'est-à-dire de taille binaire $O(\log n)$) et α entre $1/q$ et 1. Si α est extrêmement petit, alors, avec probabilité très proche de 1, l'erreur e de chaque échantillon demandé est nulle. Dans ce cas, retrouver $\mathbf{s}$ est facile : à partir d'un peu plus de n échantillons, on obtient un système linéaire de rang plein d'inconnue $\mathbf{s}$, que l'on peut résoudre en temps polynomial. Si, au contraire, le facteur d'erreur α est grand, alors e est très proche d'être uniformément distribué modulo q, la distribution $D_{n,q,\alpha}^{\mathrm{LWE}}(\mathbf{s})$ est essentiellement la distribution uniforme, indépendamment de $\mathbf{s}$, et, ainsi, retrouver $\mathbf{s}$ est impossible. Un exemple d'instance est donné à la Figure 4). La difficulté de LWE semble donc croître avec α.

Équivalence des deux variantes de LWE. Les deux variantes de LWE, décisionnelle et calculatoire, sont en fait calculatoirement équivalentes. Comme nous allons le voir, il s'avère que la seconde est beaucoup plus facilement exploitable pour élaborer des primitives cryptographiques.

Une des deux réductions est plus simple que l'autre : résoudre la variante calculatoire permet de résoudre la variante décisionnelle. Soient $D_0 = D_{n,q,\alpha}^{\mathrm{LWE}}(\mathbf{s})$, pour un $\mathbf{s}$ choisi uniformément, et $D_1 = U(\mathbb{Z}_q^{n+1})$ [4] ; nous avons accès à un oracle fournissant des échantillons selon D_b et nous devons retrouver b. Nous commençons par calculer un candidat pour $\mathbf{s}$ en envoyant des échantillons à un oracle qui résout la variante calculatoire. Ensuite, pour de nouveaux échantillons de D_b, nous utilisons les membres gauches $\mathbf{a}_i$ et soustrayons les $\langle \mathbf{a}_i, \mathbf{s} \rangle$

4. Il s'agit de deux suites de distributions, dépendant du paramètre n.

Trouver $(s_1, s_2, s_3, s_4, s_5)$ tel que :

$$\begin{aligned}
s_1 + 22s_2 + 17s_3 + 2s_4 + s_5 &\approx 16 \mod 23\\
3s_1 + 2s_2 + 11s_3 + 7s_4 + 8s_5 &\approx 17 \mod 23\\
15s_1 + 13s_2 + 10s_3 + s_4 + 22s_5 &\approx 3 \mod 23\\
17s_1 + 11s_2 + s_3 + 10s_4 + 3s_5 &\approx 8 \mod 23\\
2s_1 + s_2 + 13s_3 + 6s_4 + 2s_5 &\approx 9 \mod 23\\
4s_1 + 4s_2 + s_3 + 5s_4 + s_5 &\approx 18 \mod 23\\
11s_1 + 12s_2 + 5s_3 + s_4 + 9s_5 &\approx 7 \mod 23
\end{aligned}$$

Figure 4. Une instance de la version calculatoire du problème LWE, pour $q = 23$, $n = 5$ et $m = 7$.

aux membres droits. Nous renvoyons 0 si les résultats obtenus sont tous petits par rapport à q (la distribution de l'oracle est D_0). Sinon, nous renvoyons 1 (la distribution de l'oracle est D_1).

Une première illustration de la flexibilité de LWE est l'existence d'une réduction dans le sens inverse, plus surprenante. Nous nous restreignons ici au cas où le module q est premier et polynomial en n (ces restrictions peuvent être supprimées en travaillant plus, *cf.* [19]). Supposons qu'un algorithme $\mathcal{A}$ résolve efficacement la version décisionnelle de LWE, c'est-à-dire distingue $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$ et $U(\mathbb{Z}_q^{n+1})$ avec avantage non-négligeable, et probabilité non-négligeable, quand les probabilités sont prises par rapport au choix de $\mathbf{s}$ et à l'aléa interne éventuel de $\mathcal{A}$: nous allons utiliser cet algorithme pour résoudre la version calculatoire.

Commençons dans un premier temps par résoudre la version calculatoire de LWE, avec probabilité non-négligeable vis-à-vis de $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. Nous avons accès à un oracle échantillonnant selon $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$, ainsi qu'à $\mathcal{A}$, et souhaitons trouver $\mathbf{s}$. Il suffit de voir comment retrouver la première coordonnée s_1 de $\mathbf{s}$, les autres coordonnées pouvant être traitées de façon identique. Comme q est petit, on peut tester toutes les valeurs possibles $s_1^\star$ de s_1, et se servir de $\mathcal{A}$ pour confirmer ou infirmer cette valeur. Pour cela, on prend un échantillon $(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$, et on donne à $\mathcal{A}$ l'entrée

$$(\mathbf{a}, b) + (u, 0, \ldots, 0, us_1^\star) = (\mathbf{a}', \langle \mathbf{a}', \mathbf{s}\rangle + e + u(s_1^\star - s_1)), \text{ avec } u \hookleftarrow U(\mathbb{Z}_q).$$

De deux choses l'une : si $s_1^\star = s_1$, alors l'échantillon donné en entrée à $\mathcal{A}$ est distribué selon $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$; si au contraire $s_1^\star \neq s_1$, alors la dernière coordonnée est uniformément distribuée indépendamment des n premières, du fait de l'uniformité de u et de l'invertibilité de $s_1^\star - s_1$ modulo q. La réponse de $\mathcal{A}$ nous dit donc si on a bien deviné s_1. Pour augmenter la certitude de ne pas commettre d'erreur, on peut réessayer plusieurs fois.

Dans un deuxième temps, s'il est possible de retrouver $\mathbf{s}$ avec probabilité non-négligeable, quand $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, alors on peut le faire pour tout $\mathbf{s}$. En effet, pour un $\mathbf{s}$ fixé, on peut tirer un $\mathbf{t} \hookleftarrow U(\mathbb{Z}_q^n)$, et transformer un échantillon $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$ de $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$ en un échantillon

$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}+\mathbf{t}\rangle + e)$ de $D^{\mathrm{LWE}}_{n,q,\alpha}(\mathbf{s}+\mathbf{t})$, an ajoutant $\langle \mathbf{a}, \mathbf{t}\rangle$ au membre droit de l'échantillon. Maintenant, puisque $\mathbf{s}+\mathbf{t}$ est uniforme, on peut retrouver $\mathbf{s}+\mathbf{t}$ avec probabilité non-négligeable sur le choix de $\mathbf{t}$, et donc retrouver $\mathbf{s}$ avec probabilité non-négligeable (la probabilité portant sur l'aléa interne du nouvel algorithme).

Difficulté de LWE. Le problème $\mathrm{LWE}_{n,q,\alpha}$ semble extrêmement difficile à résoudre. Le meilleur algorithme connu pour des paramètres (n, q, α) généraux repose sur la réduction des réseaux euclidiens. En particulier sur l'algorithme BKZ de Schnorr [67], améliorant le célèbre algorithme LLL [41]. Le lecteur intéressé pourra trouver une introduction à ces algorithmes dans l'ouvrage [56]. Cette attaque, décrite dans [53], a une complexité $\exp(\widetilde{O}(n \log q / \log^2 \alpha))$.

Ce lien avec les réseaux euclidiens est très profond : les algorithmes sur les réseaux offrent les meilleures attaques contre LWE, mais, par ailleurs, on peut montrer que s'il existait un algorithme efficace résolvant LWE, alors il existerait des algorithmes efficaces contre des problèmes réputés difficiles portant sur les réseaux. En particulier, l'existence d'un algorithme pour LWE impliquerait l'existence d'un algorithme efficace pour résoudre GapSVP_γ. Le théorème suivant regroupe des résultats de [64] et [19].

Théorème 1 *Soient $q \geq 2$ et $\alpha \in (0,1)$ des fonctions de n telles que $\alpha q \geq 2\sqrt{n}$. Si q est premier et polynomial en n, alors il existe une réduction polynomiale quantique de $GapSVP_\gamma$ en dimension n à $LWE_{n,q,\alpha}$, avec $\gamma = \widetilde{O}(n/\alpha)$. Pour tout q, il existe une réduction polynomiale classique de $GapSVP_\gamma$ en dimension $\Theta(\sqrt{n})$ à $LWE_{n,q,\alpha}$, avec $\gamma = \widetilde{O}(n^2/\alpha)$.*

Nous pouvons à présent introduire les concepts utiles de cryptographie et démontrer la puissance du problème LWE pour concevoir des protocoles cryptographiques.

4 Chiffrement à clé publique à partir de LWE

Avant d'aborder les primitives avancées de chiffrement, nous allons définir la primitive élémentaire de chiffrement à clé publique, et décrire deux solutions reposant sur la difficulté du problème LWE. Pour compléter cet introduction au chiffrement, le lecteur pourra par exemple consulter [31,61,38].

4.1 Définitions du chiffrement et de sa sécurité

Le rôle du chiffrement est d'assurer la *confidentialité* des données. Nous nous intéressons dans ce cours à la cryptographie *à clé publique*, dite également cryptographie asymétrique, par opposition à la cryptographie à clé secrète, symétrique. Le concept de cryptographie à clé publique a été introduite par Diffie et Hellman, au milieu de années 1970 [26]. Dans ce contexte, chaque utilisateur possède un couple de clés, l'une étant rendue publique et

disponible (notée pk), l'autre étant gardée secrète par l'utilisateur (elle est notée sk). Ces deux clés sont liées par une relation forte (souvent une relation algébrique) qui est telle que retrouver la clé secrète à partir de la clé publique est difficile. Si Alice souhaite envoyer un message à Bob de façon confidentielle, celle-ci va transformer son message *clair* en un message *chiffré* qui sera illisible (il aura l'apparence d'une donnée « aléatoire ») par tout destinataire illégitime. La clé secrète de Bob permettra à lui et lui seul d'extraire de ce chiffré le message clair initial.

Il faut noter que le chiffrement à clé publique est souvent utilisé pour distribuer des clés symétriques pour ensuite mettre en œuvre un chiffrement à clé secrète, plus rapide, pour chiffrer des communications. On parle alors de chiffrement *hybride* [25].

La définition formelle d'un protocole de chiffrement est la suivante :

Définition 3 (Chiffrement) *Soit λ un entier, appelé paramètre de sécurité. Un* protocole de chiffrement *Π est la donnée de trois algorithmes probabilistes polynomiaux :*

$\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$**.** *L'algorithme de génération de clés prend en entrée λ et renvoie une paire de clés (pk, sk)* [5] *.La première est la clé publique et la seconde est la clé secrète associée.*

$\mathsf{Enc}(1^\lambda, pk, M) \to C$**.** *L'algorithme de chiffrement prend en entrée le paramètre de sécurité λ, une clé publique pk et un message $M \in \{0,1\}^*$. Il renvoie un message chiffré C.*

$\mathsf{Dec}(1^\lambda, sk, C) \to \{M, \perp\}$**.** *L'algorithme de déchiffrement prend en entrée le paramètre de sécurité λ, une clé secrète sk et un chiffré C. Il renvoie soit un message M, soit le symbole $\perp$ qui indique que le chiffré n'est pas valide.*

Le protocole de chiffrement doit être correct*, ce qui signifie que pour tout λ suffisamment grand, et pour tout message $M \in \{0,1\}^*$, si $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ alors, avec probabilité (prise sur tous les aléas internes des algorithmes) négligemment proche de* 1*, on a $\mathsf{Dec}(1^\lambda, sk, \mathsf{Enc}(1^\lambda, pk, M)) = M$.*

Diffie et Hellman n'ont pas proposé de protocole de chiffrement dans leur article fondateur, il a fallu attendre un an pour voir apparaître les premières réalisations : RSA [65] dont la sécurité repose sur la difficulté d'un problème lié à la factorisation des entiers, Merkle-Hellman [48] dont la sécurité repose sur la difficulté d'instances particulières du problème du sac-à-dos et McEliece [47], dont la sécurité repose sur la difficulté d'instances particulières du décodage de codes correcteurs d'erreurs.

Sécurité du chiffrement. Nous définissons maintenant la *sécurité* d'un protocole de chiffrement. Pour se faire, et c'est le cas pour toute primitive cryptographique, il faut précisément évaluer les attaques contre lesquelles on souhaite se prémunir. Que signifie donc « assurer la confidentialité d'une communication » ? Définir un *modèle de sécurité* d'une primitive

5. Le paramètre λ est donné en écriture unaire à l'algorithme : celui-ci est de complexité polynomiale en λ.

cryptographique consiste à définir les objectifs d'un attaquant et les informations dont il dispose pour mener à bien son attaque. Dans le cas du chiffrement, l'objectif d'un attaquant pourrait être, par exemple, de pouvoir déchiffrer tous les messages à destination de Bob qu'il intercepte, ayant comme information à sa disposition la seule clé publique de celui-ci. Ce n'est pas le seul scénario possible.

Un protocole de chiffrement est *totalement cassé* si un attaquant peut retrouver la clé secrète à partir (au moins) de la clé publique. Cela signifie que tout algorithme probabiliste polynomial (dit adversaire ou attaquant) $\mathcal{B}$ aura un succès pour retrouver la clé secrète négligeable, lorsque le *succès* est défini, pour un entier λ comme

$$\Pr\Big[(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda) : \mathcal{B}(pk) = sk\Big].$$

Un protocole cryptographique qui se casse totalement est évidemment complètement inutile. Cependant, la littérature comporte un certain nombre de protocoles qui ont été considérés comme sûrs pendant des années, avant d'être totalement cassés (voir par exemple [23] qui décrit un chiffrement dont la sécurité était censée reposer sur une problème de type sac-à-dos, et sa cryptanalyse [70]).

La notion de sécurité peut-être la plus intuitive pour un protocole de chiffrement est associée à la notion de *sens-unique* : il doit être difficile de retrouver un message clair étant donné son chiffré. Un protocole de chiffrement est dit *à sens-unique* si tout adversaire probabiliste et polynomial $\mathcal{A}$ a un succès pour inverser la fonction de chiffrement négligeable, lorsque son succès est défini, pour un entier λ, comme

$$\Pr\Big[(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda) : \mathcal{A}(pk, \Pi.\mathsf{Enc}(1^\lambda, pk, M)) = M\Big].$$

Il est naturellement plus difficile pour un attaquant de casser totalement un protocole de chiffrement, que de casser son sens unique (c'est-à-dire d'inverser la fonction de chiffrement sans la clé secrète).

Dans les définitions précédentes, l'attaquant n'a pas d'autres informations que la clé publique de l'utilisateur qu'il attaque : il est engagé dans une *attaque à clairs choisis* (CPA pour *chosen plaintext attack*), puisqu'il a la possibilité de chiffrer les messages de son choix grâce à sa connaissance de la clé publique. Néanmoins, on peut renforcer ce modèle en lui donnant accès à un *oracle de déchiffrement*. Dans ce cas, l'attaquant soumet à l'oracle de déchiffrement un chiffré c, qui lui répond par le résultat du déchiffrement $\mathsf{Dec}(1^\lambda, sk, C)$. Cet attaquant a donc des moyens beaucoup plus importants. L'attaque est dite *à chiffrés choisis* (CCA pour *chosen ciphertext attack*).

Un schéma de chiffrement doit en réalité atteindre un niveau de sécurité plus élevé. La meilleure sécurité possible d'un système cryptographique est celle pour laquelle un attaquant ayant l'objectif le plus faible, avec le plus de moyens à sa disposition, n'existe pas. Quel peut être l'objectif le plus faible d'un attaquant ? Obtenir la totalité du contenu d'un message à partir du chiffré peut être inutile pour un attaquant : il peut souhaiter connaître simplement une *partie* du message. La notion qui formalise cette intuition est

l'indistinguabilité des chiffrés (IND) (ou sécurité sémantique). Un protocole de chiffrement assure une indistinguabilité des chiffrés, si un attaquant ne peut pas gagner le jeu suivant : il choisit deux messages M_0 et M_1, les envoie à un challenger qui en chiffre un des deux (tiré uniformément) ; ce dernier envoie le chiffré à l'attaquant qui doit décider (avec une meilleure stratégie que le choix au hasard) lequel des deux messages lui correspond.

Plus formellement, on définit cette notion de sécurité par le jeu suivant :

Expérience $\mathbf{Exp}_{\Pi}^{\mathsf{ind\text{-}atk}}(\mathcal{A})$

$(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda)$
$(M_0, M_1, st) \leftarrow \mathcal{A}_1^{\mathcal{O}}(pk)$
$b^\star \hookleftarrow U(\{0,1\})$
$C^\star \leftarrow \Pi.\mathsf{Enc}(1^\lambda, pk, M_{b^\star})$
$b \leftarrow \mathcal{A}_2^{\mathcal{O}}(st, C^\star)$
Renvoie 1 si $b = b^\star$ et 0 sinon

avec

- Si $\mathsf{atk} = \mathsf{cpa}$, alors $\mathcal{O} = \emptyset$
- Si $\mathsf{atk} = \mathsf{cca}$, alors $\mathcal{O} = \Pi.\mathsf{Dec}(1^\lambda, sk, \cdot)$

dans lequel l'adversaire $\mathcal{A}$ est modélisé par un algorithme fonctionnant en deux étapes $(\mathcal{A}_1, \mathcal{A}_2)$ (l'algorithme $\mathcal{A}_1$ pouvant communiquer des informations à l'algorithme $\mathcal{A}_2$ dans l'état st). L'*avantage* de l'attaquant est alors défini par

$$\mathsf{Adv}_{\Pi}^{\mathsf{ind-atk}}(\mathcal{A}) = \left|\Pr\left(\mathbf{Exp}_{\Pi}^{\mathsf{ind\text{-}atk}}(\mathcal{A}) = 1\right) - 1/2\right|.$$

Le schéma sera considéré sûr du point de vue de l'indistinguabilité des chiffrés si tout attaquant probabiliste polynomial de ce type a un avantage négligeable.

Dans l'expérience aléatoire décrite précédemment, l'indistinguabilité est l'objectif de l'attaquant. Pour y parvenir, il peut avoir divers moyens. Le scénario le plus important est donc celui dans lequel l'attaquant a accès à un oracle de déchiffrement, avec la restriction naturelle qu'il ne doit pas interroger l'oracle sur le chiffré challenge : on parle d'*indistinguabilité dans une attaque à chiffrés choisis adaptative* (IND-CCA). Lorsque l'attaquant n'a pas accès à ces oracles, on parle d'*indistinguabilité dans une attaque à clairs choisis* (IND-CPA). Cette dernière notion, bien que plus faible que la précédente, est importante, car elle est plus simple à obtenir, et il existe des méthodes génériques pour passer à la notion supérieure.

La Figure 5 récapitule les types d'attaques élémentaires qui concernent le chiffrement.

Il est important de noter que pour atteindre l'indistinguabilité des chiffrés, il est nécessaire que le processus de chiffrement soit probabiliste (en particulier, le chiffrement RSA classique n'est pas indistinguable).

Remarque 1 *Dans de nombreuses applications (comme le vote électronique), l'algorithme de chiffrement doit être* homomorphe. *Typiquement, l'espace des messages clairs est muni d'une loi de groupe* $\times$. *Le chiffrement est homomorphe pour cette loi si, à partir de deux chiffrés* C_0 *et* C_1 *de deux messages clairs* M_0 *et* M_1, *on peut obtenir le chiffré du produit des*

Objectif de l'attaquant (difficulté croissante)	Type d'attaque (moyens croissants)
Casse totale Sens unique Indistinguabilité	Clairs choisis Chiffrés choisis

Figure 5. Objectifs et moyens d'un attaquant.

deux messages $M_0 \times M_1$. Un chiffrement qui vérifie cette propriété ne peut pas atteindre *l'indistinguabilité des chiffrés dans une attaque à chiffrés choisis adaptative. Néanmoins, ce type de chiffrement reste fondamental pour concevoir des systèmes cryptographiques complexes. Récemment, Gentry a proposé un chiffrement* complètement homomorphe *[29], permettant d'appliquer n'importe quel circuit à des clairs en ne manipulant que leurs chiffrés. Il permet, entre autres, de sécuriser l'externalisation des calculs.*

Constructions génériques. D'un point de vue théorique, on peut s'interroger sur l'existence de techniques génériques pour obtenir un schéma de chiffrement sûr du point de vue de l'indistinguabilité dans une attaque à clairs choisis, à partir d'un objet élémentaire. Yao a montré dans [72] comment construire des protocoles de chiffrement à clé publique simplement à partir de familles de permutations à trappe. Une permutation à trappe est une permutation qui s'évalue efficacement, mais qui ne s'inverse « facilement » que grâce à une information supplémentaire, la trappe. L'idée de la construction de Yao, pour chiffrer un bit $M \in \{0,1\}$ à partir d'une permutation à trappe f, est informellement la suivante :

1. Tirer un élément x uniformément dans l'ensemble de départ de la permutation à trappe.
2. Calculer $y = f(x)$.
3. Définir le chiffré comme $(C_1, C_2) = (y, \mathrm{hc}_f(x) \oplus M)$.

La fonction hc_f est ici un *bit difficile* pour la fonction f. C'est un prédicat booléen qui s'évalue efficacement à partir d'une entrée x de f, mais qu'il est difficile de calculer connaissant $f(x)$. Par exemple, considérons la fonction $f : x \mapsto g^x \pmod p$, où g est un générateur de $\mathbb{Z}_p^\star$, avec p premier : le bit de poids faible de x est difficile pour f. Plus formellement, cela signifie que pour tout attaquant probabiliste fonctionnant en temps polynomial $\mathcal{A}$, il existe une fonction négligeable negl tel que $\Pr(\mathcal{A}(f(x)) = \mathrm{hc}_f(x)) \leq \frac{1}{2} + \mathsf{negl}(\lambda)$.

Pour déchiffrer, le destinataire utilise sa trappe pour calculer $x = f^{-1}(C_1)$ et calcule $\mathrm{hc}_f(x) \oplus C_2 = M$.

Intuitivement, cette construction est sûre puisque pour un attaquant ne connaissant que $f(x)$ et f, la quantité $\mathrm{hc}_f(x)$ est pseudo-aléatoire (cela provient directement de la définition d'un bit difficile).

Même si cette construction est inefficace, elle illustre une approche très classique pour construire des systèmes de chiffrement, dans laquelle le message clair est masqué par un

élément produit de façon aléatoire, et reproductible par le destinataire possédant la clé secrète associée à la clé publique utilisée.

Il existe également des techniques génériques pour augmenter la sécurité d'un schéma de chiffrement. Par exemple, Dolev, Dwork et Naor ont conçu dans [27] une méthode permettant de transformer un schéma IND-CPA en un chiffrement IND-CCA. Pour cela, ils utilisent comme outils cryptographiques des signatures à usage unique et des preuves à divulgation nulle de connaissance non-interactives (NIZK pour *non-interactive zero-knowledge*). Le schéma résultant est assez inefficace. En particulier, il est nécessaire de produire 2λ paires de clé, puis de chiffrer λ fois le message sous des clés publiques spécifiquement choisies pour pouvoir être « figées » grâce à la signature à usage unique, puis de prouver grâce aux preuves NIZK que les messages chiffrés cachent tous le même message clair. Des améliorations de ce travail précurseur existent. Nous verrons une autre construction générique de chiffrement IND-CCA dans la Section 5.

4.2 Le schéma de chiffrement de Regev

Le premier chiffrement dont la sécurité repose sur la difficulté du problème LWE a été proposé par Regev [63]. La dimension n est le paramètre de sécurité de ce chiffrement. Les autres paramètres sont m, q et α. Dans la description qui suit, toutes les opérations se font modulo l'entier q.

Définition 4 (Chiffrement de Regev) *Soient n, m, et q des entiers avec q premier et $m \geq 4(n+1)\log_2 q$, et α un réel dans $]0, 1/(4m)[$.*

KeyGen **:** *La clé secrète est $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. La clé publique est un couple $(\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \mathbf{As}+\mathbf{e})$, où $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ et $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$.*

Enc **:** *Étant donné $M \in \{0,1\}$, échantillonner $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$ et renvoyer le chiffré*

$$(\mathbf{r}^T\mathbf{A}, \mathbf{r}^T\mathbf{b} + \lfloor q/2 \rfloor \cdot M) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

Dec **:** *Étant donné un chiffré $(\mathbf{u}^T, v) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, calculer $v - \mathbf{u}^T\mathbf{s}$: le clair est 0 si le résultat est plus proche de 0 que de $\lfloor q/2 \rfloor$, et 1 sinon.*

Remarque 2 *Comme nous allons le voir, les conditions sur les paramètres listées ci-dessus permettent de garantir la correction du protocole, et de prouver sa sécurité sous l'hypothèse que le problème LWE est difficile. Si l'on souhaite également utiliser le Théorème 1 pour faire reposer la sécurité sur l'hypothèse que le problème GapSVP est difficile, alors la condition $\alpha q \geq 2\sqrt{n}$ doit être ajoutée. Comme c'est souvent le cas en cryptographie reposant sur les réseaux, nous avons maintenant plusieurs contraintes sur différents paramètres, et, de prime abord, il peut ne pas sembler évident qu'il existe une solution. Ici, en combinant les inégalités, on obtient la condition $q > 8m\sqrt{n} \geq 32(n+1)\sqrt{n}\log_2 q$. Il suffit alors de fixer $q = \Theta(n^{3/2}\log n)$, $m = \Theta(n \log n)$ et $\alpha^{-1} = \Theta(n \log n)$.*

Le principe de ce chiffrement est de donner comme clé publique des échantillons $(\mathbf{a}_i, b_i)$ de la distribution $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$, et de conserver comme clé secrète le vecteur $\mathbf{s}$. Pour chiffrer un clair $M \in \{0,1\}$, on choisit un vecteur $\mathbf{r}$ uniformément dans $\{0,1\}^m$, ce qui correspond à choisir un sous-ensemble aléatoire des couples $(\mathbf{a}_i, b_i)$. On fait ensuite la somme les couples choisis, puis on ajoute $\lfloor q/2 \rfloor \cdot M$ à la somme des b_i. Pour déchiffrer un chiffré $(\mathbf{u}^T, v)$ correctement généré, on utilise la clé secrète $\mathbf{s}$ pour calculer $v - \mathbf{u}^T\mathbf{s} = \mathbf{r}^T\mathbf{e} + \lfloor q/2 \rfloor \cdot M$. Or le vecteur $\mathbf{e}$ est échantillonné selon une gaussienne discrète de paramètre $\alpha q \leq q/(4m)$, ainsi, en utilisant le Lemme 1, nous savons qu'avec probabilité proche de 1,

$$\Big| \sum_{i \leq m} r_i e_i \Big| \leq \|\mathbf{r}\| \cdot \|\mathbf{e}\| \leq \sqrt{m} \cdot \frac{q}{4\sqrt{m}} = \frac{q}{4}.$$

La valeur de $v - \mathbf{u}^T\mathbf{s}$ est donc soit proche de 0, soit proche de $\lfloor q/2 \rfloor$, ce qui permet de retrouver le clair M.

Remarque 3 *La fonction de chiffrement ne permet de chiffrer qu'un seul bit. Pour en chiffrer plusieurs, on peut soit l'utiliser plusieurs fois, soit utiliser plusieurs colonnes* $\mathbf{b}_i$ *au lieu d'une seule.*

Sécurité. Le chiffrement de Regev est IND-CPA sous l'hypothèse que LWE est difficile. Nous donnons ici une idée de la preuve, la preuve complète pouvant être lue dans [64]. Nous allons utiliser la difficulté présumée de LWE, et un outil supplémentaire, le *leftover hash lemma*. Ce dernier indique qu'une combinaison binaire uniforme des lignes d'une matrice uniformément distribuée dans $\mathbb{Z}_q^{m \times n}$ est elle-même quasiment uniformément distribuée. La version donnée ci-dessous est un cas très restreint d'un résultat de [37]. Le *leftover hash lemma* est un outil classique en cryptographie, qui sert par exemple à extraire de l'aléa uniformément distribué à partir d'une mauvaise source d'aléa.

Lemme 3 (*Leftover hash lemma*) *Supposons que* $m \geq 4n \log_2 q$ *et que* q *est premier. Soient* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ *et* $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$*. Alors la distribution de la paire* $(\mathbf{A}, \mathbf{Ar})$ *est à distance statistique* $\leq 2^{-n}$ *de la distribution* $U(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n)$*.*

L'hypothèse portant sur m est cruciale pour que le résultat soit correct. Quand $\mathbf{r}$ est choisi uniformément dans $\{0,1\}^m$, il y a 2^m possibilités pour $\mathbf{r}$, ce qui est bien plus élevé que le nombre de valeurs possibles pour le produit $\mathbf{Ar} \in \mathbb{Z}_q^n$. Le lemme nous assure que ces produits sont bien répartis. Dans la suite, nous raisonnerons comme si la distribution de la paire $(\mathbf{A}, \mathbf{Ar})$ était *exactement* la distribution $U(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n)$.

Nous avons maintenant tous les outils en main pour prouver que si le chiffrement de Regev n'est pas sûr contre les attaques à clair choisi, c'est-à-dire s'il existe un attaquant efficace qui peut distinguer entre les chiffrés de 0 et de 1 en temps polynomial avec un avantage non-négligeable, alors il est possible d'utiliser cet attaquant pour distinguer les distributions $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$ et $U(\mathbb{Z}_q^{n+1})$ avec probabilité non-négligeable vis-à-vis du choix de

$\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, et donc de résoudre la version décisionnelle de LWE. On suppose ainsi qu'il existe un attaquant probabiliste polynomial $\mathcal{A}$ qui, étant donné la clé publique $(\mathbf{A}, \mathbf{b})$ échantillonnée selon $(D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s}))^m$, peut trouver le bit chiffré avec probabilité au moins $1/2 + 1/\text{poly}(n)$ pour un sous-ensemble de mesure non-négligeable des secrets $\mathbf{s}$.

Maintenant, supposons que l'on donne en entrée à cet algorithme $\mathcal{A}$, non pas un échantillon de LWE, mais un couple $(\mathbf{A}, \mathbf{b}) \hookleftarrow U(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m)$ comme clé publique, ainsi que le chiffré d'un bit aléatoire, qui a été chiffré en utilisant ce couple. D'après le Lemme 3 (utilisé en dimension $n+1$), la distribution de $(\mathbf{A}, \mathbf{b}, \mathbf{r}^T\mathbf{A}, \mathbf{r}^T\mathbf{b})$ est essentiellement uniforme. Ainsi, dans ce cas, les chiffrés de 0 et de 1 suivent (essentiellement) la même distribution, la distribution uniforme, et sont donc calculatoirement indistinguables (en particulier pour $\mathcal{A}$).

Une instance de la version décisionnelle du problème LWE est une paire $(\mathbf{A}, \mathbf{b})$ distribuée soit uniformément, soit selon $(D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s}))^m$ avec $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$ choisi une fois pour toutes. Il suffit donc de donner l'instance du problème comme clé publique, et de l'utiliser pour chiffrer un bit, que l'on envoie à l'algorithme $\mathcal{A}$. Si $\mathcal{A}$ trouve le bit chiffré avec une probabilité suffisante, alors on sait qu'il s'agissait d'un échantillon de $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$.

4.3 Le schéma de chiffrement dual-Regev

En 2008, Gentry, Peikert et Vaikuntanathan [30] ont proposé un autre chiffrement, appelé *dual-Regev*. L'adjectif dual provient du fait que ce schéma de chiffrement peut être obtenu en ≪ échangeant ≫les algorithmes KeyGen et Enc, comme illustré par la Figure 6. Ce chiffrement a ensuite été utilisé pour construire des chiffrements plus avancés, comme nous le verrons dans les sections suivantes.

Définition 5 (Chiffrement dual-Regev) *Soient n, m, et q des entiers avec q premier et $m \geq 4n + \log_2 q$, et α un réel dans $]0, 1/(8m)[$. Les utilisateurs partagent une matrice $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$.*

KeyGen **:** *La clé secrète est $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$. La clé publique est $\mathbf{y}^T = \mathbf{r}^T\mathbf{A} \bmod q$.*

Enc **:** *Étant donné $M \in \{0,1\}$, échantillonner $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ et $e' \hookleftarrow D_{\mathbb{Z},\alpha q}$. Le chiffré est*

$$(\mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M) \in \mathbb{Z}_q^m \times \mathbb{Z}_q.$$

Dec **:** *Etant donné un chiffré $(\mathbf{b}, c)$, calculer $c - \mathbf{r}^T\mathbf{b}$: le clair est 0 si le résultat est plus proche de 0 que de $\lfloor q/2 \rfloor$, et 1 sinon.*

Le principe ici est de donner comme clé publique $\mathbf{y}^T = \mathbf{r}^T\mathbf{A} \bmod q$ et de garder comme clé secrète le vecteur $\mathbf{r}$. On utilise ensuite LWE pour chiffrer le message : on prend un $\mathbf{s}$ uniforme, et le chiffré $(\mathbf{b}, c)$ correspond à $m+1$ membres droits d'échantillons, avec les lignes de $\mathbf{A}$ et le vecteur $\mathbf{y}$ comme membres gauches. Pour déchiffrer $(\mathbf{b}, c)$, il suffit de calculer :

$$c - \mathbf{r}^T\mathbf{b} = \mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M - \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = e' - \mathbf{r}^T\mathbf{e} + \lfloor q/2 \rfloor \cdot M.$$

Comme pour le chiffrement de Regev, le paramètre α a été choisi pour que $e' - \mathbf{r}^T\mathbf{e}$ soit petit devant $q/2$, ainsi, la valeur de $c - \mathbf{r}^T\mathbf{b}$ permet de retrouver M.

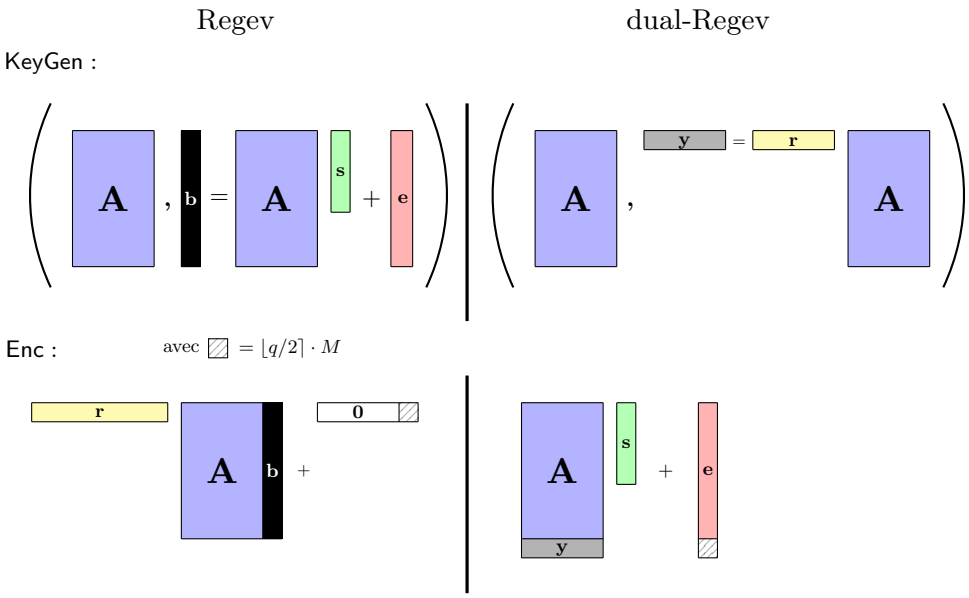

Figure 6. Comparaison du chiffrement de Regev et du chiffrement dual-Regev.

Sécurité. Le schéma de chiffrement dual-Regev est IND-CPA. La preuve complète est disponible dans [30]. Les arguments de sécurité sont les mêmes que pour le schéma de chiffrement de Regev, mais utilisés dans un ordre différent.

Le *leftover hash lemma* garantit que la distribution des données publiques $(\mathbf{A}, \mathbf{y})$ est essentiellement uniforme. Ainsi, le masque $(\mathbf{As} + \mathbf{e}, \mathbf{y}^T\mathbf{s} + e')$ utilisé dans la procédure de chiffrement est essentiellement distribué comme $m + 1$ échantillons de la distribution $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$ (pour un vecteur $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$).

Étant donné un chiffré $(\mathbf{b}, c)$, la distribution de $(\mathbf{A}, \mathbf{b}, \mathbf{y}, c)$ est calculatoirement indistinguable de la distribution uniforme, sous l'hypothèse que le problème LWE est difficile.

Remarque 4 *Une propriété intéressante des chiffrements de Regev et dual-Regev est leur propriété d'*anonymat*, sous l'hypothèse de difficulté de LWE. Pour un chiffrement à clé publique, Bellare* et al. *ont défini dans [11] l'anonymat (ou* key-privacy*) comme le fait pour un chiffré de ne pas dévoiler d'information sur la clé publique avec laquelle il a été produit. Plus précisément, l'attaque contre l'anonymat est décrite par un jeu dans lequel l'attaquant*

reçoit deux clés publiques pk_0 et pk_1, puis propose à son challenger un message M. Celui-ci lui répond par le chiffré de M sous l'une des deux clés publique. Un protocole de chiffrement à clé publique est anonyme si tout attaquant polynomial a un avantage négligeable pour retrouver la clé publique.

Remarque 5 *Les deux schémas de chiffrement décrits sont IND-CPA mais ne sont pas IND-CCA (c'est un exercice simple, pour lequel on peut par exemple d'inspirer de la Remarque 1). Il existe cependant plusieurs schémas de chiffrement qui sont IND-CCA sous l'hypothèse que LWE est difficile (citons par exemple [59,52,2]).*

5 Chiffrement fondé sur l'identité

Comme à la section précédente, nous commençons par définir la notion de chiffrement fondé sur l'identité, avant de construire un tel chiffrement à partir du problème LWE.

5.1 Définitions du chiffrement et de sa sécurité

L'un des inconvénients majeurs de la cryptographie à clé publique réside dans la gestion des clés publiques. En effet, celles-ci doivent être authentifiées pour pouvoir être utilisées sans danger. Une solution, qui a elle-même ses inconvénients, est de faire confiance à une autorité de certification qui va délivrer un certificat pour chaque clé publique de chaque utilisateur. Plus généralement, la mise en place d'une *infrastructure à clé publique* (PKI, pour *Public Key Infrastructure*) permet de gérer les arrivées et départs d'utilisateurs, les dates de péremption des clés, les renouvellements de certificats, les révocations, etc. Pour simplifier cette procédure, Shamir a proposé, en 1984, le principe de la cryptographie *fondée sur l'identité* [68]. Dans un schéma de chiffrement à clé publique, cette dernière est en général un élément aléatoire, dont on ne contrôle pas la valeur. Au contraire, dans le cas de la cryptographie fondée sur l'identité, la donnée publique, associée à chaque utilisateur pour leur chiffrer un message, est une fonction déterministe de leur identité, c'est-à-dire, un élément les représentant de façon non ambiguë (comme un numéro de sécurité sociale, une adresse email, etc.). L'autre différence majeure concerne la génération des clés secrètes. Celles-ci doivent donc être calculées à partir de l'identité d'une personne : un utilisateur ne peut pas, par conséquent, créer lui-même sa clé secrète (toute autre personne pourrait la créer également, puisque l'identité est publique) : il doit donc faire appel à une autorité de confiance. Ceci est bien sûr critique du point de vue de la vie privée des utilisateurs, puisque cette autorité possède toutes leurs clés secrètes. Ce séquestre de clés est parfois souhaité, dans le cas contraire, il existe des techniques pour partager la génération des secrets entre plusieurs entités de confiance.

On définit formellement un protocole de chiffrement fondé sur l'identité de la façon suivante.

Définition 6 (Chiffrement fondé sur l'identité) *Soient λ un entier, le paramètre de sécurité, et $\mathcal{ID} = \mathcal{ID}(\lambda)$ un ensemble possible d'identités. Un* protocole de chiffrement fondé sur l'identité *Π est la donnée de quatre algorithmes probabilistes polynomiaux :*

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**.** *Cet algorithme prend en entrée λ, et renvoie les paramètres publics du système* mpk*, et une clé secrète maîtresse* msk*.*

$\mathsf{Extract}(1^\lambda, \mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \to sk_{\mathsf{id}}$**.** *L'algorithme d'extraction de clé (secrète) prend en entrée le paramètre de sécurité λ, les paramètres du système* mpk*, une clé secrète maîtresse* msk *et une identité* $\mathsf{id} \in \mathcal{ID}$*. Il renvoie la clé secrète sk_{id} associée à l'identité* id*.*

$\mathsf{Enc}(1^\lambda, \mathsf{mpk}, \mathsf{id}, M) \to C$**.** *L'algorithme de chiffrement prend en entrée le paramètre de sécurité λ, les paramètres du système* mpk*, une identité* $\mathsf{id} \in \mathcal{ID}$ *et un message $M \in \{0,1\}^*$. Il renvoie un chiffré C.*

$\mathsf{Dec}(1^\lambda, \mathsf{mpk}, sk_{\mathsf{id}}, C) \to \{M, \perp\}$**.** *L'algorithme de déchiffrement prend en entrée le paramètre de sécurité λ, les paramètres du système* mpk*, une clé secrète sk_{id} et un chiffré C. Il renvoie soit un message M, soit un symbole $\perp$ qui indique que le chiffré n'est pas valide.*

Le protocole de chiffrement fondé sur l'identité doit être correct*, ce qui signifie pour tout λ suffisamment grand, pour toute identité* $\mathsf{id} \in \mathcal{ID}$ *et pour tout message $M \in \{0,1\}^*$, si* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*, alors*

$$\mathsf{Dec}(1^\lambda, \mathsf{mpk}, \mathsf{Extract}(1^\lambda, \mathsf{mpk}, \mathsf{msk}, \mathsf{id}), \mathsf{Enc}(1^\lambda, \mathsf{mpk}, \mathsf{id}, M)) = M,$$

avec probabilité (prise sur tous les aléas internes des algorithmes) négligemment proche de 1*.*

Shamir, dans son article fondateur, a non seulement inventé le concept de cryptographie fondée sur l'identité, mais il a également proposé une signature numérique réalisant ce paradigme. La conception d'un mécanisme de chiffrement est restée ouverte jusqu'en 2001, année au cours de laquelle Boneh et Franklin ont proposé le premier protocole de chiffrement fondé sur l'identité [15], à l'aide de couplages sur des courbes elliptiques. Cocks [24] a parallèlement proposé une autre protocole, plus inefficace, utilisant les résidus quadratiques.

Sécurité du chiffrement fondé sur l'identité. Comme pour le chiffrement classique, la principale exigence de sécurité d'un IBE concerne la confidentialité, et la bonne notion à atteindre est donc l'indistinguabilité des chiffrés. Cependant, une différence notable provient du fait qu'on va donner à l'attaquant un accès à un oracle d'extraction de clé secrète. Cet oracle répond à une requête sur une identité id, par la clé secrète $\mathsf{sk}_{\mathsf{id}} = \mathsf{Extract}(1^\lambda, \mathsf{mpk}, \mathsf{msk}, \mathsf{id})$. Par ailleurs, l'attaquant va *choisir* l'identité qu'il attaque. Ce choix peut se faire au début de l'attaque – on parle de sécurité *sélective* [13,20], ou au moment du choix des messages du challenge d'indistinguabilité – la sécurité est alors dite *adaptative*. Notons qu'il est possible de passer d'une sécurité sélective à une sécurité adaptative, si l'on accepte de dégrader (significativement) la preuve de sécurité : le challenger peut tenter de deviner à l'avance l'identité que l'attaquant va choisir ; ce choix est correct avec probabilité $1/|\mathcal{ID}|$.

Définition 7 (IND-ID-CPA) *Soit* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *un protocole de chiffrement fondé sur l'identité. Soit* λ *le paramètre de sécurité. Soit* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *un adversaire fonctionnant en deux étapes ayant accès à un oracle d'extraction* $\mathcal{O}_{\mathsf{Extract}}(\cdot)$*. Considérons le jeu suivant :*

$\boxed{\textit{Expérience } \mathbf{Exp}_{\Pi}^{\mathsf{ind-id-cpa}}(\mathcal{A}, \mathcal{ID})}$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
$(M_0, M_1, \mathsf{id}_{ch}, st) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Extract}}(\cdot)}(1^\lambda, \mathsf{mpk})$
$b^\star \hookleftarrow U(\{0,1\})$
$C^\star \leftarrow \Pi.\mathsf{Enc}(1^\lambda, \mathsf{mpk}, \mathsf{id}_{ch}, M_{b^\star})$
$b \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{Extract}}(\cdot)}(1^\lambda, C^\star, st)$
Renvoie 1 si $b = b^\star$ *et 0 sinon*

Au cours de ces deux étapes, une restriction naturelle est nécessaire : l'attaquant ne peut pas interroger l'oracle d'extraction $\mathcal{O}_{\mathsf{Extract}}$ *sur l'identité challenge* id_{ch}*. L'*avantage *de l'attaquant est défini par :*

$$\mathsf{Adv}_{\Pi}^{\mathsf{ind-id-cpa}}(\mathcal{A}, \mathcal{ID}) = |\Pr(\mathbf{Exp}_{\Pi}^{\mathsf{ind-id-cpa}}(\mathcal{A}, \mathcal{ID}) = 1) - 1/2|.$$

Le schéma Π *est considéré sûr du point de vue de l'indistinguabilité des chiffrés dans une attaque à clairs choisis si tout attaquant probabiliste polynomial tel que décrit ci-dessus a un avantage négligeable.*

Comme pour le chiffrement à clé publique, il est possible de définir la notion d'indistinguabilité des chiffrés dans une attaque à chiffrés choisis. Une autre notion de sécurité importante pour le chiffrement fondé sur l'identité est celle d'*anonymat*. Elle assure qu'un chiffré ne dévoile pas d'information sur l'identité du destinataire. Cette notion correspond d'ailleurs à celle de *key-privacy* pour le chiffrement à clé publique [11]. Elle a été formalisée dans [1] dans un scénario adaptatif (les liens entre attaques adaptative et sélective pour l'indistinguabilité et l'anonymat sont étudiés dans [35]) . Nous ne formaliserons pas cette notion, qui porte également tout son sens dans le cas des chiffrements par attributs, comme nous le verrons dans la suite.

Remarque 6 *Une application intéressante des IBE anonymes concerne la recherche de mots-clés dans un chiffré [1]. S'il est a priori impossible de savoir si un mot particulier apparaît dans un message chiffré, des applications à la gestion des emails, par exemple, nécessitent une telle propriété. Supposons que* Π *est un IBE anonyme. L'idée pour permettre cette recherche est d'associer à un mot particulier* ω*, une trappe calculée grâce à l'extraction de clé sur l'identité* ω *:* $T_\omega = \Pi.\mathsf{Extract}(1^\lambda, \mathsf{mpk}, \mathsf{msk}, \omega)$*. Pour permettre la recherche dans un chiffré de ce mot, le chiffré est accompagné du tag* (C_ω, R) *avec* R *un message tiré uniformément et* $C_\omega = \Pi.\mathsf{Enc}(1^\lambda, \mathsf{mpk}, \omega, R)$*. Étant donné un message chiffré et son tag* $(\tilde{C}, \tilde{R})$*, pour vérifier que le mot apparaît dans le chiffré, il suffit de tester si*

$\Pi.\mathsf{Dec}(1^\lambda, \mathsf{mpk}, T_\omega, \tilde{C}) = \tilde{R}$. *Dans [1], Abdalla et al. ont démontré que si l'ABE est anonyme, le protocole ainsi obtenu est sûr.*

On peut noter que, réciproquement, l'existence de schéma de chiffrement permettant la recherche de mot clé implique l'existence de chiffrement fondé sur l'identité [14,1].

Un protocole de chiffrement fondé sur l'identité possédant une sécurité sélective IND-CPA permet d'obtenir des protocoles de chiffrement à clé publique atteignant le niveau de sécurité IND-CCA. La construction suivante, due à Canetti, Halevi et Katz [21,12] , utilise un protocole de chiffrement fondé sur l'identité $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ et un protocole de signature à usage unique $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$. Un protocole de signature à usage unique [38] permet à un utilisateur d'authentifier un message (et pas forcément plus) à l'aide d'une clé secrète, auprès de n'importe quel récepteur possédant une clé de vérification associée. L'existence d'un protocole de signature à usage unique n'impose pas d'hypothèse supplémentaire, puisqu'on peut en obtenir un à partir d'une fonction à sens unique, et, a fortiori, d'un chiffrement fondé sur l'identité. Le principe de la construction est le suivant. La clé publique du nouveau protocole est la donnée publique mpk obtenue à partir de l'exécution de $\Pi.\mathsf{Setup}(1^\lambda)$, et la clé secrète la clé maîtresse msk correspondante. Pour chiffrer un message M à partir de mpk, on utilise l'algorithme de chiffrement $\Pi.\mathsf{Enc}$ avec comme identité une clé de vérification vk fraîche produite par $\Sigma.\mathsf{KeyGen}(1^\lambda)$; si $C = \Pi.\mathsf{Enc}(1^\lambda, \mathsf{mpk}, vk, M)$ est le chiffré, la clé secrète de signature sk est alors utilisée pour *signer* ce chiffré C, et la clé de vérification vk et la signature accompagneront ce dernier lors de la transmission. Pour déchiffrer, le récepteur utilise $\Pi.\mathsf{Dec}$ pour retrouver le clair M et $\Sigma.\mathsf{Verify}$ pour vérifier la signature : si la signature est correcte, le récepteur conserve M, et sinon, le récepteur considère le chiffré comme étant frauduleux. Cette signature, différente à chaque chiffrement, permet de « figer »le chiffré, de sorte à ce que les requêtes de déchiffrement du jeu de sécurité IND-CCA ne fournissent pas d'information à l'attaquant.

Une version *hiérarchique* du chiffrement fondé sur l'identité a été proposée par Horwitz et Lynn dans [36]. Cette généralisation du concept de chiffrement fondé sur l'identité permet de simuler une organisation hiérarchique, décrite par exemple par un arbre. Une identité de niveau k dans l'arbre de hiérarchie peut générer les clés secrètes de tous ses descendants, mais il ne peut pas déchiffrer les messages destinés aux autres identités. Cette extension est intéressante en elle-même. En outre, les constructions présentées dans [21,12] s'appliquent encore. Elles permettent d'obtenir des chiffrements fondés sur l'identité (et hiérarchiques) sûrs du points de vue CCA, à partir de d'(H)IBE seulement CPA.

5.2 Une trappe pour LWE

Tous les protocoles de chiffrement fondés sur l'identité construits à partir de LWE et connus à ce jour [30,22,2,3,52] utilisent la notion de trappe *complète* pour LWE [7,8]. Les schémas de chiffrements proposés à la section précédente contiennent déjà implicitement

des trappes liées à LWE, puisqu'ils contiennent des algorithmes de déchiffrement. Dans le cas du chiffrement de Regev, il s'agit du vecteur $\mathbf{s}$ utilisé pour les échantillons : cette clé secrète n'est une trappe que pour $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$, et ne fonctionnerait pas si l'on avait utilisé un autre secret $\mathbf{s}'$. Dans le cas du chiffrement dual-Regev, il s'agit du vecteur $\mathbf{r}$ utilisé dans le *leftover hash lemma* : ce vecteur permet de résoudre la version décisionnelle de LWE (distinguer entre une distribution $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$, pour $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, et la distribution uniforme),mais non la version calculatoire (consistant à retrouver $\mathbf{s}$). La trappe *complète*, construite avec les membres gauches $\mathbf{a}_i$ des échantillons, permet de retrouver $\mathbf{s}$ efficacement à partir des membres droits des échantillons de $D^{\text{LWE}}_{n,q,\alpha}(\mathbf{s})$, pour tout secret $\mathbf{s}$. Elle permet également d'exprimer n'importe quel vecteur de $\mathbb{Z}_q^n$ comme combinaison linéaire à petits coefficients des $\mathbf{a}_i$. C'est cette dernière propriété qui s'avérera particulièrement utile à la sous-section suivante, pour construire un chiffrement fondé sur l'identité.

Pour $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, que l'on peut imaginer formée par les $\mathbf{a}_i$ d'échantillons LWE successifs, on définit le réseau $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A} = \mathbf{0}^T \bmod q\}$. Il s'agit bien d'un réseau (une base peut être calculée à partir de $\mathbf{A}$ en utilisant la forme normale de Hermite), et il a pour dimension m (il contient $q\mathbb{Z}^m$). À partir de $\mathbf{A}$, il est a priori difficile de trouver des vecteurs courts de $\Lambda_q^\perp(\mathbf{A})$, car cela permettrait de résoudre la version décisionnelle de LWE (le déchiffrement de dual-Regev repose sur ce principe). Par contre, il est possible d'échantillonner $\mathbf{A}$ et une base courte de $\Lambda_q^\perp(\mathbf{A})$ simultanément. Un exemple est donné à la Figure 7.

Lemme 4 ([7,8]) *Il existe un algorithme probabiliste polynomial* GenBasis *qui prend en entrée des entiers n, m et $q \geq 2$ (écrits en unaire) tels que $m \geq \Omega(n \log q)$ et qui renvoie deux matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ et $\mathbf{S} \in \mathbb{Z}^{m \times m}$ telles que : la distribution de $\mathbf{A}$ est à distance statistique négligeable de la distribution uniforme, les lignes $\mathbf{s}_i$ de $\mathbf{S}$ forment une base de $\Lambda_q^\perp(\mathbf{A})$, et $\max_i \|\mathbf{s}_i\| \leq O(\sqrt{n \log q})$.*

L'algorithme GenBasis repose sur le *leftover hash lemma* (Lemme 3), qui permet de créer de nouvelles lignes de $\mathbf{A}$ en même temps que des lignes de $\mathbf{S}$.

$$\begin{bmatrix} 2 & -3 & 3 & 0 & -1 & 1 & 4 & -3 \\ 3 & -2 & -4 & -1 & -2 & -2 & 4 & 2 \\ 0 & 4 & 4 & -1 & 4 & -3 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 8 & 0 & 0 \\ 5 & 4 & 2 & -4 & -4 & -2 & 1 & -3 \\ -5 & 1 & 3 & -1 & -3 & 4 & 6 & 2 \\ 1 & 3 & -6 & 6 & 4 & -5 & 1 & -2 \\ -4 & 3 & -4 & -7 & -1 & -2 & 3 & -6 \end{bmatrix} \cdot \begin{bmatrix} 185 & 97 & 202 \\ 146 & 148 & 11 \\ 208 & 219 & 164 \\ 218 & 173 & 117 \\ 211 & 176 & 187 \\ 79 & 255 & 112 \\ 47 & 136 & 232 \\ 204 & 172 & 58 \end{bmatrix} = \mathbf{0} \bmod 257.$$

Figure 7. Une matrice $\mathbf{A}$ et une petite base de $\Lambda_q^\perp(\mathbf{A})$, pour $q = 257$, $n = 3$ et $m = 8$.

À l'aide de la trappe complète on peut en effet retrouver $\mathbf{s}$ à partir des membres droits d'échantillons de $D^{\mathrm{LWE}}_{n,q,\alpha}(\mathbf{s})$, pour tout $\mathbf{s}$, si le paramètre α est au plus de l'ordre de $1/\sqrt{mn\log q}$. Il suffit de multiplier, modulo q, le vecteur correspondant $\mathbf{b} = \mathbf{As} + \mathbf{e}$ à gauche par $\mathbf{S}$, ce qui permet d'obtenir $\mathbf{Se}$ exactement (sans modulo, car chacune de ses entrées est petite par rapport à q), puis $\mathbf{e}$ en multipliant par l'inverse rationnelle de $\mathbf{S}$, et enfin $\mathbf{s}$ par résolution de système linéaire sur-contraint non bruité.

En combinant la trappe avec l'algorithme du Lemme 2, on peut, à partir de $\mathbf{A}$, $\mathbf{S}$ et $\mathbf{y} \in \mathbb{Z}_q^n$, échantillonner $\mathbf{x} \in \mathbb{Z}^m$ gaussien de paramètre $s = \Omega(\sqrt{n \log q \log m})$ tel que $\mathbf{x}^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$. Il suffit pour cela de trouver (par algèbre linéaire) un vecteur $\mathbf{x}_0 \in \mathbb{Z}^m$ tel que $\mathbf{x_0}^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$, d'échantillonner $\mathbf{x}_1 \hookleftarrow D_{\Lambda_q^\perp(\mathbf{A}),s,-\mathbf{x}_0}$, et de renvoyer $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1$.

Enfin, si l'on dispose d'une paire $(\mathbf{A}, S) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}^{m\times m}$ où $\mathbf{S}$ est une base courte de $\Lambda_q^\perp(\mathbf{A})$, alors pour toute extension verticale de $\mathbf{A}$ en une matrice $\mathbf{B} \in \mathbb{Z}_q^{m'\times n}$ dont les m premières lignes sont celles de $\mathbf{A}$, on peut efficacement calculer une trappe complète $\mathbf{T} \in \mathbb{Z}^{m'\times m'}$ de $\mathbf{B}$. En effet, pour i de $m+1$ à m', on peut utiliser l'algorithme précédent afin d'échantillonner $\mathbf{s}_i$ petit tel que $-\mathbf{s}_i^T\mathbf{A}$ est égal à la i-ème ligne de $\mathbf{B}$. On a alors, modulo q :

$$\left[\begin{array}{c|c} \mathbf{S} & \mathbf{0} \\ \hline \mathbf{s}_{m+1}^T & \\ \cdots & I_{m'-m+1} \\ \mathbf{s}_{m'}^T & \end{array}\right] \cdot \mathbf{B} = \mathbf{0},$$

et on peut vérifier que la matrice à gauche du produit est une base du réseau $\Lambda_q^\perp(\mathbf{B})$. Nous appellerons cet algorithme $\mathsf{ExtBasis}$.

5.3 Le protocole de Cash, Hofheinz, Kiltz et Peikert [22]

Les paramètres du chiffrement fondé sur l'identité de [22] sont les mêmes que ceux des chiffrements de la Section 4 (n, m, q et α), auxquels on ajoute un paramètre ℓ qui détermine l'ensemble des identités $\mathcal{ID} = \{0,1\}^\ell$, et un paramètre s quantifiant la taille des clés secrètes des utilisateurs.

Définition 8 (Protocole CHKP) *Soient n, m, q et ℓ des entiers tels que $m \geq \Omega(n \log q)$, et deux réels α et s tels que $s \geq \Omega(\sqrt{mn\log q}\log(m\ell))$ et $0 < \alpha \leq O(q/sm\ell)$.*

Setup **:** *Créer* $(\mathbf{A}_0, \mathbf{S}_0) \leftarrow \mathsf{GenBasis}(1^n, 1^m, q)$.
Pour tout $1 \leq i \leq \ell$ *et* $b \in \{0,1\}$, *échantillonner* $\mathbf{A}_i^{(b)} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$.
Échantillonner $\mathbf{y} \hookleftarrow U(\mathbb{Z}_q^n)$.
Renvoyer $\mathsf{mpk} = (\mathbf{A}_0, (\mathbf{A}_i^{(b)})_{b,i}, \mathbf{y})$ *et* $\mathsf{msk} = \mathbf{S}_0$.
La clé publique d'un utilisateur d'identité $\mathsf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell) \in \{0,1\}^\ell$ *est la matrice* $\mathbf{A}_{\mathsf{id}}$

suivante :

$$\mathbf{A}_{\mathsf{id}} = \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^{(\mathsf{id}_1)} \\ \hline \cdots \\ \hline \mathbf{A}_\ell^{(\mathsf{id}_\ell)} \end{array}\right] \in \mathbb{Z}_q^{(\ell+1)m \times n}.$$

Extract : *Calculer* $\mathbf{A}_{\mathsf{id}}$ *à partir de* mpk *et* id*, puis créer* $\mathbf{S}_{\mathsf{id}} \leftarrow \mathsf{ExtBasis}(\mathbf{S}_0, \mathbf{A}_{\mathsf{id}})$*, qui étend la trappe* $\mathbf{S}_0$ *de* $\mathbf{A}_0$ *en une trappe complète pour* $\mathbf{A}_{\mathsf{id}}$*. À l'aide de* $\mathbf{S}_{\mathsf{id}}$*, échantillonner* $\mathbf{x}_{\mathsf{id}} \in \mathbb{Z}^{(m+1)\ell}$ *gaussien de paramètre* s *tel que* $\mathbf{x}_{\mathsf{id}}^T \mathbf{A}_{\mathsf{id}} = \mathbf{y}^T$ *(comme expliqué à la sous-section précédente). La clé secrète de l'utilisateur* id *est* $\mathsf{sk}_{\mathsf{id}} = \mathbf{x}_{\mathsf{id}}$.

Enc : *Pour chiffrer* $M \in \{0,1\}$ *pour l'utilisateur d'identité* id*, échantillonner* $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$*,* $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^{(m+1)\ell}, \alpha q}$ *et* $e' \hookleftarrow D_{\mathbb{Z}, \alpha q}$*. Le chiffré est* $(\mathbf{A}_{\mathsf{id}}\mathbf{s} + \mathbf{e}, \mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M) \in \mathbb{Z}_q^{(\ell+1)m} \times \mathbb{Z}_q$.

Dec : *Étant donnés un chiffré* $(\mathbf{b}, c)$ *et la clé secrète* $\mathbf{x}_{\mathsf{id}}$*, calculer* $c - \mathbf{x}_{\mathsf{id}}^T\mathbf{b}$ *: le clair est* 0 *si* b *est plus proche de* 0 *que de* $\lfloor q/2 \rfloor$*, et* 1 *sinon.*

Le principe de ce protocole est d'avoir un ensemble de matrices $\mathbf{A}_{\mathsf{id}_1 \ldots \mathsf{id}_k}$ pour $0 \leq k \leq \ell$ tel que l'autorité connaît une trappe complète $\mathbf{S}_0$ pour la matrice $\mathbf{A}_0$ à la racine. La Figure 8 illustre comment ces matrices sont construites.

À l'aide de l'algorithme ExtBasis présenté à la sous-section précédente, l'autorité peut créer une trappe complète $\mathbf{S}_{\mathsf{id}}$ pour chaque matrice $\mathbf{A}_{\mathsf{id}}$. Grâce à $\mathbf{S}_{\mathsf{id}}$, l'autorité peut échantillonner, à l'aide de l'algorithme du Lemme 2, un vecteur court $\mathbf{x}_{\mathsf{id}}$ tel que $\mathbf{x}_{\mathsf{id}}^T \mathbf{A}_{\mathsf{id}} = \mathbf{y}^T$, et ainsi extraire une clé secrète $\mathbf{x}_{\mathsf{id}}$ associée à $\mathbf{y}$. Cette clé secrète $\mathbf{x}_{\mathsf{id}}$ et le vecteur public $\mathbf{y}$ jouent exactement les rôles respectifs de la clé secrète et de la clé publique du chiffrement dual-Regev. Les procédures de chiffrement et de déchiffrement sont d'ailleurs identiques à celles du chiffrement dual-Regev, avec comme clé publique $(\mathbf{A}_{\mathsf{id}}, \mathbf{y})$ et comme clé secrète $\mathbf{x}_{\mathsf{id}}$. La correction du protocole se prouve comme celle du chiffrement dual-Regev.

Sécurité. Ce chiffrement fondé sur l'identité admet une sécurité sélective IND-ID-CPA. On peut réduire la sécurité du chiffrement dual-Regev à celle du protocole CHKP : on va montrer que s'il existe un adversaire probabiliste polynomial $\mathcal{A}$ contre le protocole CHKP, alors on peut utiliser $\mathcal{A}$ pour construire un algorithme probabiliste polynomial $\mathcal{B}$ capable de distinguer les chiffrés de 0 et de 1 dans le chiffrement dual-Regev, pour les paramètres n, $(\ell+1)m$, q et α.

L'algorithme $\mathcal{B}$ prend comme entrée une instance de dual-Regev, c'est-à-dire une clé publique $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{(\ell+1)m \times n} \times \mathbb{Z}_q^n$, et un chiffré $(\mathbf{b}, c) \in \mathbb{Z}_q^{(\ell+1)m} \times \mathbb{Z}_q$. Son objectif est de déterminer si c'est un chiffré de 0 ou de 1, avec un avantage non-négligeable.

À partir de cette instance de dual-Regev, on construit une instance de CHKP, pour pouvoir utiliser l'attaquant $\mathcal{A}$. Tout d'abord, comme nous nous sommes placés dans le cas d'une sécurité sélective, l'attaquant $\mathcal{A}$ fournit à $\mathcal{B}$ l'identité challenge $\mathsf{id}^\star$. Ensuite, l'algorithme $\mathcal{B}$ va produire les données publiques et la clé secrète maîtresse de CHKP.

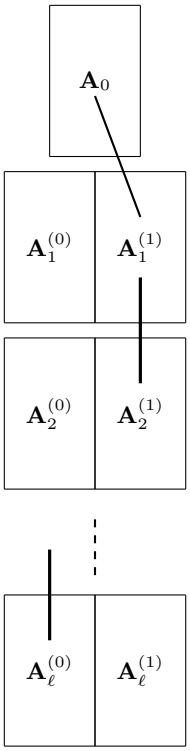

Figure 8. Clé publique associée à l'identité $\mathsf{id} = 11\cdots00$.

Pour cela, il décompose la matrice $\mathbf{A}$ en $\ell + 1$ matrices de tailles $m \times n$ qu'il appelle $\mathbf{A}_0, \mathbf{A}_1^{(\mathsf{id}_1^\star)}, \ldots, \mathbf{A}_\ell^{(\mathsf{id}_\ell^\star)}$. Pour tout $1 \leq i \leq \ell$, l'algorithme $\mathcal{B}$ génère $\mathbf{A}_i^{(1-\mathsf{id}^\star)}$ avec une trappe complète associée S_i, en utilisant l'algorithme $\mathsf{GenBasis}$. L'algorithme $\mathcal{B}$ donne alors à l'attaquant $\mathcal{A}$ les données publiques $\mathsf{mpk} = (\mathbf{A}_0, (\mathbf{A}_i^{(b)})_{b,i}, \mathbf{y})$ ainsi que le chiffré $(\mathbf{b}, c)$.

L'attaquant $\mathcal{A}$ a la possibilité de faire des requêtes d'extraction, pour des identités autres que $\mathsf{id}^\star$. L'algorithme $\mathcal{B}$ doit y répondre, et ses réponses doivent être indistinguables de celles qui correspondent à l'attaque réelle. Pour une identité $\mathsf{id} \neq \mathsf{id}^\star$, l'algorithme $\mathcal{B}$ peut calculer une trappe $\mathbf{S}_{\mathsf{id}}$ pour la matrice $\mathbf{A}_{\mathsf{id}}$ (construite à partir des matrices $\mathbf{A}_i^{(\mathsf{id}_i)}$), en utilisant la première trappe qu'il trouve parmi les trappes des $\mathbf{A}_i^{(1-\mathsf{id}^\star)}$. Il procède exactement comme dans l'algorithme d'extraction de clé, mais utilise une autre $\mathbf{S}_i$ à la place de $\mathbf{S}_0$. Comme $\mathsf{id} \neq \mathsf{id}^\star$, une telle trappe $\mathbf{S}_i$ existe. L'algorithme $\mathcal{B}$ peut ensuite échantillonner, grâce à cette trappe, un vecteur court $\mathbf{x}_{\mathsf{id}}$ dont la distribution est exactement celle de la clé secrète de l'utilisateur id dans le protocole réel.

Enfin, l'attaquant $\mathcal{A}$ renvoie un bit, que l'algorithme $\mathcal{B}$ retransmet comme sortie de l'attaque de l'instance du chiffrement dual-Regev. On peut montrer que l'algorithme $\mathcal{B}$ permet bien d'attaquer le chiffrement dual-Regev, dans le sens de la sécurité IND-CPA.

6 Chiffrement par attributs

Le chiffrement par attributs est une instance d'un concept de chiffrement beaucoup plus sophistiqué que le chiffrement à clé publique : le chiffrement *fonctionnel*, introduit par Sahai et Waters [66] en 2005, et formalisé par Boneh, Sahai et Waters en 2011 [17]. Contrairement au chiffrement à clé publique orienté vers une communication point-à-point, le chiffrement fonctionnel a vocation d'autoriser le chiffrement à tous les utilisateurs satisfaisant une certaine politique de sécurité. Plus généralement, une clé de déchiffrement permettra à un utilisateur d'apprendre une certaine *fonction* de la donnée chiffrée. On peut par exemple imaginer un filtre à spams capable de déterminer si un chiffré correspond à un clair qui est ou non un spam, et aucune autre information sur le clair. Nous nous restreindrons ici à une classe moins générale mais déjà très puissante, la classe de chiffrement par prédicats avec indices publics [17].

Définition 9 (Chiffrement par prédicats avec indices publics) *Soient λ un entier, le paramètre de sécurité, et $R : \Sigma_k \times \Sigma_e \to \{0,1\}$ une fonction booléenne, avec Σ_k et Σ_e représentant les espaces d'indices de clé et d'indices de chiffré. Un* protocole de chiffrement par prédicats *Π pour une relation R est la donnée de quatre algorithmes probabilistes polynomiaux :*

$\mathsf{Setup}(1^\lambda, \mathsf{des}) \to (\mathsf{mpk}, \mathsf{msk})$ **:** *Cet algorithme prend en entrée λ et une description générale* des[6]*. Il renvoie une clé publique maîtresse* mpk *et une clé secrète maîtresse* msk*.*

$\mathsf{Extract}(1^\lambda, \mathsf{mpk}, \mathsf{msk}, X) \to \mathsf{SK}_X$ **:** *L'algorithme d'extraction de clé (secrète) prend en entrée le paramètre de sécurité λ, les clés maîtresses* mpk *et* msk*, et un indice de clé $X \in \Sigma_k$. Il renvoie une clé secrète sk_X associée à X.*

$\mathsf{Enc}(1^\lambda, \mathsf{mpk}, M, Y) \to C$ **:** *L'algorithme de chiffrement prend en entrée le paramètre de sécurité λ, la clé publique maîtresse* mpk*, un message $M \in \{0,1\}^*$ et un indice de chiffré $Y \in \Sigma_e$. Il renvoie un chiffré C.*

$\mathsf{Dec}(1^\lambda, \mathsf{mpk}, \mathsf{SK}_X, X, C, Y) \to \{M, \perp\}$ **:** *L'algorithme de déchiffrement prend en entrée le paramètre de sécurité λ, la clé publique maîtresse* mpk*, une clé secrète sk_X pour l'indice X et un chiffré C pour l'indice Y. Il renvoie soit un message M, soit un symbole $\perp$ qui indique que le chiffré n'est pas valide.*

Le protocole de chiffrement par prédicats doit être correct*, ce qui signifie que pour tout λ suffisamment grand, pour tous $(X,Y) \in \Sigma_k \times \Sigma_e$ tels que $R(X,Y) = 1$ et tout message $M \in \{0,1\}^*$, si $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{des})$, alors*

$$\mathsf{Dec}(1^\lambda, \mathsf{mpk}, \mathsf{Extract}(1^\lambda, \mathsf{msk}, X), X, \mathsf{Enc}(1^\lambda, \mathsf{mpk}, M, Y), Y) = M,$$

6. Cette description des consistera par exemple en l'univers des attributs possibles des utilisateurs.

avec probabilité (prise sur tous les aléas internes des algorithmes) négligemment proche de 1*.*

Cette définition généralise le chiffrement fondé sur l'identité. Pour le voir, il suffit de prendre le prédicat R consistant à tester l'égalité de deux chaînes de caractère. En effet, on a alors $\Sigma_e = \Sigma_k = \{0,1\}^*$ et $R(X,Y) = 1$ si $X = Y$, et 0 sinon. Elle englobe également le chiffrement par attributs, le chiffrement fonctionnel pour les langages réguliers présent dans [71], le chiffrement par produit scalaire [39] et le chiffrement par attributs pour des circuits généraux présentés dans [32] par exemple.

Sécurité du chiffrement par prédicats avec indices publics. Donnons maintenant la définition standard de la sécurité d'un chiffrement par prédicats. Les constructions satisfaisant cette propriété sont souvent dites *payload hiding*, mais on peut conserver le terme d'indistinguabilité des chiffrés. Nous décrivons la version *adaptative* du scénario d'attaque, dans lequel l'attaquant choisit l'indice du chiffré challenge $Y^\star$ en même temps que les messages M_0 et M_1. Le schéma que nous décrirons dans la section 6.1 ne sera sûr que dans un scénario *sélectif*, plus faible, dans lequel l'attaquant s'engage sur cet indice du chiffré challenge *avant* le lancement du setup par le challenger.

Définition 10 *Soit* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *un protocole de chiffrement par prédicats pour une relation* R*. Soient* λ *le paramètre de sécurité. Soit* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *un adversaire fonctionnant en deux étapes ayant accès à un oracle d'extraction* $\mathcal{O}_{\mathsf{Extract}}(\cdot)$*. Considérons le jeu suivant :*

$\boxed{\textit{Expérience } \mathbf{Exp}_{\Pi}^{\mathsf{ph-cpa}}(\mathcal{A}, \mathsf{des})}$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \mathsf{des})$
$(M_0, M_1, Y^\star, st) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Extract}}(\cdot)}(1^\lambda, \mathsf{mpk})$
$b^\star \hookleftarrow U(\{0,1\})$
$C^\star \leftarrow \Pi.\mathsf{Enc}(1^\lambda, \mathsf{mpk}, M_{b^\star}, Y^\star)$
$b \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{Extract}}(\cdot)}(1^\lambda, C^\star, st)$
Renvoie 1 *si* $b = b^\star$ *et* 0 *sinon.*

Durant sa première phrase, l'attaquant $\mathcal{A}_1$ *interroge l'oracle* $\mathcal{O}_{\mathsf{Extract}}$ *sur un indice* X *de son choix pour obtenir une clé secrète* $\mathsf{sk}_X = \mathsf{Extract}(\mathsf{msk}, X)$*, et ce, de façon adaptative. L'indice du chiffré challenge* $Y^\star$ *est tel que* $R(X, Y^\star) = 0$ *pour tout indice de clé* X *ayant été soumis à l'oracle. Dans la seconde phase, l'attaquant* $\mathcal{A}_2$ *peut poursuivre ses requêtes avec comme même restriction naturelle* $R(X, Y^\star) = 0$*. L'avantage de l'attaquant est défini par :*

$$\mathsf{Adv}_{\Pi}^{\mathsf{ph-cpa}}(\mathcal{A}) = |\Pr(\mathbf{Exp}_{\Pi}^{\mathsf{ph-cpa}}(\mathcal{A}) = 1) - 1/2|.$$

Le schéma Π est considéré sûr du point de vue de l'indistinguabilité des chiffrés dans une attaque à clairs choisis (ou est payload-hiding*) si tout attaquant probabiliste polynomial tel que décrit ci-dessus a un avantage négligeable.*

Le protocole que nous allons présenter ci-dessous est un chiffrement par attributs (ABE). Ce dernier se décline sous deux formes différentes : le chiffrement par attributs avec politique d'accès dans les chiffrés (CP-ABE) et celui avec politique d'accès dans les clés *key-policy* (KP-ABE). Dans les chiffrements par attributs de type CP-ABE, l'ensemble des attributs sert à caractériser les clés secrètes des utilisateurs, alors que la politique d'accès, désignant les destinataires pouvant déchiffrer, est spécifiée par l'expéditeur dans le chiffré. De façon duale, dans les chiffrement par attributs de type KP-ABE, les attributs accompagnent les chiffrés, et aux clés secrètes sont associées les politiques d'accès qui spécifient les chiffrés qu'un utilisateur a le droit de déchiffrer.

Le chiffrement par attributs que nous allons décrire est de type KP-ABE. En voici une définition formelle. Si le protocole autorise n'importe quel circuit comme politique d'accès, alors il est possible de transformer le chiffrement par attributs de type KP-ABE en un chiffrement par attributs de type CP-ABE en utilisant un circuit universel (c'est-à-dire un circuit qui prend en entrée un circuit et une entrée valide pour ce dernier, et qui évalue sa première entrée sur sa seconde entrée).

Définition 11 (KP-ABE) *Soit U l'espace des attributs et soit ℓ le nombre d'attributs par chiffré. Un protocole de chiffrement par attributs de type KP-ABE pour une collection $\mathcal{AS}$ de structures d'accès sur U est un chiffrement par prédicats pour $R^{\mathsf{KP}} : \mathcal{AS} \times U^\ell \to \{0,1\}$ définie par $R^{\mathsf{KP}}(\mathbb{A}, \omega) = 1$ ssi $\omega \in \mathbb{A}$ (pour $\omega \in U^\ell$ et $\mathbb{A} \in \mathcal{AS}$). De plus, la description* des *consiste en l'univers des attributs U et l'entier ℓ, et $\Sigma_k^{\mathsf{KP}} = \mathcal{AS}$ and $\Sigma_e^{\mathsf{KP}} = U^\ell$.*

Une autre propriété de sécurité est parfois désirable en pratique : il s'agit de la notion de *masquage des attributs* [39] (*attribute-hiding* en anglais) qui garantit que les chiffrés ne dévoilent aucune information sur les attributs sous-jacents. Le schéma que nous présentons plus bas ne satisfait pas cette propriété de sécurité.

6.1 Chiffrement par attributs reposant sur LWE

Comme nous l'avons vu, tout chiffrement fondé sur l'identité peut être vu comme un chiffrement par attributs, restreint à une classe très simple de politiques d'accès : un utilisateur peut déchiffrer exactement les messages chiffrés pour l'attribut égal à son identité, c'est-à-dire, si $\mathsf{att} = \mathsf{id}$. Plusieurs travaux récents ont proposé des schémas de chiffrement par attributs dont la sécurité repose sur la difficulté présumée de LWE, permettant des politiques d'accès plus riches. Dans le schéma de [4], un utilisateur peut déchiffrer un message chiffré si et seulement si $\mathsf{id} \approx \mathsf{att}$ (pour la distance de Hamming). Dans le schéma de [5], les attributs et identités sont des vecteurs de bits de longueur fixée, et un utilisateur peut déchiffrer exactement quand le produit scalaire $\langle \mathsf{att}, \mathsf{id} \rangle$ vaut 1. Dans le schéma de [18],

l'espace d'attributs est $\{0,1\}^\ell$ et la structure d'accès associée à un utilisateur est une formule logique booléenne f sans négations avec ℓ variables : l'utilisateur peut déchiffrer si et seulement si $f(\mathsf{att}_1, \ldots, \mathsf{att}_\ell)$. Ici, nous allons décrire un schéma encore plus riche, proposé par Gorbunov, Vaikuntanathan et Wee dans [32], qui à chaque utilisateur associe un circuit booléen $\mathcal{C}$: un utilisateur peut déchiffrer si et seulement si son circuit $\mathcal{C}$ s'évalue en 1 sur l'entrée $(\mathsf{att}_1, \ldots, \mathsf{att}_\ell)$. Une construction alternative aussi riche a été proposée également dans [28], mais la sécurité de ce dernier repose sur des hypothèses non standard. Enfin, plus récemment encore, un autre protocole s'inspirant de celui de [32] mais plus efficace a été proposé [16].

TOR. Le schéma de [32] repose sur un outil intermédiaire, appelée recodage de 2 vers 1 (TOR pour *Two to One Recoding*). Supposons que l'on dispose de trois clés publiques $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_{\mathsf{tgt}}$ pour le schéma de chiffrement dual-Regev, indépendamment échantillonnées selon $U(\mathbb{Z}_q^{m\times n})$. L'objectif du TOR est de trouver deux matrices $\mathbf{R}_0$ et $\mathbf{R}_1$ dans $\mathbb{Z}^{m\times m}$ à petits coefficients telles que $\mathbf{R}_0\mathbf{A}_0 + \mathbf{R}_1\mathbf{A}_1 = \mathbf{A}_{\mathsf{tgt}}$. Ces deux matrices sont appelées *matrices de rechiffrement* de $\mathbf{A}_0$ et $\mathbf{A}_1$ vers $\mathbf{A}_{\mathsf{tgt}}$. Si l'on dispose de deux vecteurs $\mathbf{A}_0\mathbf{s}+\mathbf{e}_0$ et $\mathbf{A}_1\mathbf{s}+\mathbf{e}_1$ de $\mathbb{Z}_q^m$, qui encodent le même secret $\mathbf{s}$, alors on peut obtenir :

$$\mathbf{R}_0 \cdot (\mathbf{A}_0\mathbf{s}+\mathbf{e}_0) + \mathbf{R}_1 \cdot (\mathbf{A}_1\mathbf{s}+\mathbf{e}_1) = \mathbf{A}_{\mathsf{tgt}}\mathbf{s} + \mathbf{e}_{\mathsf{tgt}}, \text{ avec } \mathbf{e}_{\mathsf{tgt}} = \mathbf{R}_0\mathbf{e}_0 + \mathbf{R}_1\mathbf{e}_1,$$

c'est-à-dire un encodage de $\mathbf{s}$ sous $\mathbf{A}_{\mathsf{tgt}}$, avec une erreur sous-jacente $\mathbf{e}_{\mathsf{tgt}}$. Ce dernier vecteur est court, car $\mathbf{R}_0, \mathbf{R}_1, \mathbf{e}_0$ et $\mathbf{e}_1$ ont tous de petites entrées.

Il y a plusieurs façons de trouver ces matrices de rechiffrement $\mathbf{R}_0$ et $\mathbf{R}_1$:

- *Avec une trappe pour* $\mathbf{A}_0$. Supposons que l'on connaisse $\mathbf{S}_0 \in \mathbb{Z}^{m\times m}$, une petite base de $\Lambda_q^\perp(\mathbf{A}_0)$. Alors on peut échantillonner chaque ligne de $\mathbf{R}_1$ selon $D_{\mathbb{Z}^m,\sigma}$, et utiliser l'algorithme du Lemme 2 comme expliqué à la fin de la Section 5.2, une fois pour chaque ligne, pour trouver $\mathbf{R}_0$ gaussien tel que $\mathbf{R}_0\mathbf{A}_0 = \mathbf{A}_{\mathsf{tgt}} - \mathbf{R}_1\mathbf{A}_1 \bmod q$.
- *Avec une trappe pour* $\mathbf{A}_1$. L'algorithme est le symétrique du précédent.
- *Sans trappe, en simulant.* On peut échantillonner chaque ligne de $\mathbf{R}_0$ et $\mathbf{R}_1$ selon $D_{\mathbb{Z}^m,s}$, et définir $\mathbf{A}_{\mathsf{tgt}} = \mathbf{R}_0\mathbf{A}_0 + \mathbf{R}_1\mathbf{A}_1$.

Une variante du *leftover hash lemma* (Lemme 3) pour des distributions gaussiennes discrètes permet de montrer que les distributions obtenues pour $(\mathbf{A}_0, \mathbf{A}_1, \mathbf{R}_0, \mathbf{R}_1, \mathbf{A}_{\mathsf{tgt}})$ sont à distances statistiques négligeables les unes des autres.

Le protocole de Gorbunov, Vaikuntanathan et Wee. Comme les clés de déchiffrement sont associées à des circuits, il convient de préciser comment les circuits sont représentés. Un circuit $\mathcal{C}$ est constitué de portes logiques $g : \{0,1\}^2 \mapsto \{0,1\}$ et d'arêtes, qui prennent des valeurs dans $\{0,1\}$. On suppose qu'il y a exactement ℓ arêtes d'entrées $w_1, \ldots, w_\ell$ et une arête de sortie $w_{|\mathcal{C}|}$ (le nombre d'arêtes est noté $|\mathcal{C}|$). Chaque arête w_i qui n'est pas une arête d'entrée (c'est-à-dire $\ell < i \leq |\mathcal{C}|$) est une arête de sortie d'une porte logique g_i, qui a deux arêtes d'entrée $w_{\mathsf{inp}_0(i)}$ et $w_{\mathsf{inp}_1(i)}$, avec $\mathsf{inp}_0(i) \leq \mathsf{inp}_1(i) < i$. Si les valeurs portées

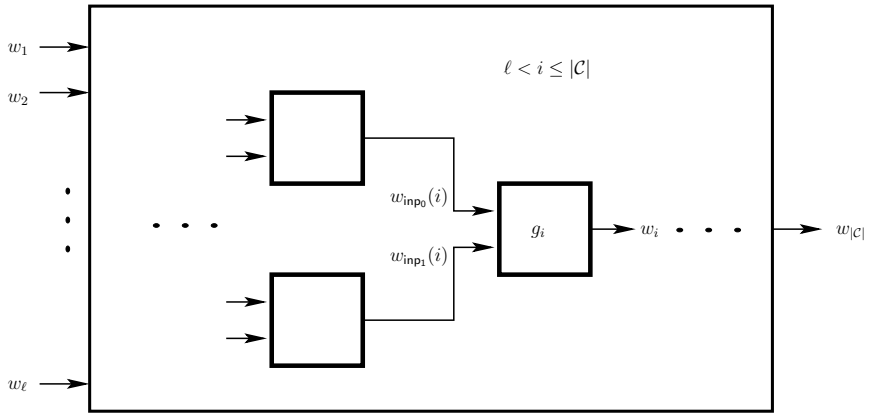

Figure 9. Description d'un circuit $\mathcal{C}$.

par $w_{\mathsf{inp}_0(i)}$ et $w_{\mathsf{inp}_1(i)}$ sont b_0 et b_1, alors la valeur portée par w_i est $g_i(b_0, b_1)$. La Figure 9 illustre ces notations.

Dans le protocole de Gorbunov, Vaikuntanathan et Wee, les attributs att_i sont à valeurs binaires, et correspondent aux ℓ arêtes d'entrée. Pour chacun des attributs, et chaque valeur binaire que peut prendre cet attribut, on génère deux paires de clés $(\mathbf{A}_i^{(0)}, \mathbf{S}_i^{(0)}), (\mathbf{A}_i^{(1)}, \mathbf{S}_i^{(1)}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}^{m\times m}$, en utilisant l'algorithme du Lemme 4. La donnée publique mpk est constituée des $\mathbf{A}_i^{(b)}$ (pour $i \leq \ell$ et $b \in \{0,1\}$) et d'une matrice $\mathbf{A}_{\mathsf{out}} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ associée à l'arête de sortie. La clé secrète maîtresse est $\mathsf{msk} = (\mathbf{S}_i^{(b)})_{i\leq\ell, b\in\{0,1\}}$.

Pour chiffrer un message $\mathbf{m} \in \{0,1\}^m$ pour les attributs $(\mathsf{att}_1, \ldots, \mathsf{att}_\ell)$, l'expéditeur échantillonne $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$ et $\ell + 1$ vecteurs $\mathbf{e}_1, \ldots, \mathbf{e}_\ell, \mathbf{e}_{\mathsf{out}} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$, et envoie $\mathbf{c}_i = \mathbf{A}_i^{\mathsf{att}_i}\mathbf{s} + \mathbf{e}_i$ pour chaque $i \leq \ell$, ainsi que $\mathbf{c}_{\mathsf{out}} = \mathbf{A}_{\mathsf{out}}\mathbf{s} + \mathbf{e}_{\mathsf{out}} + \lfloor q/2 \rfloor \cdot \mathbf{m}$.

Le récepteur possède un circuit $\mathcal{C}$, et une clé de déchiffrement associée, fournie par l'autorité. Celle-ci la crée à l'aide de la clé secrète maîtresse msk. L'intuition est qu'à chaque porte va être associée un TOR, de telle sorte que si le circuit s'évalue en 1 quand on lui donne les attributs att_i du chiffré en entrée (pour $i \leq \ell$), alors le récepteur pourra combiner les $\mathbf{A}_i^{(b)}$ pour obtenir $\mathbf{A}_{\mathsf{out}}$, en appliquant le TOR tout au long du circuit. Ainsi, le récepteur pourra combiner les $\mathbf{c}_i$ de telle sorte à pouvoir ôter le terme $\mathbf{A}_{\mathsf{out}}\mathbf{s}$ de $\mathbf{c}_{\mathsf{out}}$, et ainsi retrouver $\mathbf{m}$.

La clé de déchiffrement associée à $\mathcal{C}$ est créée de la manière suivante. Pour chaque arête w_i qui n'est pas une arête d'entrée ni l'arête de sortie, l'autorité crée deux paires, une pour chaque valeur que peut porter l'arête, de matrices $(\mathbf{A}_i^{(0)}, \mathbf{S}_i^{(0)}), (\mathbf{A}_i^{(1)}, \mathbf{S}_i^{(1)}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}^{m\times m}$, avec l'algorithme du Lemme 4, pour $\ell < i < |\mathcal{C}|$. L'autorité fixe $\mathbf{A}_{|\mathcal{C}|}^{(1)} = \mathbf{A}_{\mathsf{out}}$ et échantillonne $\mathbf{A}_{|\mathcal{C}|}^{(0)} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$. Soient g_i la porte logique dont est issue w_i, et $w_{\mathsf{inp}_0(i)}$ et $w_{\mathsf{inp}_1(i)}$ ses deux

arêtes d'entrée. L'autorité crée quatre matrices de rechiffrement, une pour chaque paire de valeurs d'entrées $(b_0, b_1) \in \{0,1\}^2$ possible de la porte g_i. La matrice de rechiffrement $\mathbf{R}_i^{(b_0,b_1)} \in \mathbb{Z}^{m \times 2m}$ satisfait

$$\mathbf{R}_i^{(b_0,b_1)} \cdot \begin{bmatrix} \mathbf{A}_{\mathsf{inp}_0(i)}^{(b_0)} \\ \hline \mathbf{A}_{\mathsf{inp}_1(i)}^{(b_1)} \end{bmatrix} = \mathbf{A}_i^{(g_i(b_0,b_1))}.$$

Si les bits portés par les arêtes d'entrée sont b_0 et b_1, alors le récepteur saura recoder $\mathbf{A}_{\mathsf{inp}_0(i)}^{(b_0)}$ et $\mathbf{A}_{\mathsf{inp}_1(i)}^{(b_1)}$ en $\mathbf{A}_i^{(g_i(b_0,b_1))}$, qui est la matrice de l'arête w_i pour la valeur $g_i(b_0, b_1)$. La clé de déchiffrement associée au circuit est alors $\mathsf{sk}_{\mathcal{C}} = (\mathbf{R}_i^{(b_0,b_1)})_{b_0,b_1,i}$ pour $b_0, b_1 \in \{0,1\}$ et $\ell < i \leq |\mathcal{C}|$.

Afin de déchiffrer le chiffré $(\mathbf{c}_1, \ldots, \mathbf{c}_\ell, \mathbf{c}_{\mathsf{out}})$ pour les attributs $(\mathsf{att}_1, \ldots, \mathsf{att}_\ell)$, le récepteur évalue le circuit $\mathcal{C}$ sur les att_i et les $\mathbf{c}_i$ en parallèle : pour une porte g_i avec des arêtes d'entrées portant les valeurs b_0 et b_1 et avec les vecteurs associés $\mathbf{c}_{\mathsf{inp}_0(i)}$ et $\mathbf{c}_{\mathsf{inp}_1(i)}$, le récepteur calcule $g_i(b_0, b_1)$ et $\mathbf{c}_i = \mathbf{R}_i^{(b_0,b_1)} \cdot [\mathbf{c}_{\mathsf{inp}_0(i)}^T, \mathbf{c}_{\mathsf{inp}_1(i)}^T]^T$. Le vecteur $\mathbf{c}_i$ calculé est de la forme $\mathbf{A}_i^{(g_i(b_0,b_1))}\mathbf{s} + \mathbf{e}_i$, pour un vecteur $\mathbf{e}_i$ tel que $\|\mathbf{e}_i\| \leq n^{O(1)} \cdot \max(\|\mathbf{e}_{\mathsf{inp}_0(i)}\|, \|\mathbf{e}_{\mathsf{inp}_1(i)}\|)$. Si $\mathcal{C}$ s'évalue en 1 sur l'entrée $(\mathsf{att}_1, \ldots, \mathsf{att}_\ell)$, alors, à la fin de l'évaluation du circuit, le récepteur obtient un vecteur de la forme $\mathbf{A}_{|\mathcal{C}|}^{(1)}\mathbf{s} + \mathbf{e}_{|\mathcal{C}|}$, qu'il peut soustraire à $\mathbf{c}_{\mathsf{out}}$ et ainsi obtenir $\mathbf{e}_{\mathsf{out}} - \mathbf{e}_{|\mathcal{C}|} + \lfloor q/2 \rfloor \cdot \mathbf{m}$. Si le paramètre de bruit α a été fixé suffisamment petit (de l'ordre de $n^{-\Omega(d)}$ avec d est la profondeur du circuit $\mathcal{C}$), alors ce vecteur $\mathbf{e}_{\mathsf{out}} - \mathbf{e}_{|\mathcal{C}|} + \lfloor q/2 \rfloor \cdot \mathbf{m}$ est petit par rapport à q, et le récepteur peut alors retrouver $\mathbf{m}$.

Sécurité sous LWE. Le schéma de Gorbunov, Vaikuntanathan et Wee peut être prouvé sûr contre des attaques à clairs choisis, pour une sécurité sélective. Ainsi, l'attaquant annonce à l'avance les attributs $\mathsf{att}_1^\star, \ldots, \mathsf{att}_\ell^\star$ pour lesquels il va plus tard obtenir un challenge ; il peut ensuite demander à voir des clés de déchiffrements pour un nombre polynomial de circuits $\mathcal{C}$ tels que $\mathcal{C}(\mathsf{att}_1^\star, \ldots, \mathsf{att}_\ell^\star) = 0$; enfin, il choisit un des deux clairs $\mathbf{m}_0$ et $\mathbf{m}_1$ et obtient le chiffré de l'un des deux pour les attributs $\mathsf{att}_1^\star, \ldots, \mathsf{att}_\ell^\star$, et il doit deviner si le clair sous-jacent est $\mathbf{m}_0$ ou $\mathbf{m}_1$. La principale difficulté technique, pour le challenger, est de répondre correctement aux requêtes de clés de déchiffrement de l'attaquant : en particulier, les distributions des $\mathsf{sk}_{\mathcal{C}}$ renvoyées doivent êtres indistinguables des distributions des $\mathsf{sk}_{\mathcal{C}}$ du vrai schéma.

Le challenger fournit à l'attaquant des matrices $\mathbf{A}_i^{(b)}$ (pour $i \leq \ell$) et une matrice $\mathbf{A}_{\mathsf{out}} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$. Il s'agit de la donnée publique mpk. Les matrices $\mathbf{A}_i^{(\mathsf{att}_i^\star)}$ et $\mathbf{A}_{\mathsf{out}}$ proviennent d'une instance du problème LWE (et le challenger ne connaît pas de $\mathbf{S}_i^{(\mathsf{att}_i^\star)}$ associée). Les matrices $\mathbf{A}_i^{(1-\mathsf{att}_i^\star)}$ sont construites avec des $\mathbf{S}_i^{(1-\mathsf{att}_i^\star)}$, comme dans le protocole.

Supposons que l'attaquant demande au challenger une clé secrète pour un circuit $\mathcal{C}$ qui s'évalue en 0 sur l'entrée $\mathsf{att}^\star$. Le challenger va procéder de telle sorte que pour toute arête w_i il connaisse une matrice $\mathbf{S}_i^{(1-b^\star)}$ mais pas de matrice $\mathbf{S}_i^{(b^\star)}$, avec $b^\star \in \{0,1\}$ la

valeur portée par w_i si on donne $\mathsf{att}^\star$ en entrée à $\mathcal{C}$. Fixons une arête interne w_i, sortant de la porte g_i. Alors le challenger crée définit $\mathbf{A}_i^{(1-b^\star)}, \mathbf{S}_i^{(1-b^\star)}$ comme dans le protocole réel. Pour chaque paire de valeurs d'entrées (b_0, b_1) de g_i distincte de la paire de valeurs $(b_0^\star, b_1^\star)$ prises pour l'entrée $\mathsf{att}^\star$, on crée la matrice de rechiffrement comme dans le protocole réel (on connaît alors soit $\mathbf{S}_{\mathsf{inp}_0(i)}^{(1-b_0^\star)}$ soit $\mathbf{S}_{\mathsf{inp}_1(i)}^{(1-b_1^\star)}$). Pour $(b_0, b_1) = (b_0^\star, b_1^\star)$, on utilise le troisième algorithme du TOR : on échantillonne la matrice de rechiffrement d'abord, et on l'utilise pour créer $\mathbf{A}_i^{(b^\star)}$. Enfin, la clé $\mathsf{sk}_\mathcal{C}$ est l'ensemble des clés de rechiffrement créées. Grâce aux propriétés statistiques du TOR, sa distribution est statistiquement indistinguable de la distribution de $\mathsf{sk}_\mathcal{C}$ dans le vrai protocole.

7 Conclusion

Le problème LWE est un formidable outil pour concevoir des systèmes cryptographiques évolués, répondant aux besoins de sécurité liés aux nouveaux usages. Il a suffi à la mise en œuvre de systèmes sophistiqués, pour certains (comme le chiffrement par attributs pour des circuits arbitraires) jamais réalisés auparavant. D'autres systèmes nécessitaient des couplages définis sur les courbes algébriques, et faisaient souvent intervenir des problèmes nouveaux *ad hoc* pour démontrer la sécurité. Le problème LWE suffit à lui seul pour assurer la sécurité de ces mêmes systèmes.

La principale difficulté dans la conception des cryptosystèmes réside dans l'équilibre à atteindre dans le triptyque sécurité-efficacité-fonctionnalité. Les chiffrements fondés sur l'identité et par attributs que nous avons étudiés précédemment témoignent de cette difficulté. Les systèmes sont expressifs, mais leur sécurité pourrait être renforcée. Obtenir de tels schémas avec une sécurité adaptative (sans grosse perte dans la preuve de sécurité) reste un problème ouvert difficile. Un autre problème ouvert, lié à la fonctionnalité, est celui de trouver des variantes hiérarchiques efficaces. Les solutions existantes [22,2,3,16] nécessitent des paramètres d'erreur α qui décroissent très vite avec le nombre de niveaux de hiérarchie, obligeant à augmenter les paramètres pour préserver le niveau de sécurité.

Un problème proche de LWE que nous n'avons pas abordé est le problème SIS (pour *Short Integer Solution*). Il est défini de la manière suivante : soient n et q des paramètres entiers (avec habituellement q polynomial en n), et $\beta > 0$; étant donnée une matrice $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{n\times m}$, le problème SIS consiste à trouver un vecteur $\mathbf{z} \in \mathbb{Z}^m$ tel que $\mathbf{Az} = \mathbf{0} \bmod q$ et $0 < \|\mathbf{z}\| \leq \beta$. Ce problème s'avère utile pour construire des fonctions de hachage résistant aux collisions [6] ainsi que des signatures numériques [30,45,43]. Comme pour LWE, des problèmes pire cas portant sur les réseaux euclidiens se réduisent à SIS [6,53,30].

Pour gagner en efficacité, des variantes de LWE et SIS faisant intervenir des matrices structurées ont été introduites [50,69,46]. Cet ajout de structure permet d'effectuer les multiplications matrices-vecteurs en temps quasi-linéaire, plutôt que quadratique. Du point de vue algébrique, si les structures de matrices sont bien choisies, on se retrouve avec des opérations sur des anneaux de polynômes du type $\mathbb{Z}_q[x]/f(x)$. Les problèmes correspondant,

Ring-SIS et *Ring-LWE*, ont été prouvés au moins aussi difficiles qu'il l'est de trouver une petite base d'un réseau correspondant à un idéal d'un anneau de polynômes $\mathbb{Z}[x]/f(x)$ [44], ou à un idéal de l'anneau d'entiers d'un corps de nombres [60,46]. Certaines constructions cryptographiques à partir de SIS et LWE se transposent facilement à Ring-SIS et Ring-LWE avec le gain d'efficacité attendu, mais pour d'autres cela semble plus difficile.

Remerciements. Ce travail a été financé en partie par le Conseil Européen de la Recherche (ERC Starting Grant ERC-2013-StG-335086-LATTAC). Nous remercions Guilhem Castagnos pour sa lecture attentive de ce chapitre.

Références

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. *J. Cryptology*, 21(3):350–391, 2008.
2. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of Eurocrypt 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010.
3. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Proc. of Crypto 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010.
4. S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional encryption for threshold functions (or fuzzy IBE) from lattices. In *Proc. of PKC*, volume 7293 of *LNCS*, pages 280–297. Springer, 2012.
5. S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *Proc. of Asiacrypt 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, 2011.
6. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of STOC*, pages 99–108. ACM, 1996.
7. M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
8. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
9. N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Proc. of PKC*, volume 6056 of *LNCS*, pages 384–402. Springer, 2010.
10. W. Banaszczyk. New bounds in some transference theorems in the geometry of number. *Math. Ann*, 296:625–635, 1993.
11. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *Proc. of Asiacrypt 2001*, volume 2248 of *LNCS*, pages 566–582. Springer, 2001.
12. B. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput*, 36(5):915–942, 2006.
13. D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Proc. of Eurocrypt*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
14. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Proc. of Eurocrypt 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, 2004.

15. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput*, 32(3):586–615 (electronic), 2003.
16. D. Boneh, V. Nikolaenko, and G. Segev. Attribute-based encryption for arithmetic circuits, 2013. Available at `http://eprint.iacr.org/2013/669`.
17. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Proc. of TCC*, volume 6597 of *LNCS*, pages 253–273. Springer, 2011.
18. X. Boyen. Attribute-based functional encryption on lattices. In *Proc. of TCC*, pages 122–142, 2013.
19. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. In *Proc. of STOC*, pages 575–584. ACM, 2013.
20. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *Proc. of Eurocrypt 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.
21. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Proc. of Eurocrypt 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.
22. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of Eurocrypt 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010. Full version available at `http://www.cc.gatech.edu/~cpeikert/pubs/bonsai.pdf`.
23. B. Chor and R. L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory*, 34:901–909, 1988.
24. C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of IMA Cryptography and Coding 2001*, volume 2260 of *LNCS*, pages 360–363. Springer, 2001.
25. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput*, 33(1):167–226, 2004.
26. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Th.*, 22:644–654, 1976.
27. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. of STOC*. ACM, 1991.
28. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *Proc. of Crypto 2013*, volume 8043 of *LNCS*, pages 479–499. Springer, 2013.
29. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. of STOC*, pages 169–178. ACM, 2009.
30. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of STOC*, pages 197–206. ACM, 2008.
31. O. Goldreich. *Foundations of Cryptography*, volume I – Basic Tools. Cambridge University Press, 2001.
32. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *Proc. of STOC*, pages 545–554. ACM, 2013.
33. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.
34. I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. *Theory of Computing*, 8(1):513–531, 2012.
35. J. Herranz, F. Laguillaumie, and C. Ràfols. Relations between semantic security and anonymity in identity based encryption. *Information Processing Letters*, 111(10):453–460, 2011.
36. J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *Proc. of Eurocrypt 2002*, volume 2332 of *LNCS*, pages 466–481. Springer, 2002.
37. R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstract). In *Proc. of STOC*, pages 12–24. ACM, 1989.

38. J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC Press, 2007.
39. J. Katz, S. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. *J. Cryptology*, 26(2):191–224, 2013.
40. P. N. Klein. Finding the closest lattice vector when it's unusually close. In *Proc. of SODA*, pages 937–941. ACM, 2000.
41. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann*, 261:515–534, 1982.
42. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Proc. of Eurocrypt 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, 2010.
43. V. Lyubashevsky. Lattice signatures without trapdoors. In *Proc. of Eurocrypt 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.
44. V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proc. of ICALP*, volume 4052 of *LNCS*, pages 144–155. Springer, 2006.
45. V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *Proc. of TCC*, volume 4948 of *LNCS*, pages 37–54. Springer, 2008.
46. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Proc. of Eurocrypt*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
47. R. McEliece. A public-key cryptosystem based on algebraic number theory. Technical report, Jet Propulsion Laboratory, 1978. DSN Progress Report 42-44.
48. R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inf. Th.*, 24(5):525–530, 1978.
49. D. Micciancio. *On the hardness of the shortest vector problem*. PhD thesis, Massachussets Institute of Technology, 1998.
50. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Comput. Complexity*, 16(4):365–411, 2007.
51. D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective*. Kluwer Academic Press, 2002.
52. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. of Eurocrypt 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
53. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.
54. D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. *SIAM J. Comput*, 42(3):1364–1391, 2013.
55. P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Proceedings of the 2001 Cryptography and Lattices Conference (CALC'01)*, volume 2146 of *LNCS*, pages 146–180. Springer, 2001.
56. P. Q. Nguyen and B. Vallée (editors). *The LLL Algorithm: Survey and Applications*. Information Security and Cryptography. Springer, 2009.
57. A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Cryptology and Computational Number Theory*, volume 42 of *Proc. of Symposia in Applied Mathematics*, pages 75–88. AMS, 1990.
58. T. Okamoto and T. K. Fully secure functional encryption with general relations from the decisional linear assumption. In *Proc. of Crypto 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, 2010.
59. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009.

60. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proc. of TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.
61. D. Pointcheval. *Advanced Course on Contemporary Cryptology*, chapter Provable Security for Public Key Schemes, pages 133–189. Birkhäuser Publishers, 2005.
62. O. Regev. Lecture notes of *lattices in computer science*, taught at the Computer Science Tel Aviv University. Available at `http://www.cs.tau.il/~odedr`.
63. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC*, pages 84–93. ACM, 2005.
64. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
65. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
66. A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proc. of Eurocrypt 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.
67. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theor. Comput. Science*, 53:201–224, 1987.
68. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.
69. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *Proc. of Asiacrypt*, volume 5912 of *LNCS*, pages 617–635. Springer, 2009.
70. S. Vaudenay. Cryptanalysis of the Chor-Rivest cryptosystem. *J. Cryptology*, 14(2):87–100, 2001.
71. B. Waters. Functional encryption for regular languages. In *Proc. of Crypto 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, 2012.
72. A. Yao. Theory and applications of trapdoor functions. In *Proc. of FOCS*, pages 80–91. IEEE, 1982.