Étude d'un schéma de chiffrement à clé publique

Jérémie Dumas
École Normale Supérieure de Lyon

Décembre 2011

Table des matières

Introduction

Le présent rapport traite d'un article récent de Lyubashevsky, Palacio et Segev [LPS10]. Ce papier présente une nouvelle méthode de chiffrement à clé publique (notée parfois LPS dans ce rapport) et dont la complexité est liée au problème de la somme de sous-ensembles (désigné par *subset sum* ou SS par la suite).

Dans un premier temps nous présenterons le problème de subset sum, avant de mettre en relation le schéma de Segev et al. avec le chiffrement GPV basé sur LWE vu en cours. On présentera ensuite les différentes variantes couvertes par l'article, pour des formes de sécurités plus restrictives ou un protocole de transfert inconscient.

1 Schéma de chiffrement

1.1 Problème de subset sum

Initialement, le problème de la somme de sous-ensembles consiste, étant donné des nombres $a_1, \ldots, a_n$ et une cible T, à trouver un sous-ensemble $\mathbf{s} \in \{0,1\}^n$ tel que $\sum a_i s_i = T$. Le problème d'optimisation est connu pour être NP-difficile dans le cas général.

Cependant nous travaillons ici dans $\mathbb{Z}_M$, sur des entrées randomisées. Plus formellement, le problème de *subset sum* $SS(n, M)$ considéré est le suivant : étant donnés n et $q^m = M$, tirer aléatoirement $\mathbf{a} \in \mathbb{Z}_{q^m}^n$ et $\mathbf{s} \in \{0,1\}^n$, puis renvoyer $\mathbf{a}$ et $T = \mathbf{a} \cdot \mathbf{s}$. La question consiste alors à retrouver $\mathbf{s}$, étant donné $\mathbf{a}$ et $\mathbf{T}$.

Pour des petites valeurs de M le problème peut être résolu efficacement par programmation dynamique. En revanche, lorsque la densité $\frac{n}{\log M}$ diminue, le problème peut encore se résoudre facilement sur des instances aléatoires. En particulier, il existe des transformations vers le problème de plus court vecteur (SVP) dans un réseau. De fait, le statut du problème *subset sum* est un peu particulier : il y a une zone floue quand la densité n'est ni trop grande, ni trop faible, où aucun algorithme de complexité moyenne sous-exponentielle n'est connu actuellement.

1.2 Présentation et contexte

L'idée principale derrière la méthode proposée par Segev et al. [LPS10] est de transformer une instance de subset sum en un problème similaire à LWE, dont le "bruit" ne sera pas vraiment aléatoire, mais contiendra néanmoins des valeurs suffisamment petites pour permettre au schéma de cryptage de fonctionner avec forte probabilité.

Notations Soient $\mathbf{a} \in \mathbb{Z}_{q^m}^n$ et $\mathbf{s} \in \{0,1\}^n$ deux éléments de subset sum générés aléatoirement. On note $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ la représentation en base q des composants de $\mathbf{a}$, où les a_i sont écrits en colonne. Si l'on écrit $T = \mathbf{a} \cdot \mathbf{s} \in \mathbb{Z}_{q^m}$ en base q sous forme de vecteur colonne $\mathbf{t}$, on peut écrire $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{c}$, où $\mathbf{c} \in \mathbb{Z}_q^m$ est le vecteur de *retenue* analogue au *bruit* dans LWE. On impose que les représentants t_i des coefficients de $\mathbf{t}$ vérifient $|t_i| \leqslant (q-1)/2$, et on notera enfin $\mathbf{t} = \mathbf{A} \odot \mathbf{s}$ cette opération de subset sum. L'opération duale se définit naturellement par $\mathbf{r}^T \odot \mathbf{A} = (\mathbf{A}^T \odot \mathbf{r})^T$.

Rappels sur GPV Le protocole comprends plusieurs paramètres n, m, q. On génère $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ uniformément. La clé secrète $\mathbf{r} \in \mathcal{E}^m$ est tirée selon une gaussienne bien choisie, et la clé publique vaut $\mathbf{u} = \mathbf{r}^T\mathbf{A}$. Pour chiffrer $z \in \{0,1\}$ on renvoie $(\mathbf{p}, c) = (\mathbf{A}\mathbf{s}+\mathbf{e}, \mathbf{u}\cdot\mathbf{s}+e_0+[\frac{q}{2}]z)$, où $\mathbf{e}, e_0$ sont des termes de bruit "petits" tirés selon une certaine gaussienne, et $\mathbf{s}$ pris uniformément dans $\mathbb{Z}_q^m$. Enfin, le déchiffrement de $(\mathbf{p}, c)$ se fait en calculant $c - \mathbf{r}^T\mathbf{p}$.

Chiffrement LPS Le schéma présenté dans l'article [LPS10] permet d'encrypter en une seule fois un nombre de bits fixé k, mais pour simplifier les notations concentrons-nous sur le cas d'un message $z \in \{0,1\}$. La méthode de chiffrement proposée repose essentiellement sur le fait que si $\mathbf{A'} \in \mathbb{Z}_q^{n \times n}$ et $\mathbf{s} \in \{0,1\}^n$ sont tirés uniformément, alors la distribution $(\mathbf{A'}, \mathbf{A'} \odot \mathbf{s})$ est indistinguable algorithmiquement de l'uniforme (moyennant la difficulté du problème subset sum sous-jacent). Formellement, le schéma de chiffrement proposé se décompose de la manière suivante :

`KeyGen`	Échantillonner $\mathbf{A'} \in \mathbb{Z}_q^{n \times n}$ et $\mathbf{s} \in \{0,1\}^n$ uniformes indépendamment. Calculer $\mathbf{t} = \mathbf{A'} \odot \mathbf{s}$ et poser $\mathbf{A} = [\mathbf{A'} \| \mathbf{t}]$. Renvoyer $(PK, SK) = (\mathbf{A}, \mathbf{s})$.
`Enc`(z)	Échantillonner $\mathbf{r} \in \{0,1\}^n$ de manière uniforme. Renvoyer le chiffré $\mathbf{u}^T = \mathbf{r}^T \odot \mathbf{A} + \frac{q-1}{2}[0^n \| z]$.
`Dec`$(\mathbf{u}^T)$	Décomposer $\mathbf{u}^T = [\mathbf{v}^T \| w]$. Calculer $y = \mathbf{v}^T\mathbf{s} - w \in \mathbb{Z}_q$. Si $\lvert y \rvert < q/4$ renvoyer 0, sinon renvoyer 1.

1.3 Analyse de sécurité et correction

1.3.1 Sécurité

La notion de sécurité considérée est celle de *sécurité sémantique*, ou *sécurité cpa*, tel que vu en cours. Pour montrer la sûreté du système au sens *cpa*, il suffit de voir que pour n'importe quels messages $m_0, m_1 \in \{0,1\}$, les distributions $(PK, \texttt{Enc}_{PK}(m_0))$ et $(PK, \texttt{Enc}_{PK}(m_1))$ sont algorithmiquement indistinguables (l'adversaire aura donc un avantage *cpa* négligeable). Le résultat découle de la difficulté du problème de subset sum, qui rend la distribution $(\mathbf{A}, \mathbf{r}^T \odot \mathbf{A})$ indistinguable de l'uniforme.

1.3.2 Correction

Il s'agit de montrer ici que la fonction de décryptage est bien correcte, c'est-à-dire que $\texttt{Dec}(\mathbf{u}^T) = z$ avec forte probabilité. La preuve repose sur deux lemmes qui montrent que l'opération subset sum $\odot$ est assez similaire à la multiplication de matrices, et donc que $|\mathbf{v}^T\mathbf{s} - w| \approx \frac{q-1}{2}z$.

Remarque. Il est important de noter qu'ici, si $e \in \mathbb{Z}_q$, on désigne par $|e|$ la valeur absolue de son représentant dans $[-\frac{q-1}{2}, \frac{q-1}{2}]$. Les représentants sont donc centrés en 0. De même pour un vecteur $\mathbf{e} \in \mathbb{Z}_q^n$, on notera $\|e\|_\infty = \max_i |e_i|$.

Plus exactement alors, les auteurs de [LPS10] montrent que pour n'importe quel vecteur $\mathbf{r} \in \{0,1\}^n$, avec forte probabilité, les coefficients de $\mathbf{r}^T \odot \mathbf{A} - \mathbf{r}^T\mathbf{A}$ sont petits ($< \sqrt{n}\log n$). De même, pour tout $r, s \in \{0,1\}^n$, avec forte probabilité on a $\|(r^T \odot \mathbf{A})\mathbf{s} - \mathbf{r}^T\mathbf{A}\mathbf{s}\|_\infty < n\log^2 n$.

La fin de la preuve découle ensuite assez aisément en étudiant la différence $\mathbf{v}^T\mathbf{s} - w$, qui peut s'écrire sous la forme $(\mathbf{r}^T \odot \mathbf{A}')\mathbf{s} + \nu s_n - \mathbf{r}^T(\mathbf{A}' \odot \mathbf{s}) - \eta - \frac{q-1}{2}z$, où ν et η sont des termes de retenue assez petits ($< n$). En utilisant les lemmes établis précédemment, on peut donc majorer la première partie de la somme, et montrer que $\frac{q-1}{2}z$ est bien le terme dominant dans l'expression de $\mathbf{v}^T\mathbf{s} - w$. Plus précisément, on trouve que $|\mathbf{v}^T\mathbf{s} - w| < q/4$ si et seulement $z = 0$, avec forte probabilité.

2 Extensions envisagées

2.1 Résistance aux attaques avec fuite d'informations sur la clé

La notion de sécurité face aux attaques avec fuites d'informations sur la clé reprend le schéma de l'attaque *cpa*, mais suppose cette fois que l'adversaire dispose d'informations sur la clé secrète utilisée au cours du jeu. Plus précisément, l'adversaire a la possibilité de soumettre une fonction d'évaluation $f : \mathcal{SK}_n \to \{0,1\}^\lambda$, et de recevoir $f(SK)$, où SK est la clé secrète qui a été générée au début du protocole. La taille de la sortie de f doit être bornée par un paramètre de fuite λ. Dans une attaque *adaptative*, l'adversaire peut choisir la fonction f qu'il soumet après avoir pris connaissance de la clé publique PK qui a été générée, contrairement à une attaque *non-adaptative* où l'adversaire doit soumettre sa fonction f avant toute chose.

Dans leur article [LPS10], les auteurs traitent d'un schéma sémantiquement sûr contre des attaques non-adaptatives, et laissent ouvert la question de prouver la sécurité d'un tel système face à des attaques adaptatives avec fuite d'informations sur la clé.

La méthode de chiffrement proposée est une variante de celle présentée dans la section 1.2. Les paramètres du schéma sont des entiers n, m, q, T. Les primitives de chiffrement et déchiffrement restent identiques, seule change la fonction de génération des clés qui devient :

`KeyGen`	Échantillonner $\mathbf{A}' \in \mathbb{Z}_q^{n\times m}$ et $\mathbf{s} \in \{-\frac{T-1}{2}, \ldots, \frac{T-1}{2}\}^m$ uniformes indépendamment. Calculer $\mathbf{A} = [\mathbf{A}'\|\mathbf{A}'\mathbf{s}]$. Renvoyer $(PK, SK) = (\mathbf{A}, \mathbf{s})$.

La preuve de sécurité repose là encore sur la difficulté supposée du problème de subset sum, et consiste à montrer que pour n'importe quelle fonction $f : \mathcal{SK}_n \to \{0,1\}^n$, les distributions $(PK, \mathtt{Enc}_{PK}(0), f(SK))$ et $(PK, \mathtt{Enc}_{PK}(1), f(SK))$ sont algorithmiquement indistinguables.

2.2 Un protocole de transfert inconscient

Une application intéressante de la méthode de chiffrement proposée par Segev et al. concerne le transfert inconscient (*oblivious transfert* ou OT en anglais). Le principe est

le suivant : un émetteur possède deux messages secrets $\mathbf{z}_0, \mathbf{z}_1$, et un receveur choisit un bit $b \in \{0,1\}$; à la fin du transfert, le receveur doit avoir pris connaissance de $\mathbf{z}_b$, en ignorant tout de $\mathbf{z}_{1-b}$, et l'émetteur ne doit avoir aucune connaissance de b.

Dans la version *honnête* du protocole, on doit garantir la sécurité pour l'un des partis contre un utilisateur honnête-mais-curieux, au sens qu'il suivra le protocole mais cherchera à utiliser les informations qu'il a enregistrées pendant le transfert pour découvrir $\mathbf{z}_{1-b}$ dans un cas, b dans l'autre. Le protocole de transfert inconscient décrit par Segev et al. fonctionne comme suit :

Paramètres	$n, k \in \mathbb{N}$, pour le receveur $b \in \{0,1\}$, pour l'émetteur $\mathbf{z}_0, \mathbf{z}_1$.
Receveur	Génère les clés (PK_b, SK_b), tire PK_{1-b} uniforme dans $\mathbb{Z}_q^{n\times(n+k)}$. Envoie (PK_0, PK_1).
Émetteur	Chiffre $\mathbf{u}_i^T = \mathtt{Enc}_{PK_i}(\mathbf{z}_i)$. Envoie $(\mathbf{u}_0^T, \mathbf{u}_1^T)$
Receveur	Déchiffre $z_b = \mathtt{Dec}_{SK_b}(\mathbf{u}_b^T)$

Le protocole ainsi décrit est sûr pour le receveur, même contre des émetteurs malicieux. En effet, la difficulté du problème subset sum entraîne que la distribution (PK_0, PK_1) est algorithmiquement indistinguable de l'uniforme. De fait, il est difficile pour l'émetteur de déduire b en analysant le couple (PK_0, PK_1).

En revanche, le protocole n'est pas sûr pour l'émetteur, s'il a affaire à un receveur malicieux. En effet, il suffit que le receveur génère deux couples de clés valides (PK_0, SK_0) et (PK_1, SK_1), et il sera en mesure de déchiffrer les deux messages transférés $\mathbf{u}_0^T$ et $\mathbf{u}_1^T$). Toutefois, dans le cadre d'un transfert *honnête*, le protocole ainsi décrit est sûr pour l'émetteur [LPS10].

Conclusion

Dans le présent rapport, nous avons vu un schéma de chiffrement à clé publique récent, dont la sécurité sémantique est basée sur la difficulté du problème subset sum. Ce schéma présente également une variante intéressante qui résiste aux attaques non-adaptatives avec fuite d'informations sur la clé, alors que le cas adaptatif est laissé ouvert.

Néanmoins les auteurs soulignent que la version non-adaptative couvre également des cas d'attaques par canaux auxiliaires réalistes, et citent par exemple celle du redémarrage à froid. Ce type d'attaque repose sur la persistance des informations stockées dans la RAM pendant un court instant après extinction de l'ordinateur. L'attaquant possède un accès physique à la machine chiffrant, il va couper le courant et redémarrer sur un système qu'il contrôle, afin de copier le contenu de la mémoire vive et de l'analyser plus tard. Ce type d'attaque ne dépend donc que du système matériel, et non du procédé de cryptage utilisé.

Un autre aspect peu évoqué au cours de ce rapport concerne le chiffrement simultané de plusieurs bits. Alors qu'il est possible (bien qu'inefficace) de crypter bit à bit chaque message à envoyer, LPS permet également d'encrypter directement un message de k bits. Toutefois, lorsque k est grand (de l'ordre de n^2), le problème subset sum correspondant devient facile, et le système n'est plus sécurisé.

Enfin, concernant le protocole de transfert inconscient, les auteurs soulignent également qu'il existe des techniques à base de boîtes noires ou de preuves sans apport de connaissance qui permettent de transformer un protocole OT *honnête* sûr en un protocole OT général qui soit sécurisé. De fait, il serait possible de sécuriser le protocole présenté ici lorsque l'on fait face à un receveur malicieux.

Références

[LPS10] Vadim Lyubashevsky, Adriana Palacio, and Gil Segev. Public-Key Cryptographic Primitives Provably as Secure as Subset Sum. In Daniele Micciancio, editor, *Theory of Cryptography*, volume 5978 of *Lecture Notes in Computer Science*, pages 382–400. Springer Berlin / Heidelberg, 2010.