Université de Limoges | Faculté des Sciences et Techniques

Atos Worldline

Rapport de Stage de Fin d'Études

Jean-Christophe Deneuville

Master CRYPTIS
Parcours Mathématiques, Cryptologie, Codage et Applications

Université de Limoges

Cryptographie Homomorphe

Responsable :
Stéphane Cauchie

Tuteur universitaire :
Philippe Gaborit

Diffusion Publique

Année Universitaire 2011-2012

Résumé

Ce rapport fait état du travail réalisé dans le cadre du stage de fin d'études du Master CRYPTIS de l'Université de Limoges, que j'ai réalisé au sein de l'équipe *Recherche & Développement* d'*Atos Worldline* à Blois.

Offre de stage
talent to career

À compter de janvier 2012 pour une durée de 5 à 6 mois

Cryptographie homomorphe

Vos missions ... si vous les acceptez

Contexte du stage :
L'équipe R&D assure la veille technologique et développe des prototypes qui mettent en œuvre des technologies innovantes appliquées aux nouveaux usages.

Depuis maintenant quelques années, des schémas cryptographiques dit « fully homomorphic » permettent le calcul de fonctions arbitraires sur les textes chiffrés. Votre tache, sera de dresser un état de l'art des différentes méthodes mettant en place des primitives de chiffrement de ce type.
Dans un deuxième temps, vous dresserez un panel de scenarii d'utilisation possible pour ce genre de crypto-système. Enfin nous en choisirons un pour réaliser un « Proof Of Concept » de cette technologie.

Le projet :

- État de l'art des cryptosystèmes « fully homomorphic »
- Panel des démonstrations possibles
- Réalisation d'un prototype

A la fin du stage, vous aurez appris à :

- Définir et mettre en œuvre un protocole cryptographique
- Faire vos premier pas avec J2EE, technologie client/serveur en java

Opportunité de stage - Campagne de Stage **2011 – 2012**

Atos Worldline est le centre d'expertise d'Atos dans les services transactionnels de haute technologie. Leader dans les transactions électroniques,
Atos Worldline est spécialisé dans les paiements électroniques (émission, acquisition, terminaux, solutions de paiement et traitement de cartes), les eCS (services en ligne pour les clients, les citoyens et les communautés), ainsi que les services pour les marchés financiers.

Ce stage est basé à BLOIS

Si ce stage vous intéresse,
N'hésitez pas, contactez nous :

http://www.atosoriginrecrute.fr

Référence du stage :

AWL-STG-RD-Blois-SCE-CFH

Qui êtes-vous ?

- Bonne base mathématiques
- Lecture d'article scientifique
- Programmation

Atos Worldline

FIGURE 1 – Offre de stage d'Atos Worldline sur le chiffrement homomorphe

Remerciements

Avant de rentrer dans le vif du sujet, j'aimerai remercier en premiers lieux Jean-Claude Barbezange, Stéphane Cauchie, Philippe Gaborit et Carlos Aguilar Melchor.

Merci à Jean-Claude Barbezange pour m'avoir accepté et intégré au sein de l'équipe Research & Development, pour le partage du recul dont il bénéficie au vu de son expérience, ainsi que pour sa disponibilité.

Merci à Stéphane Cauchie pour la patience dont il a su faire preuve, pour l'autonomie qu'il a réussi à m'accorder mais surtout pour ses nombreux conseils qui j'espère me permettront de commencer une thèse dans de bonnes conditions.

D'autre part, je tiens tout particulièrement à remercier mon tuteur de stage futur directeur de thèse Philippe Gaborit, ainsi que mon futur co-directeur de thèse Carlos Aguilar Melchor, pour leur disponibilité, leur soutien moral, leur aide dans les démarches administratives, et la clarté des réponses qu'ils ont su apporter à mes questions.

Par ailleurs, je remercie Juan Manuel Cabrera et Jean-Baptiste Leroy pour leurs conseils apportés lors de discussions de couloir, ou autour d'un café, et qui m'ont parfois fait gagné un temps précieux.

Enfin, je remercie mes parents et mes proches pour les discussions qui m'ont aidé à rendre ce rapport compréhensible, et épuré de ces fautes d'orthographes au possible.

Introduction

Afin de valider ma seconde année de master Cryptis, parcours Mathématiques, Cryptologie, Codage et Applications, j'ai effectué un stage de six mois au sein de l'équipe Recherche et Développement d'Atos Worldline à Blois.

Dans ce rapport, nous commençons par présenter la Société de Service en Ingénierie Informatique Atos Worldline, ainsi qu'un de ses projets LYRICS, dans lequel s'inscrit mon stage sur la cryptographie homomorphe. Un premier objectif du stage sera de déterminer dans quelles mesures il est possible de déléguer des calculs.

Suivant cet objectif, nous établirons un état de l'art des cryptosystèmes simplement homomorphes, et des problèmes sous-jacents. Ces résultats, accompagnés de quelques définitions seront présentés dans les trois sections de la seconde partie.

Nous nous intéresserons ensuite aux cryptosystèmes complètement homomorphes, en rappelant quelques notions sur les réseaux euclidiens. Dans cette troisième parties, nous essaierons de voir en quoi ils consistent, pour la majeure partie d'entre eux à améliorer les précédents.

Enfin, nous discuterons des cas d'usage qu'offrent les cryptosystèmes complètement homomorphes, et exposerons dans la dernière partie de ce rapport les cas d'usage que nous avons retenu, et la preuve de concept réalisée. Nous évoquons également dans cette section ce que nous avons réalisé d'autre au cours du stage, et ce qui a été laissé en suspens, et qui pourra constituer un point de départ pour la thèse.

Table des matières

Table des figures

Première partie
Contexte du stage

1 Présentation de l'entreprise

1.1 ATOS : Atos Origin To Siemens

Avec un chiffre d'affaire de 8,7 milliards d'euros et ses 78 500 salariés (dont 15 000 en France), le groupe Atos est l'un des leaders mondiaux dans les services informatiques.

Atos fournit des services transactionnels de haute technologie (Atos Worldline), ainsi que des solutions de conseil (Atos Consulting), d'intégration de systèmes et d'infogérance.

Le 4 juillet 2011, Atos Origin a racheté la société Siemens IT et a pris le nom d'Atos. Grâce à cela, Atos fait parti du Top 10 mondial des sociétés de services informatiques, et se classe n° 5 mondial en Infogérance et n° 1 en Europe.

1.2 Atos Worldline

1.2.1 Présentation

Atos Worldline, filiale du groupe Atos a été fondée en 2004 par fusion de plusieurs entités du groupe. Atos Worldline est le centre d'expertise d'Atos dans les services transactionnels de haute technologie. Leader dans les transactions électroniques grâce à son chiffre d'affaire de 817 millions d'euros et à ses 5400 employés, Atos Worldline est présent dans 7 pays européens [1] (Belgique, France, Allemagne, Luxembourg, Pays-Bas, Royaume-Uni, et Espagne) mais également en Inde, en Chine, et en Malaisie. Atos Worldline fournit diverses solutions à ses clients dans :

- les paiements électroniques (émission, acquisition, terminaux, solutions de paiement et traitement de cartes),
- les eCS (eServices pour les clients, les citoyens et les communautés),
- les services pour les marchés financiers.

Atos Worldline est présent dans notre vie quotidienne et ceci s'illustre parfaitement avec quelques données numériques [2] :

- **Paiement électronique** :
  - 470 millions de transactions à distance
  - 2.2 milliards d'acquisitions
  - 36 millions de cartes de crédit
  - 850 000 de Terminaux de Paiement Électroniques (TPE)
- **eCS** :
  - 2.1 milliards d'appels (centre d'appels et Serveurs vocaux interactifs)

1. http://www.atosworldline.fr/fr/8/A-propos/Implantations.html
2. http://www.atosworldline.fr/fr/46/A-propos/Chiffres-cles.html

  - 1.6 milliards de SMS
  - 61 millions de boîtes mails
  - 70 millions d'archives à valeur probante
  - 1 milliard de eDocuments
  - 166 millions de pages internet
- **Marchés financiers** :
  - 340 millions de transactions
  - gestion de biens d'une valeur globale de 450 milliards d'euros

1.2.2 Les services proposés par Atos Worldline

Atos Worldline propose de nombreux services :

- Le traitement des données des radars automatiques
- SIPS (Secure Internet Payment Services) est la solution de paiement à distance de la société Atos Worldline (prix Sesame Award en 2007)
- Les passeports biométriques
- Les dossiers médicaux personnels
- Hébergement d'E-mails de FAI
- ...

1.2.3 Organisation d'Atos Worldline

Atos Worldline est composé de sept unités d'affaires (Business Units) ainsi que deux unités transversales appelées Global Platform & Solutions (GPS) et Technical Operation (TO).

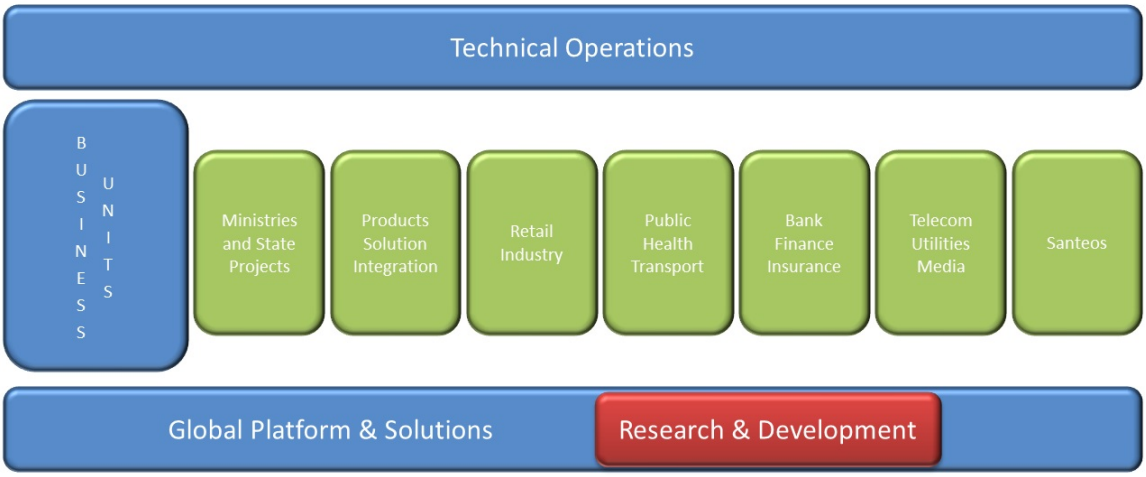


FIGURE 2 – Organisation d'Atos Worldline

GPS est une unité composée de plusieurs équipes dont le département R&D dans lequel j'ai effectué mon stage.

GPS a pour mission de créer des fondations logicielles communes et de construire les services Worldline de demain. Pour cela, GPS crée des solutions génériques à un niveau européen mais contribue également au développement de solutions permettant d'accompagner les autres unités d'Atos Worldline.

Le département R&D est quant à lui chargé d'une mission d'éclaireur dans l'innovation sur les domaines technologiques IT [3] et de ses usages.

La R&D effectue donc une veille technologique constante, et effectue des prototypes utilisant de nouvelles technologies. L'équipe R&D d'Atos Worldline est composée de 17 personnes réparties sur les sites de Tours, Blois, et Seclin.

La R&D effectue également du support auprès des Business Units qui exploitent les prototypes en les présentant aux clients. L'image ci-dessous représente les différentes activités sur lesquelles travaillait l'équipe R&D en 2010.

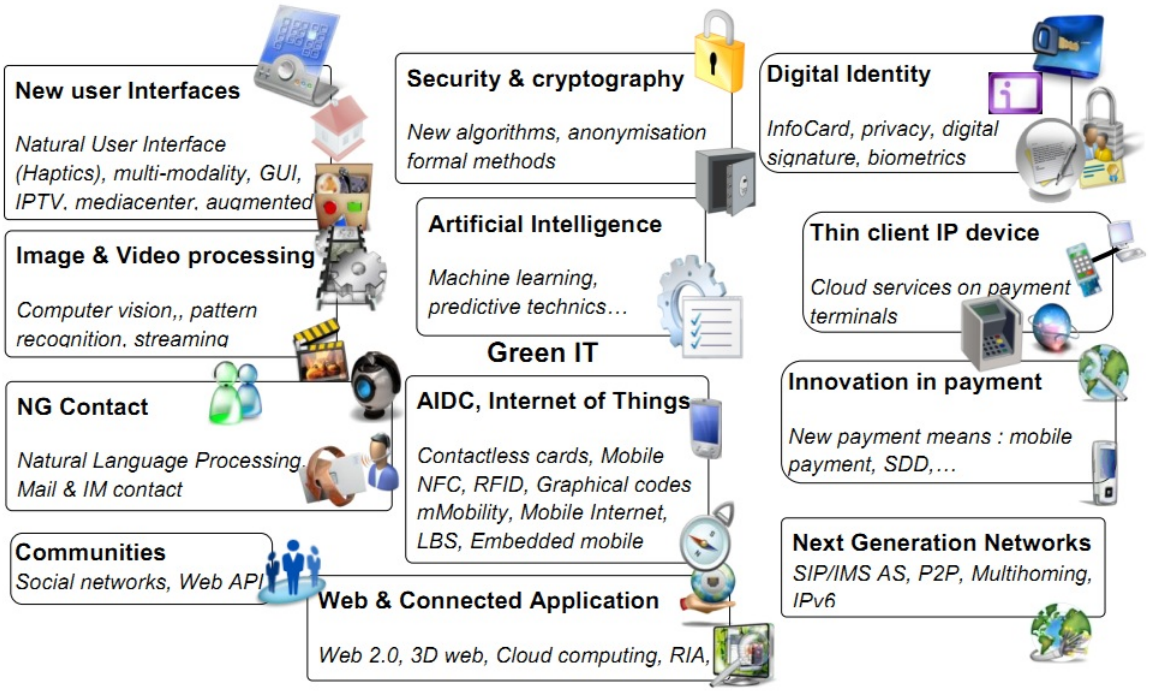


FIGURE 3 – Activités du département R & D

2 Présentation du projet LYRICS

2.1 Signification de l'acronyme

LYRICS [4], *Lightweight privacY-enhancing cRyptography for mobIle Contactless Services*, littéralement "cryptographie pour la protection de la vie privée, optimisée pour les

3. Information Technology
4. Tous les détails à http://projet.lyrics.orange-labs.fr/

services mobiles sans contact", est un projet coopératif financé par l'Agence Nationale de la Recherche [5] dans la catégorie "Ingénierie Numérique et Sécurité" (INS 2011).

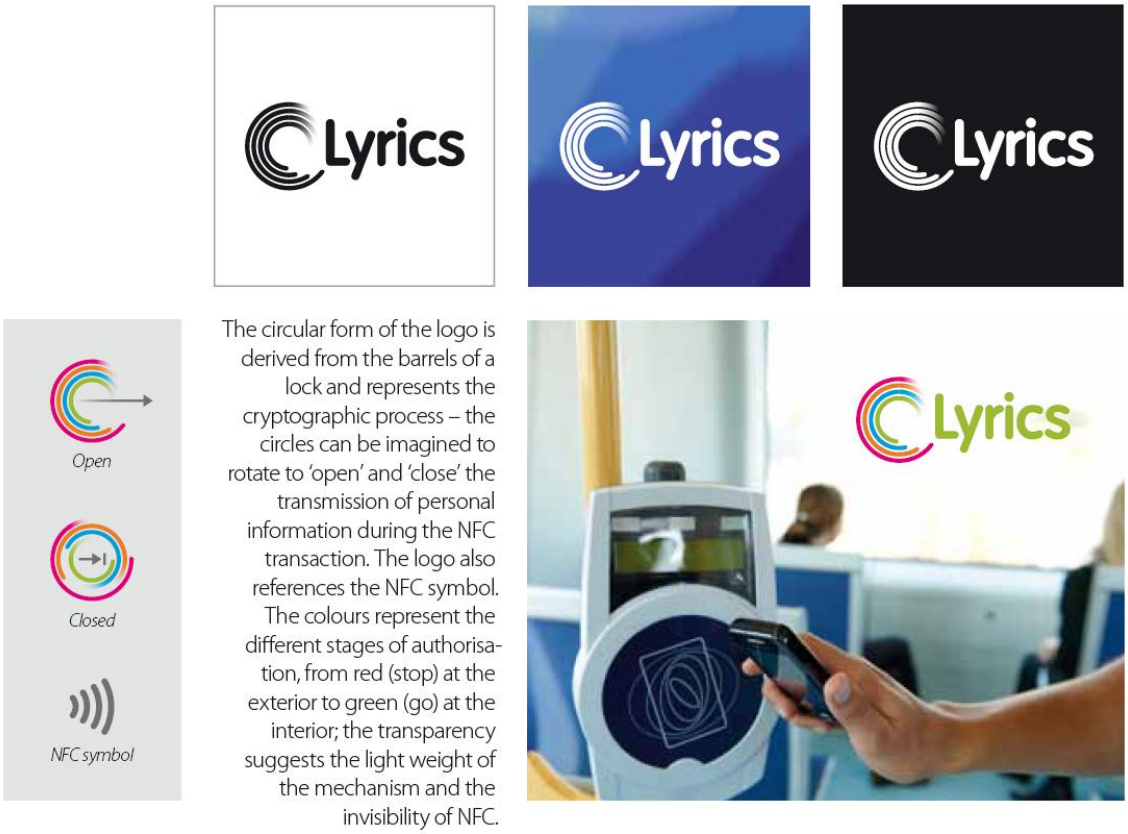


FIGURE 4 – Organisation d'Atos Worldline

2.2 Description

Description du projet par Jacques Traoré de France Télécom [6], coordinateur du projet

Les téléphones portables de prochaine génération intégreront des puces NFC (Near Field Communication). Grace à l'émergence rapide de cette technologie sans contact, les téléphones portables seront bientôt capables de jouer le rôle de tickets électroniques, cartes de crédit, cartes de fidélités, badges d'accès, jetons de vote électronique, porte-monnaie électroniques, etc. La croissance économique attendue du marché mobile NFC est considérable, et des analystes de l'industrie estiment que d'ici à 2014, un utilisateur de mobile sur six détiendra un appareil compatible NFC.

5. Référence ANR du projet LYRICS : ANR- 11-INSE-013
6. jacques.traore@orange-ftgroup.com

Dans ce contexte, protéger la vie privée d'un individu devient une tache particulièrement difficile, surtout lorsque cet individu utilise dans sa vie quotidienne des services sans contact auxquels son identité peut être associée. Les services sans contact peuvent par exemple rendre possibles l'utilisation de titres de transport électroniques, de billets électroniques pour un concert ou de toute autre information personnelle stockée dans le téléphone portable détenu par cet individu. Si une entité non-autorisée est techniquement capable de suivre toutes les traces numériques laissées durant ces interactions, alors cette tierce partie peut aisément construire un profil complet de cet individu, et ainsi occasionner une violation de vie privée. Plus grave encore, cette entité pourrait utiliser librement ces informations à des fins indésirables ou frauduleuses, allant de la publicité (non désirée) ciblée jusqu'à l'usurpation d'identité.

L'objectif de LYRICS est de permettre aux utilisateurs finaux d'accéder en toute sécurité à des services sans contact tout en préservant leur vie privée, c'est à dire sans avoir à révéler leur identité ou toute autre information personnelle non requise. Plus précisément, notre but est de concevoir de nouvelles solutions innovantes permettant d'atteindre les deux principes fondamentaux de la vie privée, à savoir la minimisation des données et la souveraineté des données. Le principe de minimisation des données (ou divulgation minimale) stipule que seules les informations strictement nécessaires pour effectuer une transaction seront divulguées (et rien de plus). En pratique, cela signifie que l'utilisateur ne devrait jamais avoir à donner plus d'informations que nécessaire pour accéder à un service sans contact spécifique. Le principe de souveraineté des données stipule que les informations relatives à un individu lui appartiennent entièrement, et qu'il doit garder un contrôle total sur la façon dont ces données sont utilisées, par qui et avec quel objectif.

Des technologies basées sur la cryptographie répondant partiellement à ces exigences dans certains contextes existent déjà. Cependant aucune d'entre elles n'a été spécifiquement conçue pour les transactions sans contact, où être déconnecté, assurer une très faible latence, ou disposer de ressources limitées sont des enjeux majeurs. LYRICS a pour but de surmonter ces obstacles en fournissant une architecture globale ouverte pour des services sans contact préservant la vie privée et en concevant un ensemble de mécanismes cryptographiques innovants afin de mettre en œuvre et déployer ces services sur des téléphones portables munis de puces NFC. Cet objectif sera réalisé dans le contexte de l'appropriation sociale d'innovations technologiques et de services.

2.3 Objectifs du projet et contexte du stage

Les principaux objectifs du projet LYRICS sont :
- Établir une architecture de haut niveau pour les services protégeant la vie privée
- Créer et spécifier des mécanismes cryptographiques peu coûteux pouvant être utilisés pour protéger la vie privée de l'utilisateur dans le contexte des services mobiles sans contact
- Implémenter de façon sécurisée les outils cryptographiques sur les téléphones NFC choisis
- Développer et expérimenter une implémentation pilote d'un service mobile sans contact préservant la vie privée

Une partie critique du projet réside dans la conception de mécanismes cryptographiques peu coûteux pouvant être assemblés dans le but de supporter une architecture applicative préservant la vie privée de plus haut niveau. Afin d'assurer à la fois des mécanismes sécurisés, efficaces et peu coûteux, plusieurs approches prometteuses sont envisagées :

- **Approche 1** : elle consiste à considérer que la carte SIM est inviolable, c'est-à-dire qu'il est impossible d'extraire les clés embarquées et que toutes les applications embarquées s'exécutent correctement. Ceci mène en général à des solutions très performantes basées sur des mécanismes de chiffrement symétrique. Cependant, dans certains cas d'usage tels que la banque, l'hypothèse d'inviolabilité de la carte SIM peut être inappropriée, typiquement en raison des potentielles pertes économiques en cas de fraude. D'autre part, lorsque l'inviolabilité de la SIM doit être assurée, le coût de fabrication peut s'avérer trop élevé pour l'application cible. Ces problèmes sont pris en compte dans la seconde approche.
- **Approche 2** : accélérer les calculs effectués dans la SIM (partie sécurisée de l'architecture) en délégant une grande partie des calculs au téléphone (partie puissante de l'architecture). Cette approche permet d'accroître l'efficacité tandis que les éléments secrets sont conservés dans la SIM. Cela permet également de profiter de la parallélisation des calculs réalisés à la fois dans la SIM et l'accélérateur cryptographique du téléphone.

  C'est dans cette approche que s'inscrit mon stage sur le chiffrement homomorphe qui permet - en quelques mots - à une tierce personne d'effectuer des calculs sur des données chiffrées, sans fuite d'informations.
- **Approche 3** : LYRICS entreprendra des efforts de recherche importants pour améliorer l'efficacité des systèmes existants, voire inventer de nouveaux mécanismes pour la protection de la vie privée. Des résultats prometteurs concernant la signature de groupe ont été obtenus par NEC.
- **Approche 4** : Utiliser un accélérateur cryptographique spécialement conçu par NEC pour la signature de groupe afin d'obtenir de meilleures performances pour le mobile.

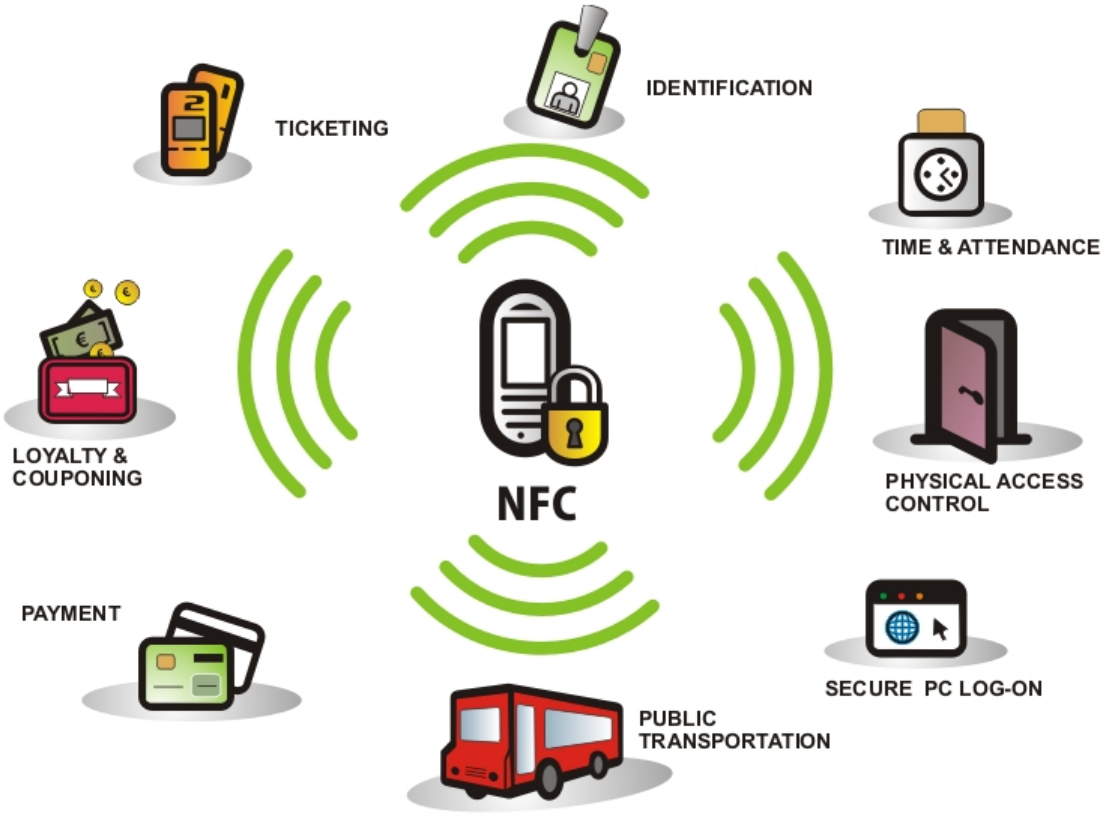


FIGURE 5 – Applications de la technologie Near Field Communication

2.4 Objectifs du stage

Du point de vue de l'approche 2, l'externalisation de certains calculs ne doit pas pour autant créer de brèche logique ou physique dans la sécurité du protocole. Bien que certains calculs puissent être faits en clair sans nuire la vie privée de l'utilisateur, d'autres doivent absolument être réalisés sur des données chiffrées.

Dans cette optique, j'ai dû dresser un état de l'art des systèmes de chiffrement homomorphes, afin de choisir de façon pertinente le(s)quel(s) utiliser pour nos besoins. Ces choix ont été effectués en prenant en compte des avantages et inconvénients de chacun de ces cryptosystèmes, que je vais présenter.

Deuxième partie

État de l'art

Notations

Dans la suite de cette partie, nous adopterons les notations suivantes :
$\square, \circ$: opération quelconque
$\otimes$: multiplication matricielle
$\oplus$: addition matricielle ou Xor selon le contexte
λ : paramètre de sécurité
pk : clé publique d'un cryptosystème asymétrique
sk : clé secrète d'un cryptosystème asymétrique
k : clé secrète d'un cryptosystème symétrique
m : message clair
c : message chiffré
$\mathcal{P}$: espace des textes clairs
$\mathcal{C}$: espace des textes chiffrés
$\mathcal{R}$: espace des aléas
$\mathcal{K}$: espace des clés

3 Quelques définitions et remarques

3.1 Cryptosystème

Le but de la cryptographie, est de permettre à Bob d'envoyer des messages à Alice avec des garanties sur :

- `l'intégrité du message` : le message n'a pas été modifié entre son envoi et sa réception
- `la confidentialité` : seuls Bob et Alice connaissent le contenu du message
- `la non-répudiation` : il est possible de prouver que Bob est bien l'expéditeur du message

Nous ne nous intéresserons pas aux autres aspects de la sécurité informatique tels que l'authentification, la disponibilité, ou le contrôle d'accès.

Définition 3.1.1. *Un cryptosystème est un ensemble d'algorithmes permettant de chiffrer et déchiffrer des messages. Éventuellement, un cryptosystème pourra être accompagné d'un algorithme permettant la génération des clés de chiffrement et déchiffrement.*

3.1.1 Asymétrique

En cryptographie asymétrique, Alice possède une paire de clés, une publique, l'autre privée. Tout le monde peut envoyer des messages à Alice en les chiffrant avec sa clé publique, seule Alice peut les déchiffrer en utilisant sa clé privée.

Définition 3.1.2. *Un système de chiffrement asymétrique consiste en trois algorithmes :*

1. *`KeyGen` : prend en entrée un paramètre de sécurité λ, retourne une paire de clés (pk, sk) où pk désigne la clé publique utilisée pour chiffrer les messages et sk celle pour déchiffrer*
2. *`Enc` : prend en entrée le message m à chiffrer et la clé publique pk, retourne un message chiffré c*
3. *`Dec` : prend en entrée un message chiffré c et la clé privée sk, retourne le message déchiffré m*

Dans la pratique, la gestion des clés publiques s'effectue par l'intermédiaire d'une infrastructure nommée PKI (Public Key Infrastructure), mais nous ne rentrerons pas dans les détails sur ce point.

3.1.2 Symétrique

Contrairement aux cryptosystèmes asymétriques, dans les symétriques ou encore, à clés secrètes, Bob et Alice doivent posséder la même clé k pour pouvoir dialoguer ensemble.

Définition 3.1.3. *Un cryptosystème symétrique consiste en deux algorithmes :*

1. *`Enc` : prend en entrée le message m à chiffrer et la clé secrète k, retourne le chiffré c de m*
2. *`Dec` : prend en entrée un message chiffré c et la clé secrète k, retourne le message déchiffré m*

Bien que ces cryptosystèmes soient généralement beaucoup plus rapides, les utilisateurs doivent partager un secret. Le problème réside dans le processus d'échange des clés ainsi que le nombre de clés à posséder pour discuter avec plusieurs utilisateurs.

3.1.3 Sémantiquement sûr

Depuis 1982, où Goldwasser et Micali ont introduit la notion de sécurité sémantique, tout "bon" système de chiffrement doit intégrer de l'aléa dans le processus de chiffrement[28]. Nous nous intéresserons également aux propriétés d'indistinguabilité, qui assurent sous certaines hypothèses, qu'un adversaire ne peut déterminer si deux chiffrés c_1 et c_2 chiffrent le même message m.

3.2 [Complètement] Homomorphe

Définition 3.2.1. *Un système de chiffrement homomorphe est un cryptosystème permettant de faire des calculs sur les données chiffrées. Formellement, si c_1 (respectivement c_2) est un chiffré de m_1 (respectivement m_2) il existe deux opérations $\square$ et $\circ$ telles que*

$$\texttt{Dec}(c_1 \square c_2) = \texttt{Dec}(c_1) \circ \texttt{Dec}(c_2) = m_1 \circ m_2$$

Typiquement, $\circ$ sera une addition ou une multiplication modulaire, mais ce n'est pas toujours le cas. Un système de chiffrement **complètement** homomorphe n'est rien d'autre qu'un système de chiffrement homomorphe où toute fonction peut être évaluée sur les données chiffrées. Comme toute fonction peut être exprimée comme un polynôme et qu'un polynôme consiste en une série d'additions et de multiplications, un système de chiffrement sera complètement homomorphe dès lors qu'il permettra d'évaluer un nombre arbitraire d'additions et de multiplications sur les données chiffrées.

Remarques sur les cryptosystèmes complètement homomorphes

Nous allons présenter les différents types d'instistingabilité évoqués section 3.1.3 et voir dans quelles mesures ils peuvent ou non s'appliquer aux systèmes de chiffrement homomorphe.

– **IND-CPA** : INDistinguishability under Chosen Plaintext Attack

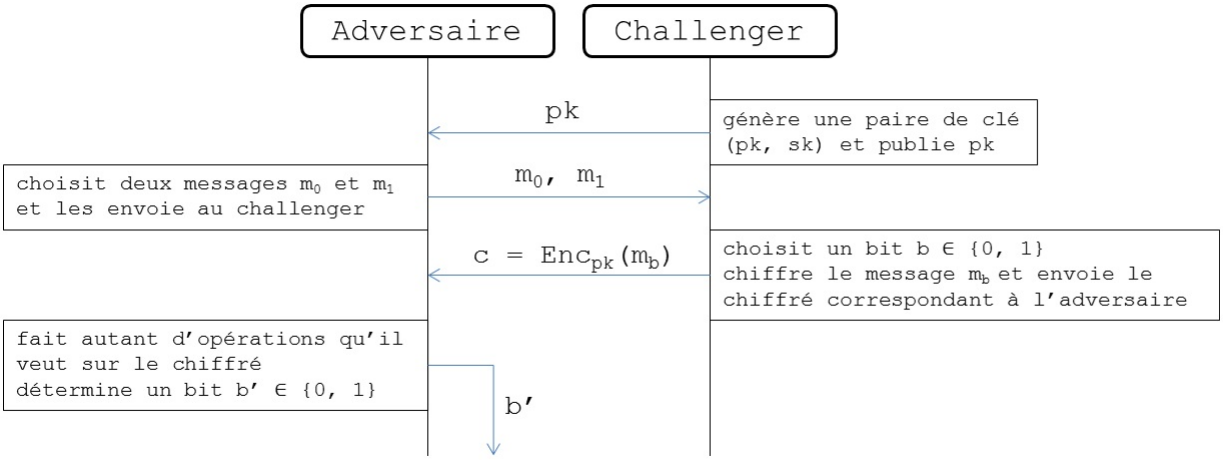


FIGURE 6 – Scénario d'attaque à clairs choisis

– **IND-CCA** : INDistinguishability under Chosen Ciphertext Attack

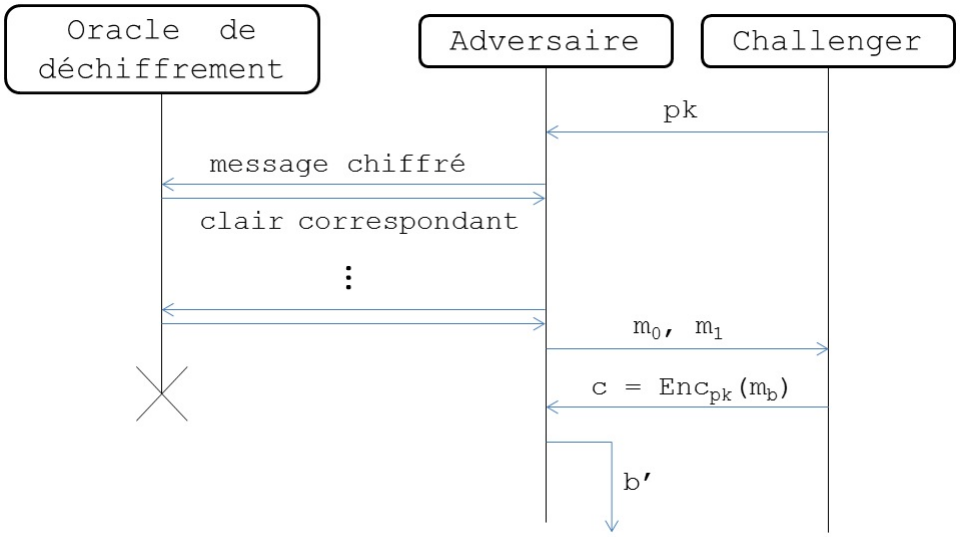


FIGURE 7 – Scénario d'attaque à chiffrés choisis

– **IND-CCA2** : INDistinguishability under Adaptive Chosen Ciphertext Attack

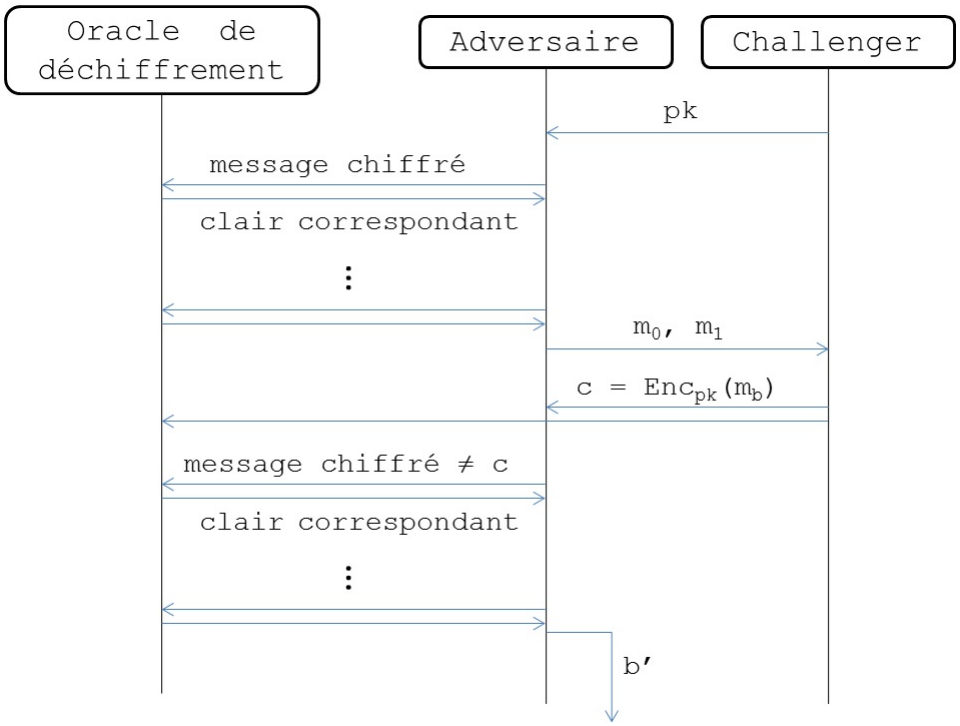


FIGURE 8 – Scénario d'attaque adaptative à chiffrés choisis

On dira qu'un cryptosystème résiste aux types d'attaques ci-dessus si la probabilité que l'adversaire retourne $b' = b$ est arbitrairement proche de $\frac{1}{2}$, qui correspond au cas où il répond au hasard. Formellement,

Définition 3.2.2. *Un cryptosystème est qualifié d'**IND-CPA** (resp. **IND-CCA**, resp. **IND-CCA2**) si pour tout adversaire probabiliste polynomial $\mathcal{A}$ et $\forall \epsilon > 0$,*

$$Pr[b' = b] < \frac{1}{2} + \epsilon$$

,

L'indistingabilité contre les attaques à clairs choisis (IND-CPA) est ce les cryptosystèmes modernes doivent posséder au minimum. La majorité des systèmes homomorphes possèdent cette propriété.
Cependant, les cryptosystèmes homomorphes étant par définition malléables, c'est-à-dire qu'on peut par exemple obtenir un chiffré de $2c$ à partir d'un chiffré de c, ils ne peuvent être résistants aux attaques adaptative à chiffrés choisis, comme le montre le diagramme suivant :

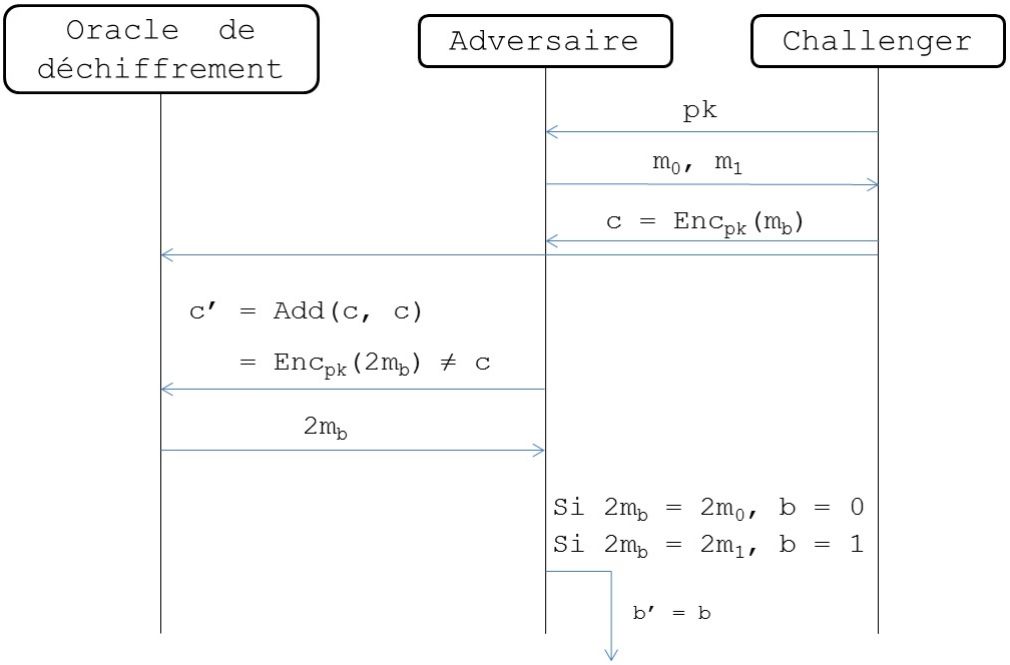


FIGURE 9 – Exemple d'attaque adaptative à chiffrés choisis sur les systèmes de chiffrement homomorphe

Concernant l'indistingabilité contre les attaques à chiffrés choisis, certains cryptosystèmes simplement homomorphes y parviennent, mais cela reste une question ouverte pour les systèmes de chiffrement complètement homomorphes.

Dans la suite, la majorité des cryptosystèmes homomorphes dont nous allons parler sont asymétriques. Cependant, l'exemple suivant est intéressant dans le sens où il permet d'évaluer un nombre arbitraire d'additions et de multiplications sur les données chiffrées. Dans cet exemple, $\square$ correspond à une multiplication matricielle, $\circ$ à une multiplication modulaire. La structure matricielle permet également de faire des additions sur les chiffrés.

3.3 Cas particulier : Xiao - Bastani - Yen

Auteurs : Liangliang XIAO - Osbert BASTANI - I-Ling YEN [58]
Année : 2012
KeyGen : L'algorithme de génération de la clé prend en entrée le paramètre de sécurité λ ainsi qu'un nombre m [7] tel que le schéma décrit ci-après puisse supporter $m \times ln(poly(\lambda))$ attaques à clairs choisis. Soient p_i, et q_i, $1 \leq i \leq m$ $2m$ nombres premiers de $\frac{\lambda}{2}$ bits chacun et deux à deux distincts, et $f_i = p_i q_i$ et $N = \prod_{i=1}^{m} f_i$. La clé secrète est une matrice k inversible tirée uniformément au hasard dans l'ensemble des matrices 4×4 à coefficients dans $\mathbb{Z}_N$

7. m doit être polynomial en λ pour qu'il y ait suffisamment de nombres premiers de tailles $\frac{\lambda}{2}$

Enc : Un message $x \in \mathbb{Z}_N$ est mis sous forme matricielle comme suit : $\begin{pmatrix} x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$.
Puis une valeur $r \in \mathbb{Z}_N$ est tirée aléatoirement, et on définit un ensemble de valeurs a_i, b_i et c_i comme suit : pour chaque i, exactement un des a_i, b_i et c_i est égal à x : $Pr(a_i = x) = 1 - \frac{1}{m+1}$, et $Pr(b_i = x) = Pr(c_i = x) = \frac{1}{2(m+1)}$. Les autres valeurs sont fixées à r. Puis, en utilisant le théorème des restes chinois, on définit a, b, et c comme étant les solutions des équations $a \ mod \ f_i = a_i$, $b \ mod \ f_i = b_i$, et $c \ mod \ f_i = c_i$. Le chiffré de x est alors :

$$C = k^{-1} \begin{pmatrix} x & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & c \end{pmatrix} k$$

Dec : $m = (kCk^{-1})_{(1,1)}$
Problème sous-jacent : Factorisation
Avantages : Complètement homomorphe
Inconvénients : Ce système n'est pas IND-CPA, du moins il ne peut supporter qu'un nombre prédéfini m d'attaques à clairs choisis.
Homomorphie : Si $c_1 = Enc_k(m_1)$ et $c_2 = Enc_k(m_2)$, alors

$$Dec_k(c_1 \otimes c_2) = m_1 \otimes m_2 \ et \ Dec_k(c_1 \oplus c_2) = m_1 \oplus m_2$$

4 Description des problèmes

4.1 Problèmes sous-jacents à la cryptographie basée sur la théorie des nombres

4.1.1 Facorisation

Tout nombre supérieur à 2 peut s'écrire sous produit de facteurs premiers. Cependant, étant donné un nombre, il est généralement difficile de trouver ces facteurs premiers. Le meilleur algorithme (non quantique) permettant de factoriser est le crible algébrique sur les corps de nombres, dont la complexité est $L_n[\frac{1}{3}, \sqrt[3]{\frac{64}{9}}] = exp((\sqrt[3]{\frac{64}{9}} + o(1))(ln(n))^{1/3}(ln(ln(n)))^{2/3})$. Le lecteur pourra trouver plus de détails dans l'annexe 11.

Le reste des problèmes disposent d'une version décisionnelle et d'une version calculatoire. Dans la version décisionnelle, l'attaquant a généralement pour but de distinguer une distribution d'une autre, alors que dans la version calculatoire il doit trouver une solution au problème.

4.1.2 Problème RSA

Le problème RSA correspond à l'extraction d'une racine e-ième modulo un entier $N = pq$. Actuellement, la meilleure façon de procéder est de factoriser le module N.
Formellement, étant donnés $N = pq$, $c = m^e \ mod \ N$, et $e > 1$, trouver m dans la version calculatoire, décider si un tel m existe dans la version décisionnelle.

4.1.3 Résidualité quadratique

Le problème de résidualité quadratique peut être vu comme une instance du problème RSA où $e = 2$. Dans la version décisionnelle, le but est de déterminer si c est un carré modulo $N = pq$, dans la version calculatoire, l'attaquant doit trouver x tel que $x^2 = c \ mod \ N$.

4.1.4 Problème de résidualité composée

Le problème de résidualité composée consiste à décider si $c \in \mathbb{Z}_{N^2}$ est une puissance N-ième modulo N^2, où $N = pq$, et de trouver ses racines dans la version décisionnelle.

4.1.5 Problème de résidualité d'ordre supérieur

Ce problème est une généralisation du problème de résidualité quadratique. Étant donnés des entiers $N = pq$ et $d|p-1$, déterminer si $x \in \mathbb{Z}_n$ est une puissance d-ième modulo N. Formellement : déterminer si $x = \alpha^d \ mod \ N$, et trouver α dans la version calculatoire.

4.1.6 Logarithme discret

Étant donnés un groupe cyclique $\mathcal{G}$, un générateur g de ce groupe, et $h = g^x \in \mathcal{G}$, le problème du logarithme discret consiste à trouver x.

Actuellement, il n'existe pas d'algorithme efficace permettant de résoudre ce problème. Cependant, si $\mathcal{G} = \mathbb{Z}_N$ avec $N = p \times q$, $pgcd(p, q) = 1$, et que p et q sont connus, il est possible de faire mieux que le calcul naïf des puissances successives de g jusqu'à trouver h [54].

4.1.7 Problème de Diffie-Hellman

Le protocole de Diffie Hellman est célèbre pour la solution qu'il apporte au problème d'échange des clés dans le domaine du chiffrement symétrique.

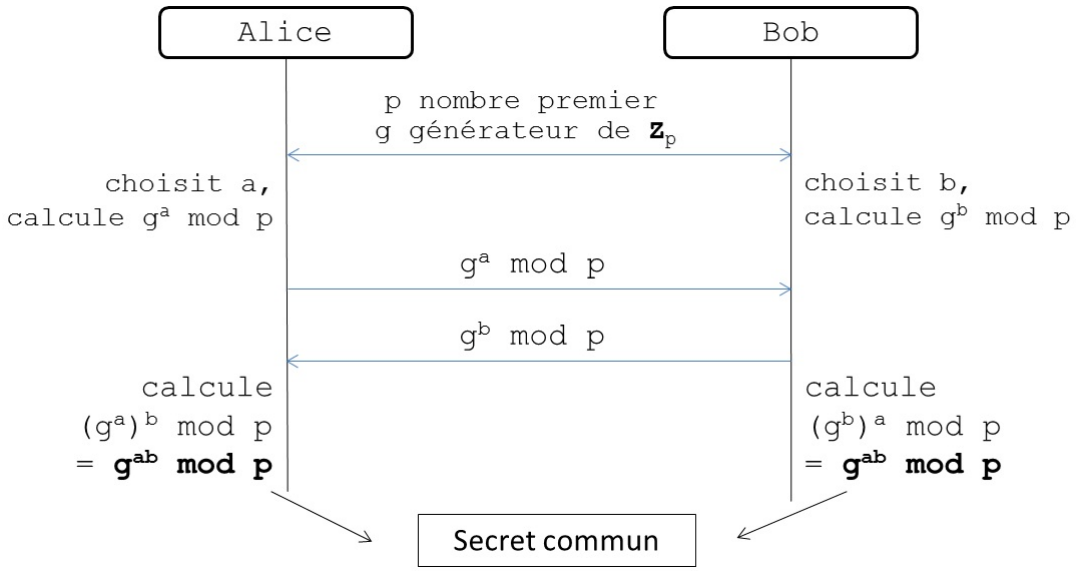


FIGURE 10 – Protocole d'échange de secret de Diffie - Hellman, 1976

Le problème de Diffie Hellman consiste à distinguer les distributions (g^a, g^b, g^{ab}) et (g^a, g^b, r) modulo le nombre premier p (pour r aléatoire), et à calculer g^{ab} à partir de g^a et g^b dans la version calculatoire.

4.2 Problèmes sous-jacents à la cryptographie sur les réseaux euclidiens

4.2.1 The Shortest Vector Problem

Définition 4.2.1. *Étant donnée une base B d'un réseau $\mathcal{L}$, le problème $Search - SVP$ consiste à trouver un vecteur non-nul $v \in \mathcal{L}$ tel que $\forall x \in \mathcal{L}\backslash\{0\}$ on a : $\|v\| \leq \|x\|$*

Comme beaucoup de problèmes, on peut s'intéresser à des versions approchées ainsi qu'à des versions décisionnelles ou d'optimisation. Ces versions donnent lieu aux définitions suivantes :

Définition 4.2.2. *Étant donnée une base B d'un réseau $\mathcal{L}$ et un facteur d'approximation $\gamma \in \mathbb{R}, \gamma \geq 1$, le problème $Search - SVP_\gamma$ consiste à trouver un vecteur non-nul $v \in \mathcal{L}$ tel que $\forall x \in \mathcal{L}\backslash\{0\}$ on a : $\|v\| \leq \gamma \cdot \|x\|$*

Définition 4.2.3. *Étant donnée une base B d'un réseau $\mathcal{L}$ et $\gamma \in \mathbb{R}, \gamma \geq 1$, le problème $Optimization - SVP_\gamma$ consiste à trouver un $d \in \mathbb{R}$ tel que $d \leq \lambda_1(\mathcal{L}) \leq \gamma \cdot d$.*

Définition 4.2.4. *Étant donnée une base B d'un réseau $\mathcal{L}$, $\gamma \in \mathbb{R}, \gamma \geq 1$, et $r \in \mathbb{Q}$, le problème $Promise - SVP$ consiste à déterminer si $\lambda_1(\mathcal{L}) \leq r$ ou si $\lambda_1(\mathcal{L}) > \gamma \cdot r$.*

Les versions exactes de ces problèmes sont obtenues en prenant comme paramètre d'approximation $\gamma = 1$. Selon l'ordre de grandeur du paramètre γ, le problème SVP_γ repose dans différentes classes de complexité.

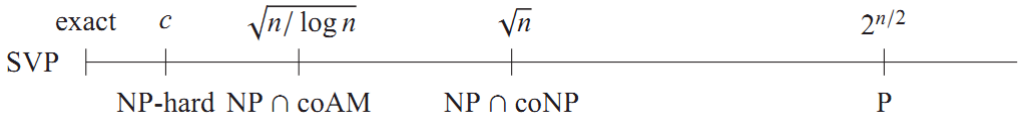


FIGURE 11 – Difficulté du problème du vecteur le plus court

4.2.2 The Closest Vector Problem

Définition 4.2.5. *Étant donnée une base B d'un réseau $\mathcal{L}$ et un vecteur cible $t \in Vect(\mathcal{L})$, le problème $Search - CVP$ consiste à trouver un vecteur $v \in \mathcal{L}$ tel que $\|v - t\| \leq dist(t, \mathcal{L}(B))$*

De façon analogue à SVP, il est d'usage de considérer des versions approchées de CVP, d'où les définitions suivantes :

Définition 4.2.6. *Étant donnée une base B d'un réseau $\mathcal{L}$, un vecteur cible $t \in Vect(\mathcal{L})$ et $\gamma \in \mathbb{R}, \gamma \geq 1$, le problème $Search - CVP_\gamma$ consiste à trouver un vecteur $v \in \mathcal{L}$ tel que $\|v - t\| \leq \gamma \cdot dist(t, \mathcal{L}(B))$*

Définition 4.2.7. *Étant donnée une base B d'un réseau $\mathcal{L}$, un vecteur cible $t \in Vect(\mathcal{L})$ et $\gamma \in \mathbb{R}, \gamma \geq 1$, le problème $Optimization - CVP_\gamma$ consiste à trouver $d \in \mathbb{R}$ tel que $d \leq dist(t, \mathcal{L}(B)) \leq \gamma \cdot d$*

Définition 4.2.8. *Étant donnée une base B d'un réseau $\mathcal{L}$, un vecteur cible $t \in Vect(\mathcal{L})$, $r \in \mathbb{Q}$ et $\gamma \in \mathbb{R}, \gamma \geq 1$, le problème $Promise - CVP_\gamma$ consiste à décider si $dist(t, \mathcal{L}(B)) \leq r$ ou si $dist(t, \mathcal{L}(B)) > \gamma \cdot r$.*

Les versions exactes de ces problèmes sont obtenues en prenant comme paramètre d'approximation $\gamma = 1$. Ici encore l'ordre de grandeur de γ a une influence directe sur la complexité de CVP_γ :

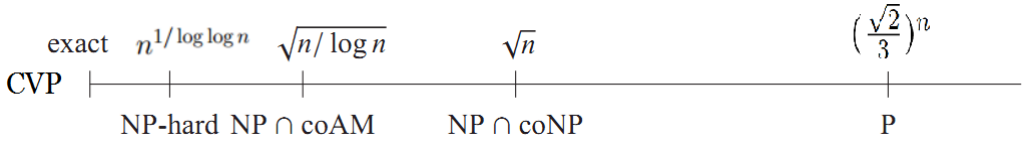


FIGURE 12 – Difficulté du problème du vecteur le plus proche

SVP versus CVP

Bien que ces deux problèmes soient NP-complets [57, 3], le schéma ci-dessous fait apparaître des différences entre ces deux problèmes.

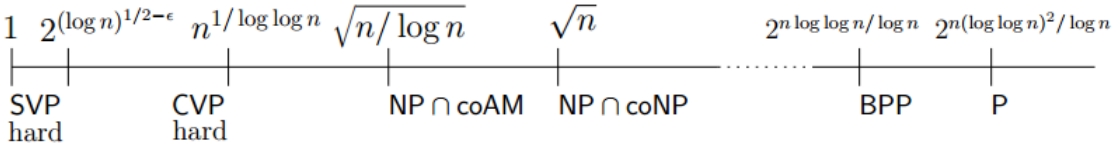


FIGURE 13 – Comparaison de SVP et CVP

Par ailleurs, Goldreich *et al.* [26] ont montré qu'approximer SVP n'était pas plus dur que d'approximer CVP.

4.2.3 The Absolute/Bounded Distance Decoding Problem

Bien que ces deux problèmes soient intitulés différemment, il ne s'agît que d'instances particulières de $Search - CVP_\gamma$:

- Bounded Distance Decoding Problem $(BDD) = Search - CVP_{1\leq\gamma\leq\lambda/2}$
- Absolute Distance Decoding Problem $(ADD) = Search - CVP_{\gamma>\rho}$

Ces deux valeurs ont une signification particulière ; dans BDD, si une solution existe, alors elle est unique, alors que dans ADD, l'existence d'une solution est assurée, mais pas son unicité...

4.2.4 The Shortests Independant Vectors Problem

Ce problème est relativement similaire à SVP, mise à part qu'ici on ne demande plus de trouver un vecteur court (éventuellement le plus court), mais plusieurs vecteurs courts, et linéairement indépendant :

Définition 4.2.9.

4.2.5 The [k] Short Integer Solution Problem

Le problème SIS consiste à trouver une solution non triviale à une équation linéaire modulo un entier, de de sorte que les coefficients de cette solution de soient pas trop grands. Dans [8], les auteurs généralisent le problème SIS, en donnant $k \geq 0$ solutions vérifiant le problème SIS. Formellement, on a la définition suivante :

Définition 4.2.10. *Pour tout $k \geq 0$, une instance du problème $k - SIS_{q,m,\beta}$ consiste en une matrice $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ et k vecteurs $\mathbf{e_1}, \ldots, \mathbf{e_k} \in \Lambda_q^{\perp}(\mathbf{A}) \subseteq \mathbb{Z}^m$ vérifiant $||\mathbf{e_i}|| \leq \beta$, $\forall\ 1 \leq i \leq k$. Le but est de trouver $\mathbf{v}\mathbb{Z}^m$ de sorte que :*

1. $||\mathbf{v}|| \leq \beta$
2. $\mathbf{A} \cdot \mathbf{v} = \mathbf{0}$ *mod* q, *c'est-à-dire* $\mathbf{v} \in \Lambda_q^{\perp}(\mathbf{A})$ *et*
3. $\mathbf{v} \notin \mathbb{Q} - Vect(\mathbf{e_1}, \ldots, \mathbf{e_k})$

Le problème SIS correspond simplement au cas où $k = 0$.

De plus, les auteurs de [8] ont montré qu'un adversaire $\mathcal{A}$ qui résolvait le problème k-SIS en dimension m pouvait être utilisé pour résoudre le problème SIS en dimension $m - k$.

4.2.6 The [Sparse] Subset Sum Problem

Le problème de la somme d'un sous ensemble est un problème prouvé NP-complet [31]. Nous allons le définir ainsi que sa variante dite "creuse".

Définition 4.2.11. *Une instance du Subset-Sum problème est donnée par un ensemble S de taille $|S| = n$, et une cible t, trouver un sous-ensemble $S' \subseteq S$ de taille $s \leq n$ tel que $\sum_{x_i \in S'} x_i = t$. Dans la version creuse de ce problème, $s << n$.*

La version creuse de ce problème est utilisée dans le cryptosystème [24] de Gentry. Sung Lee a montré [34] que les choix agressifs des paramètres facilitait la cryptanalyse (2 jours pour une probabilité de succès de 44%).

4.2.7 The General Learning With Errors Problem

Définition 4.2.12. *Pour un paramètre de sécurité λ fixé, soient $n = n(\lambda)$ une dimension entière, $f(x) = x^d + 1$, $d = 2^k$ un polynôme cyclotomique pour un $k \in \mathbb{N}$, $q = q(\lambda) \geq 2$ un module premier, $\mathcal{R} = \mathbb{Z}[x]/(f(x))$, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$, $\chi = \chi(\lambda)$ une distribution sur $\mathcal{R}$ d'écart type σ.*
Le problème GLWE consiste à distinguer les distributions $(a_i, < a_i, s > +e) \in \mathcal{R}_q^{n+1}$ et $(a_i, r) \in \mathcal{R}_q^{n+1}$ pour $a_i, s \in \mathcal{R}_q^n$ et $b \in \mathcal{R}_q$ uniformément (ou sans perte de sécurité, $s \leftarrow \chi^n$) et $e \leftarrow \chi$.

LWE et Ring-LWE
Le problème LWE correspond au problème GLWE instancié avec $d = 1$, alors que pour Ring-LWE, on choisit $n = 1$. Comme remarqué dans [11].

Regev a montré qu'un algorithme (éventuellement quantique) qui résolvait LWE dans le cas moyen impliquait un algorithme quantique qui résoud les pire cas de SIVP et gapSVP [48].

Lyubashevsky, Peikert et Regev ont montré qu'un algorithme qui résolvait Ring-LWE dans le cas moyen impliquait un algorithme quantique qui résoud les pire cas de SVP [36].

4.2.8 The Learning Parity with Noise Problem

Le problème Learning Parity with Noise (LPN) est exactement identique au problème LWE (donc $d = 1$), avec $q = 2$.

Par analogie, et à ma connaissance, il n'existe pas d'équivalent de LPN en version anneau. Il pourrait être intéressant de créer le problème suivant :

Définition 4.2.13. *Pour un paramètre de sécurité λ fixé, soient $f(x) = x + 1$, $\mathcal{R}_2 = \mathbb{Z}_2[x]/(f(x))$, $\chi = \chi(\lambda)$ une distribution sur $\mathcal{R}$ d'écart type σ.*
Le problème Ring-LPN consiste à distinguer les distributions $(a_i, < a_i, s > +e) \in \mathcal{R}_2^2$ et $(a_i, r) \in \mathcal{R}_2^2$ pour $a_i, s \in \mathcal{R}_2$ et $b \in \mathcal{R}_2$ uniformément et $e \leftarrow \chi$.

4.2.9 La conjecture de Micciancio et Regev [39]

Conjecture :
Il n'existe aucun algorithme en temps polynômial qui puissent approximer jusqu'à un facteur polynômial les problèmes basés sur les réseaux.

5 État de l'art des cryptosystèmes simplement homomorphes

La cryptographie asymétrique est apparue en 1978, lorsque Rivest, Shamir et Adleman publient leur cryptosystème RSA [50]. Ce cryptosystème est révolutionnaire dans le sens où il permet à tout le monde d'envoyer des messages chiffrés à une personne, sans que personne hormis le destinataire ne puisse déchiffrer. Ce cryptosystème donne naissance à une nouvelle ère : celle du chiffrement à clés publiques.

De nombreux cryptographes étudient ce système, et on se rend rapidement compte qu'il possède deux particularités :

- il est homomorphe multiplicativement : si $c_1 = Enc(m_1)$ et $c_2 = Enc(m_2)$, $c_1 \times c_2 = Enc(m_1 \times m_2)$. [8] Autrement dit, sans connaître m_1 ni m_2 mais seulement leur chiffrés, il est possible de calculer un chiffré de $m_1 \times m_2$.
- Étant donné un message $m \in \mathcal{P}$ et un chiffré $c \in \mathcal{C}$, il est possible de déterminer si $c = Enc(m)$ ou non.

La première propriété ne semble pas affecter la sécurité du système, mais génère cependant beaucoup d'interrogations. À un tel point que la même année, Rivest, Adleman et Dertouzos [49] conjecturent l'existence d'un système de chiffrement complètement homomorphe sous l'appellation de "privacy homomorphism", et proposent des schémas candidats. Cependant, ceux-ci ont tous été cassés par la suite.

La seconde particularité du système de chiffrement RSA, bien moins désirable, sera qualifié en 1982 de non-indistinguabilité contre les attaques à clairs choisis (non IND-CPA), par Goldwasser et Micali [28]. En effet, le fait que le processus de chiffrement soit déterministe permet de savoir a posteriori si un chiffré c chiffre un message m donné. Depuis que Goldwasser et Micali ont introduit la notion de chiffrement probabiliste, tout système de chiffrement doit intégrer de l'aléa dans le processus de chiffrement pour être considérer comme sûr. Le cryptosystème RSA, largement utilisé aujourd'hui a d'ailleurs été revisité afin de garantir cette propriété, une première fois dans une version OAEP [9] [6], puis une seconde fois dans une version OAEP+ [55]. Cependant, ces deux versions perdent le caractère homomorphe de la version initiale, ce pourquoi nous n'en parlerons pas plus.

Dans leur papier définissant la sécurité sémentique [28], Goldwasser et Micali proposent également un système de chiffrement, qui possède une particularité intéressante : étant donnés deux chiffrés $c_1 = Enc(m_1)$ et $c_2 = Enc(m_2)$, il est possible de calculer un chiffré de $m_1 \oplus m_2$. L'inconvénient majeur de ce cryptosystème est l'expansion du texte chiffré : $|\mathcal{P}| = 2$ et $|\mathcal{C}| = 2^{1024}$ pour un module $N = pq$ de 1024 bits.

Deux ans plus tard, El Gamal [23] publie un système de chiffrement qui reprend certaines idées du protocole d'échange de clés de Diffie-Hellman [18], dont l'hypothèse de

8. Certains détails sont volontairement omis, le lecteur pourra trouver plus d'informations en annexe 10

9. Optimal Asymmetric Encryption Padding

sécurité est différente des précédents cryptosystèmes. En effet, les précédents cryptosystèmes reposaient sur le problème RSA [10] et la résidualité quadratique respectivement, celui d'El Gamal repose sur l'hypothèse décisionnelle de Diffie-Hellman, qui se réduit au logarithme discret. D'autre part, comme RSA, ce cryptosystème est multiplicativement homomorphe.

En 1985, Miller [40] et Koblitz[33] suggèrent indépendamment l'utilisation des courbes elliptiques dans la cryptographie. La cryptographie sur les courbes elliptiques présentent deux intérêts majeurs par rapport aux autres cryptosystèmes : la taille des clés est bien moindre (160 bits contre 1024 pour une sécurité équivalente à $\simeq 2^{80}$), elle évolue beaucoup plus lentement (224 bits contre 2048 pour une sécurité équivalente à $\simeq 2^{112}$). D'ici peu, ce genre de cryptosystème sera plus efficace que les algorithmes asymétriques actuellement utilisés. Nous en reparlerons avec le cryptosystème de Boneh, Goh, et Nissim [9].

En 1993, Fellows et Koblitz publient "Polly Cracker" [20] : le premier système de chiffrement homomorphe capable d'évaluer des fonctions arbitraires sur les données chiffrées. Cependant, ce cryptosystème n'est pas "complètement" homomorphe dans le sens où la taille des chiffrés croît exponentiellement en la profondeur du circuit à évaluer.

En 1994, Benaloh [7] publie un cryptosystème dont la sécurité se résume au problème de la résidualité d'ordre supérieur, dont la particularité est la possibilité de choisir la taille des blocs à chiffrer. Cependant, ce cryptosystème dans sa version originale présentait un défaut sur le choix des paramètres comme l'ont remarqué et corrigé Fousse, Lafoucarde et Alnuaimi [21]. Certains paramètres devaient être choisis de façon précise pour que le déchiffrement soit correct avec une probabilité égale à 1. Toutefois, on remarquera que c'est le premier cryptosystème additivement homomorphe [11].

Puis en 1997, Naccache et Stern [41] publient un cryptosystème basé sur le problème du sac à dos. Il permet comme celui de Goldwasser-Micali d'effectuer des Xor. Cependant, le chiffrement est déterministe, ce crytposystème n'est donc pas IND-CPA. L'année suivante, Naccache et Stern proposent un autre cryptosystème [42] basé sur le problème résidualité d'ordre supérieur cette fois-ci. Il se trouve que ce cryptosystème généralise celui de Benaloh.

La même année, Okamoto et Uchiyama publient leur cryptosystème [44]. Il parviennent à trouver une nouvelle fonction à sens unique équivalente à la factorisation. Cependant la sécurité sémantique du cryptosystème n'est assurée que sous l'hypothèse du sous-groupe d'ordre p. Toutefois, leurs idées seront reprises et améliorées dans les années qui suivent. Ce cryptosystème est homomorphe additivement.

10. Tous les problèmes dont nous parlerons sont définis dans l'annexe 4.1

11. jusque là on savait faire des additions modulo 2 (Xor), maintenant on peut faire des additions modulo $r \geq 2$

En 1999, Coron, Naccache et Paillier [16] publient une version améliorée du cryptosystème d'Okamoto-Uchiyama, où le déchiffrement est optimisé. La même année, le même Paillier publie un cryptosystème qui reprend certaines idées d'Okamoto-Uchiyama[45]. Il donne également deux autres versions de son cryptosystème afin d'accélérer le déchiffrement. Cependant la deuxième version n'est plus IND-CPA, et modifie les propriétés homomorphes. D'autre part, Catalano, Gennaro, Howgrave-Graham, et Nguyen [13] ont suggéré d'utiliser un exposant de faible poids binaire afin d'accélérer le processus de chiffrement et un générateur de forme très particulière afin de remplacer une exponentiation modulaire par une multiplication. Cependant la sécurité du système se rapproche du problème RSA.

Enfin, toujours en 1999, Sander, Young et Yung modifie le cryptosystème de Goldwasser et Micali de sorte à remplacer le Xor homomorphe par un And homomorphe (qui correspond à une multiplication modulo 2) [52]. On notera que la technique utilisée pour modifier ce schéma est générique, et peut donc être utilisée pour transformer d'autres cryptosystèmes.

En 2001, Choi, Choi, et Won [15] modifient le cryptosystème d'Okamoto-Uchiyama en choisissant un générateur d'une forme particulière, afin d'accélérer le processus de déchiffrement. Cependant Sakurai et Takagi [51] soulignent qu'il n'est pas garanti que la fonction de chiffrement ne soit pas facile à inverser avec un générateur de cette forme.

En parallèle, Damgård et Jurik publient une généralisation du cryptosystème de Paillier [17], ainsi que deux versions modifiées, dont la longueur est flexible.

Motivé par l'utilisation des courbes elliptiques en cryptographie, Paillier publie une version de son cryptosystème en 2000 [46]. Galbraith montre en 2002 [22] que cette version possède une faille de sécurité : le calcul du logarithme discret sur les courbes utilisées est facile. Il suggère également une version sûre de Paillier sur les courbes elliptiques.

En 2005, Schmidt-Samoa et Takagi publie une généralisation de Paillier et Okamoto-Uchiyama [53]. En même temps, Boneh, Goh et Nissim publient un cryptosystème [9] sur les courbes elliptiques dont les propriétés homomorphes sont intéressantes. Pour la première fois, un cryptosystème permet de faire à la fois des additions et des multiplications. Plus précisément, leur cryptosystème permet d'effectuer un nombre arbitraire d'additions, une multiplication, puis un nombre arbitraire d'additions. Le nombre limité de multiplications provient du fait que cette multiplication est réalisée à l'aide d'un accouplement de Weil, qui change le groupe sous-jacent.

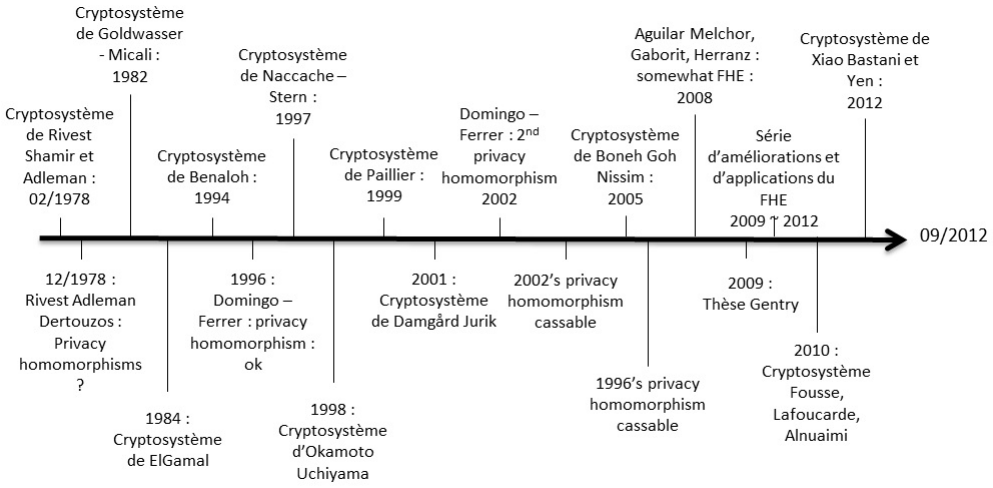


FIGURE 14 – Historique des cryptosystèmes

Conclusions

Nous avons vu un aperçu des cryptosystèmes asymétriques homomorphes développés au cours de ces 35 dernières années. Il est intéressant de constater la façon dont certains cherchent à généraliser, voire améliorer les précédant. Cette idée sera reprise dans l'état de l'art des systèmes de chiffrement complètement homomorphes qui suit.

Troisième partie

Description des systèmes de chiffrement complètement homomorphes

6 Cryptosystèmes complètement homomorphes

Notations

Dans la suite de cette partie, nous adopterons les notations suivantes :
λ : paramètre de sécurité, typiquement $\lambda = 100$
$\mathcal{R}_q$: anneau des polynômes à coefficients dans $\mathbb{Z}_q$ modulo $x^d + 1$, formellement, $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^d + 1)$
$\lfloor \cdot \rfloor$: Partie entière inférieure $\lfloor 1.2 \rfloor = 1$, $\lfloor 1.8 \rfloor = 1$
$\lceil \cdot \rceil$: Partie entière supérieure $\lceil 1.2 \rceil = 2$, $\lceil 1.8 \rceil = 2$
$\lfloor \cdot \rceil$: Partie entière la plus proche $\lfloor 1.2 \rceil = 1$, $\lfloor 1.8 \rceil = 2$
B : $B \in \mathbb{Z}^{n \times n}$, base du réseau $\mathcal{L} = \mathcal{L}(B)$
$\mathcal{L} = \mathcal{L}(B)$: Réseau euclidien de base B
δ : Paramètre de l'algorithme LLL, $\delta \in [\frac{1}{4}, 1]$

6.1 Quelques notions sur les réseaux euclidiens

Le premier système de chiffrement homomorphe dû à Gentry [24] étant basé sur les réseaux euclidiens, il nous a semblé utile que le lecteur connaissent certaines notions. D'autant plus que, comme nous l'avons déjà remarqué section 5, la plupart des cryptosystèmes complètement homomorphes qui seront présentés cherchent à améliorer les précédents, ou y apportent de légères modifications.

6.1.1 Premières définitions

Conceptuellement un réseau euclidien est un ensemble périodique de points. On trouve des réseaux un peu partout comme l'illustre l'image ci-dessous.

FIGURE 15 – Exemple de réseau euclidien

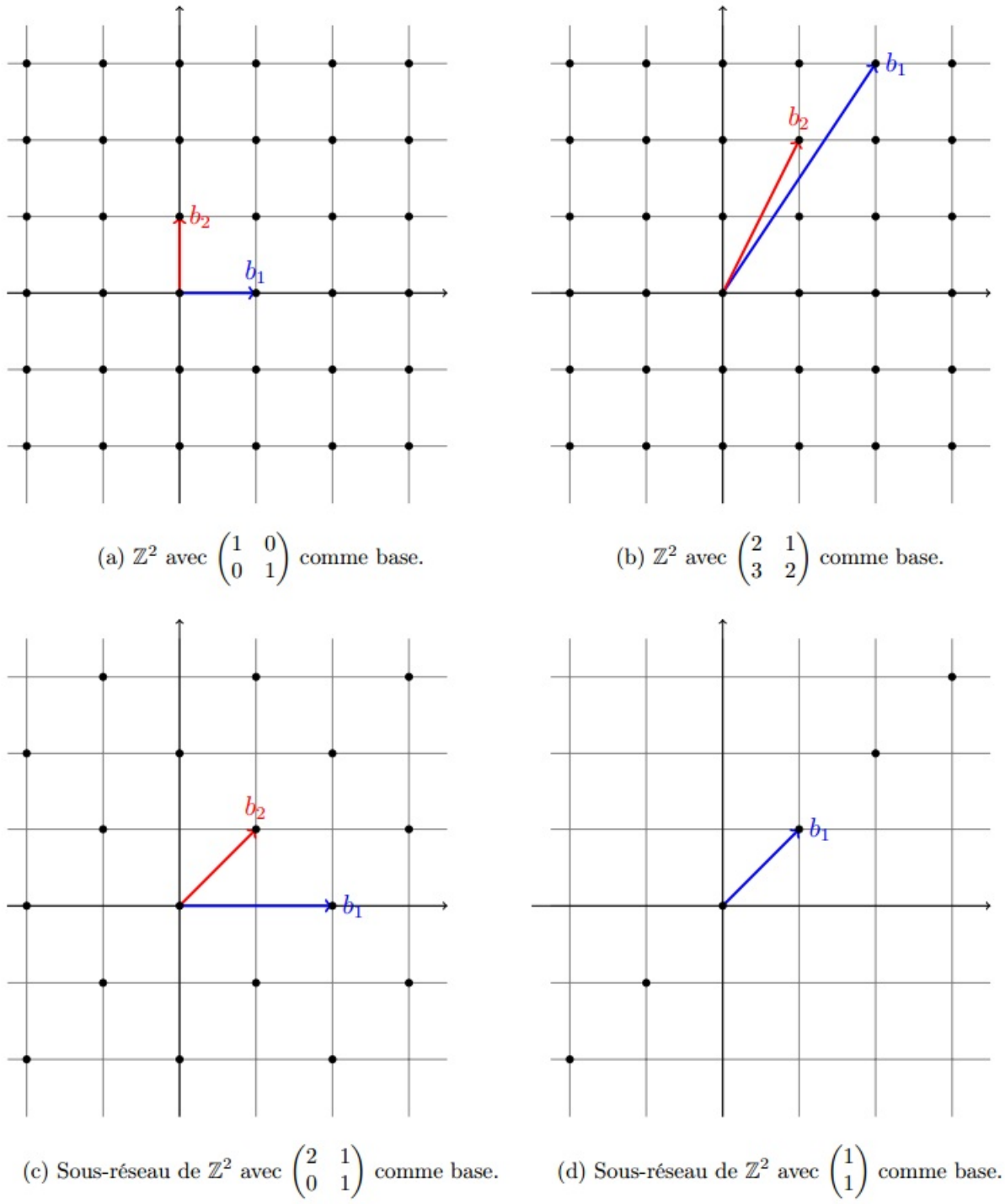
Un peu plus formellement, nous adopterons la définition suivante dans la suite de ce rapport :

Définition 6.1.1. *Soient $b_1, \ldots, b_n \in \mathbb{R}^m$ n vecteurs à m coordonnées. Le réseau qu'il engendre est donné par les combinaisons linéaires à coefficients entiers :*

$$\mathcal{L}(b_1, \ldots, b_n) = \{\sum_{i=1}^{n} x_i b_i \mid x_i \in \mathbb{Z}\}$$

Bien que l'ensemble des vecteurs $(b_1, \ldots, b_n)$ forme la ***base du réseau****, il est fréquent de considérer la matrice* $B = \begin{pmatrix} b_{11} & b_{21} & \ldots & b_{n1} \\ b_{12} & b_{22} & \ldots & b_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1m} & b_{2m} & \ldots & b_{nm} \end{pmatrix} \in \mathbb{R}^{m \times n}$ *formée des vecteurs $b_1, \ldots, b_n$ en colonne comme base du réseau. m est la* ***dimension*** *du réseau, et n est son* ***rang****. Lorsque $m = n$, on dit que le réseau est de* ***rang plein****.*

La figure suivante présente des exemples de réseaux à 2 dimensions.



(a) $\mathbb{Z}^2$ avec $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ comme base.

(b) $\mathbb{Z}^2$ avec $\begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$ comme base.

(c) Sous-réseau de $\mathbb{Z}^2$ avec $\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$ comme base.

(d) Sous-réseau de $\mathbb{Z}^2$ avec $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ comme base.

FIGURE 16 – Exemples de réseaux à deux dimensions.

Bien qu'il n'y ait pas d'obligation, les coordonnées des vecteurs formant la base du réseau seront souvent entières. Formellement $b_i \in \mathbb{Z}_m, \forall i \in \{1, \ldots, n\}$.
Dans la suite, on écrira $Vect(\mathcal{L}(B))$ ou juste $Vect(B)$ pour désigner l'espace vectoriel

engendré par les vecteurs de la base :

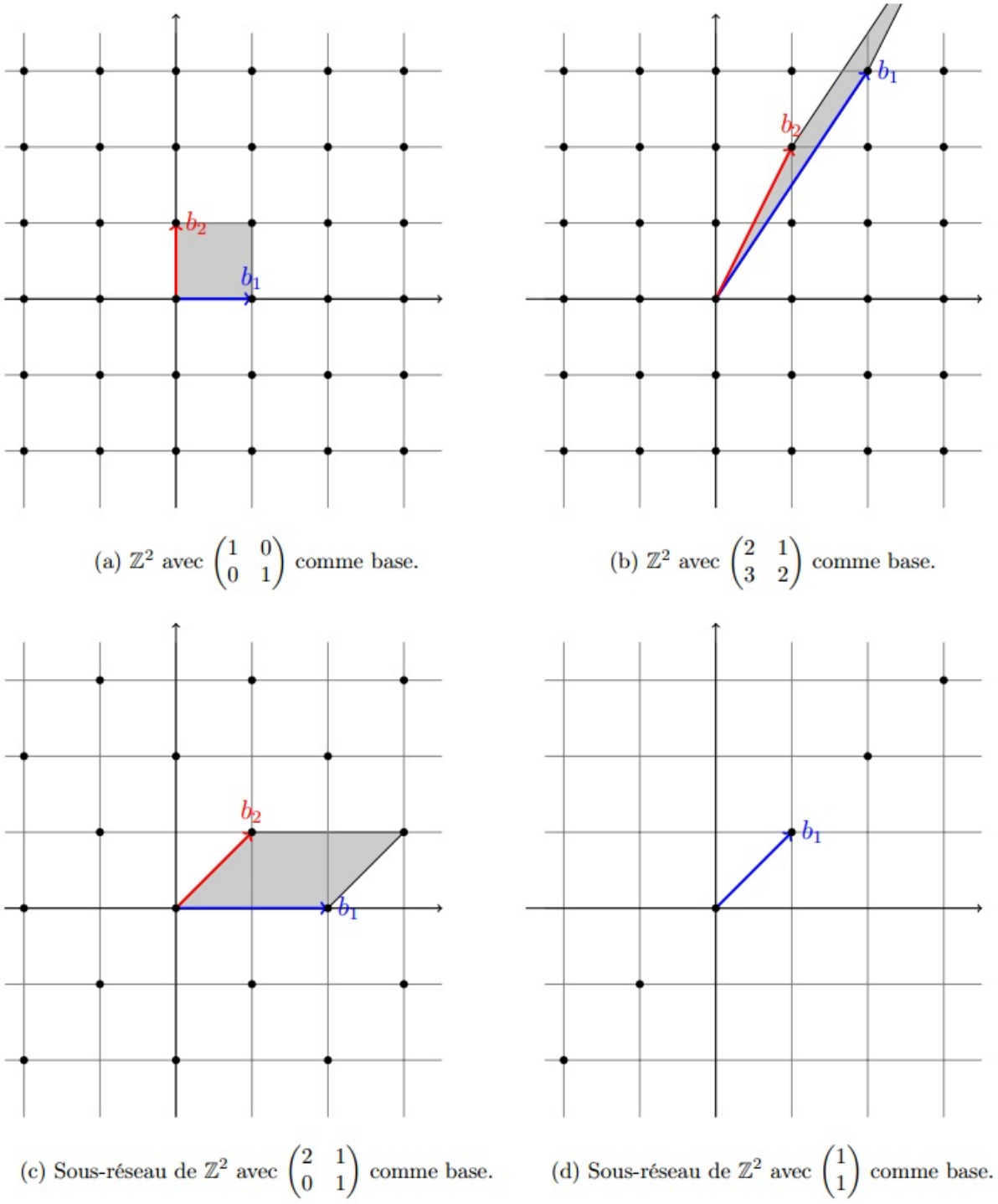
$$Vect(\mathcal{L}(B)) = Vect(B) = \{\sum_{i=1}^{n} y_i b_i | y_i \in \mathbb{R}\}$$

Parallélépipède fondamental

Définition 6.1.2. *Soit $\mathcal{L}(B)$ un réseau euclidien. Le parallélépipède fondamental de $\mathcal{L}(B)$ est défini par :*

$$\mathcal{P} = \mathcal{P}(B) = \{Bx | x \in [0, 1[^n\} = \{Bx | x \in [-\frac{1}{2}, \frac{1}{2}[^n\}$$

L'image suivante reprend les exemples donnés figure 16 et y intègre une copie du parallélépipède fondamental.



(a) $\mathbb{Z}^2$ avec $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ comme base.

(b) $\mathbb{Z}^2$ avec $\begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$ comme base.

(c) Sous-réseau de $\mathbb{Z}^2$ avec $\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$ comme base.

(d) Sous-réseau de $\mathbb{Z}^2$ avec $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ comme base.

FIGURE 17 – Parallélépipèdes fondamentaux de réseaux deux dimensions.

On remarque qu'en reportant une copie du parallélépipède fondamental à chaque point du réseau, on obtient l'espace vectoriel engendré par cette base.

Réseaux équivalents

Définition 6.1.3. *On dit que deux réseaux $\mathcal{L}(B_1)$ et $\mathcal{L}(B_2)$ sont équivalents si et seule-*

ment si il existe une matrice unimodulaire[12] U *à coefficient dans* $\mathbb{Z}$ *telle que* $B_1 = B_2U$. *On écrira* $\mathcal{L}(B_1) = \mathcal{L}(B_2)$

Déterminant

Définition 6.1.4. *Le déterminant* $det(\mathcal{L}(B))$ *d'un réseau* $\mathcal{L}(B)$ *est défini comme étant le volume du parallélépipède fondamental* $\mathcal{P}(B)$ *:*

$$det(\mathcal{L}) = vol(\mathcal{P}) = \sqrt{det(B^T B)}$$

Proposition 6.1. *Bien que la notion de parallélépipède fondamental dépende de la base choisie, la notion de déterminant a du sens car elle ne dépend pas de la base choisie.*

Démonstration. En effet, si B_1 et B_2 sont deux bases d'un même réseau, alors il existe U unimodulaire de sorte que $B_1 = B_2U$, donc

$$\begin{aligned} vol(\mathcal{P}(B_1)) &= \sqrt{det(B_1^T B_1)} \\ &= \sqrt{det((B_2U)^T(B_2U))} \\ &= \sqrt{det(U^T) \cdot det(B_2^T B_2) \cdot det(U)} \\ &= \sqrt{det(B_2^T B_2) \cdot det(U)^2} \\ &= \sqrt{det(B_2^T B_2)} \\ &= vol(\mathcal{P}(B_2)) \end{aligned}$$

□

6.1.2 Procédé d'orthogonalisation de Gram-Schmidt

Positionnement du problème
Dans les applications des réseaux euclidiens à la cryptographie, il est souvent appréciable de disposer d'une "bonne" base du réseau. Par "bonne" base, on entend que les vecteurs $b_1, \ldots, b_n \in \mathbb{R}^m$ ont une norme euclidienne $\updownarrow_2$ assez petite, et un faible défaut d'orthogonalité. Dans cette section, nous nous intéresserons à ce dernier point. Le premier sera traité dans la section 6.1.4.

Pour rappel, deux vecteurs $u, v \in \mathbb{R}^m$ sont orthogonaux si et seulement si leur produit scalaire est nul :

$$u \perp v \Leftrightarrow < u, v >= \sum_{i=1}^{m} u_i \cdot v_i = 0$$

Nous allons maintenant présenté le procédé d'orthogonalisation de Gram-Schmidt, qui étant donnés des vecteurs $(b_1, \ldots, b_n)$ retourne des vecteurs $(\tilde{b_1}, \ldots, \tilde{b_n})$ deux à deux orthogonaux entre eux.

12. Une matrice est dite unimodulaire si son déterminant est ± 1, ce qui impose implicitement qu'elle soit carrée

ALGORITHM

`Input` : $B = (b_1, \ldots, b_n) \in \mathbb{R}^m$ base du réseau $\mathcal{L}(B)$
`Output` : $\tilde{B} = (\tilde{b_1}, \ldots, \tilde{b_n})$ base orthogonale de $Vect(B)$

1. $\tilde{b_1} = b_1$
2. $\forall i \in \{2, \ldots, n\}, \tilde{b_i} = b_i - \sum_{j=1}^{i-1} \frac{<\tilde{b_j}, b_i>}{<\tilde{b_j}, \tilde{b_j}>} \cdot \tilde{b_j}$
3. Retourner $\tilde{B} = (\tilde{b_1}, \ldots, \tilde{b_n})$

Problème
Pour des espaces vectoriels, toute base peut être transformée en une base orthogonale, c'est d'ailleurs ce que fait le procédé de Gram-Schmidt. Cependant, le problème est plus compliqué pour les réseaux : le procédé d'orthogonalisation de Gram-Schmidt retourne des vecteurs qui ne sont pas nécessairement dans le réseau initial. Formellement si $B = (b_1, \ldots, b_n)$ est la base initial du réseau, et $\tilde{B} = (\tilde{b_1}, \ldots, \tilde{b_n})$, il est peu probable que $\mathcal{L}(B) = \mathcal{L}(\tilde{B})$. Cependant, ce procédé est utilisé dans l'algorithme LLL décrit section 6.1.4, pour réduire le défaut d'orthogonalité de la base donnée en entrée.

6.1.3 Minima successifs et théorème de Minkowski

Dans cette section, nous nous intéressons aux vecteurs courts. Typiquement, dans un réseau $\mathcal{L}(B)$, tout vecteur non-nul x à une longueur strictement positive $||x|| = \updownarrow_2(x) = \sqrt{x_1^2 + \ldots + x_m^2} > 0$. La question est de savoir si cette longueur est relativement petite par rapport aux autres vecteurs du réseau.

Définition 6.1.5. *Soit $\mathcal{L}(B)$ un réseau de base $B \in \mathbb{R}^{m \times n}$. Pour $i \in \{1, \ldots, n\}$ on définit le **i-ème minimum successif** λ_i par :*

$$\lambda_i = \inf r | dim(Vect(L \cap \bar{B}(0^m, r)) \geq i$$

où $\bar{B}(0^m, r) = \{x \in \mathbb{R}^m | \ ||x|| \leq r\}$ désigne la boule fermée centrée en $(0, \ldots, 0) \in \mathcal{L}(B)$ et de rayon r.

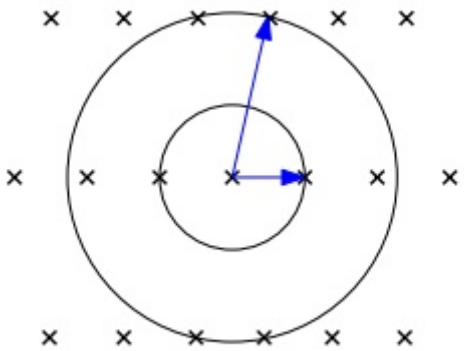


FIGURE 18 – $\lambda_1(\mathcal{L}(B)) = 1, \lambda_2(\mathcal{L}(B)) = 2.3$

$\lambda_1(\mathcal{L}(B))$ n'est rien d'autre que la distance du plus petit vecteur du réseau $\mathcal{L}(B)$. Comme nous le verrons dans la section 4.2.1, trouver cette valeur n'est pas évident. Cependant, le théorème suivant donne une borne inférieure - hélas assez large - sur cette distance.

Théorème 6.2. *Avec les notations précédentes :*

$$\lambda_1(\mathcal{L}(B)) \geq \min_{i\in\{1,\dots,n\}}\{||\tilde{b_i}||\}$$

Démonstration. Soient $x \in \mathbb{Z}^n$ un vecteur non-nul quelconque, et $j \in \{1,\dots,n\}$ le plus grand entier tel que $x_j \neq 0$. Alors

$$| < Bx, \tilde{b_j} > | = | < x_j\tilde{b_j}, \tilde{b_j} > | = |x_j| \cdot ||\tilde{b_j}||^2$$

D'après Cauchy Schwarz,

$$| < Bx, \tilde{b_j} > | \leq ||Bx|| \cdot ||\tilde{b_j}||$$

En combinant les deux inégalités, on obtient

$$||Bx|| \geq |x_j| \cdot ||\tilde{b_j}|| \geq ||\tilde{b_j}|| \geq \min_{i\in\{1,\dots,n\}}\{||\tilde{b_i}||\}$$

□

D'autre part, les minima successifs sont atteints, dans le sens où $\forall i \in \{1,\dots,n\}$, il existe un vecteur $v_i \in \mathcal{L}(B)$ tel que $||v_i|| = \lambda_i$

D'autre part, il existe des bornes supérieures pour les minima successifs, données par les théorèmes de Minkowski. Afin de présenter les deux théorèmes sur les bornes supérieures, nous aurons besoins des deux théorèmes suivant :

Théorème 6.3. *(Blichfield) Pour tout réseau $\mathcal{L} \subseteq \mathbb{R}^n$ de rang plein, et tout ensemble $S \subseteq \mathbb{R}^n$, si $vol(S) > det(\mathcal{L})$, alors il exite deux points $z_1, z_2 \in S$, $z_1 \neq z_2$ tels que $z_1 - z_2 \in \mathcal{L}$*

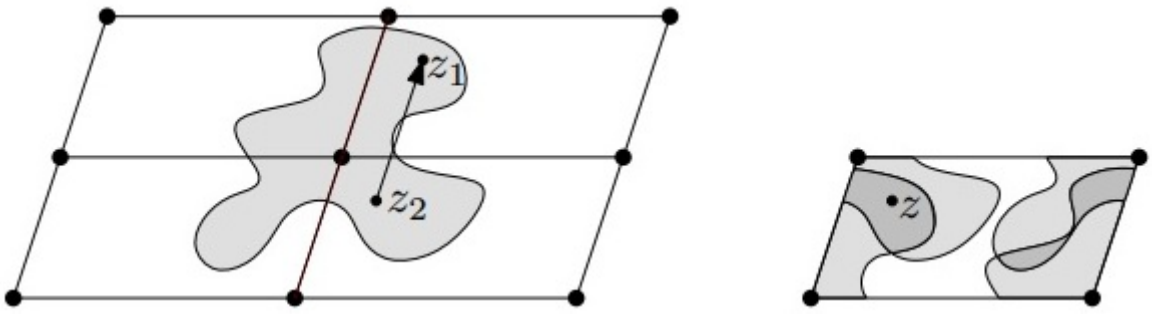


FIGURE 19 – On est sûrs de trouver deux points $z_1 \neq z_2$ tels que $z_1 - z_2 \in \mathcal{L}$ dans les zones en gris foncés

Démonstration. Soit B une base de $\mathcal{L} = \mathcal{L}(B)$. Pour $x \in \mathcal{L}(B)$, les ensembles $x+\mathcal{P}(B) = \{x+y | y \in \mathcal{P}(B)\}$ partitionnent $\mathbb{R}^n$. Soient $S_x = S \cap (x+\mathcal{P}(B)$ et $S = \bigcup_{x\in\mathcal{L}} S_x$ (voir figure 19). Comme cette union est disjointe, $vol(S) = \sum_{x\in\mathcal{L}(B)} vol(S_x)$. Soit $\tilde{S_x} = S_x \backslash \{x\} \subseteq \mathcal{P}(B)$. Alors $vol(\tilde{S_x}) = vol(S_x)$ et donc

$$\sum_{x\in\mathcal{L}} vol(\tilde{S_x}) = \sum_{x\in\mathcal{L}} vol(S_x) = vol(S) > vol(\mathcal{P}(B))$$

$\sum_{x\in\mathcal{L}} vol(\tilde{S_x}) = vol(\mathcal{P}(B))$ entraîne qu'il existe deux points $z_1, z_2 \in \mathcal{L}$ tels que $z_1 \neq z_2$ et $\tilde{S_{z_1}} \cap \tilde{S_{z_2}} \neq \emptyset$. Soit $z \in \tilde{S_{z_1}} \cap \tilde{S_{z_2}}$. Alors $z + z_1 \in S_{z_1} \subseteq S$ et $z + z_2 \in S_{z_2} \subseteq S$ et $z_1 - z_2 = (z + z_1) - (z + z_2) \in \mathcal{L}$. □

Le **théorème du corps convexe de Minkowski** suivant peut être vu comme un corollaire du théorème de Blichfield. Il affirme que pour tout ensemble convexe centré en l'origine suffisamment large contient au moins un vecteur non nul du réseau. Pour rappel, un ensemble S est centré en l'origine si pour tout $x \in S$, $-x \in S$, et il est convexe si pour tout couple de point (x, y), S contient tous les points du segment $[xy]$, formellement $\lambda \cdot x + (1 - \lambda) \cdot y \in S$, $\forall \lambda \in [0, 1]$.

Théorème 6.4. *Soit $\mathcal{L} \in \mathbb{R}^n$ un réseau de rang plein. Alors pour tout ensemble S convexe centré en l'origine, si $vol(S) > 2^n \cdot det(\mathcal{L})$, alors S contient un vecteur $v \in \mathcal{L}$ non-nul.*

Démonstration. Soit $\tilde{S} = \frac{1}{2}S = \{x | 2x \in S\}$. Alors $vol(\tilde{S}) = 2^{-n} vol(S) > det(\mathcal{L})$ par hypothèse. Donc d'après le théorème 6.3 de Blichfield, $\exists z_1, z_2 \in \tilde{S} / z_1 - z_2 \in \mathcal{L} \backslash \{0\}$. Par définition, $2z_1, 2z_2 \in S$, donc $-2z_2 \in S$ car S est centré en l'origine, et enfin $\frac{1}{2}(2z_1 - 2z_2) = (z_1 - z_2) \in S$ car S est convexe. La preuve de ce théorème est illustrée par la figure 20 □

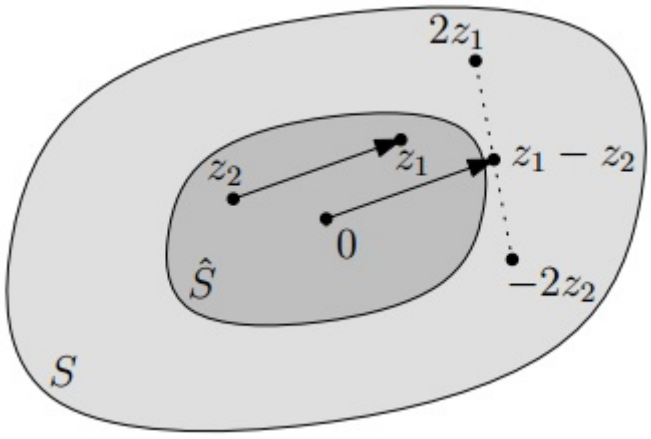


FIGURE 20 – Théorème du corps convexe de Minkowski

Proposition 6.5. *Le volume de la boule de rayon r en dimension n vérifie $vol(\mathcal{B}(0^n, r)) \geq (\frac{2r}{\sqrt{n}})^n$.*

Démonstration. Il suffit de remarquer que cette boule contient le cube $\{x \in \mathbb{R}^n | \forall i \in \{1,\dots,n\}, |x_i| < \frac{r}{\sqrt{n}}\}$. En effet soit x un tel élément, alors $||x|| = \sqrt{\sum_{i=1}^n x_i^2} < \sqrt{\sum_{i=1}^n (\frac{r}{\sqrt{n}})^2} = \sqrt{\sum_{i=1}^n \frac{r^2}{n}} = \sqrt{n \cdot \frac{r^2}{n}} = \sqrt{r^2} = r$ car $r > 0$. De plus $vol(\{x \in \mathbb{R}^n | \forall i \in \{1,\dots,n\}, |x_i| < \frac{r}{\sqrt{n}}\}) = vol(\{x \in \mathbb{R}^n | \forall i \in \{1,\dots,n\}, |x_i| = \frac{r}{\sqrt{n}}\}) = (\frac{2r}{\sqrt{n}})^n$, car $|x_i| < \frac{r}{\sqrt{n}}$ et donc $-\frac{r}{\sqrt{n}} < |x_i| < \frac{r}{\sqrt{n}}$, et cet intervalle a pour longueur $\frac{r}{\sqrt{n}} - (-\frac{r}{\sqrt{n}}) = \frac{2r}{\sqrt{n}}$ et ce pour chaque $i \in \{1,\dots,n\}$. □

Le premier théorème de Minkowski est une conséquence de cette propriété :

Théorème 6.6. *Pour tout réseau $\mathcal{L} \in \mathbb{R}^n$ de rang plein, $\lambda_1(\mathcal{L}) \leq \sqrt{n} \cdot \sqrt[n]{det(\mathcal{L})}$.*

Démonstration. Par définition, $\mathcal{B}(0^n, \lambda_1(\mathcal{L}))$ est un ensemble convexe centré en l'origine qui ne contient aucun point de $\mathcal{L}$. De plus,

$$(\frac{2\lambda_1(\mathcal{L})}{\sqrt{n}})^n \leq vol(\mathcal{B}(0^n, \lambda_1(\mathcal{L}))) \leq 2^n det(\mathcal{L})$$

d'après le théorème 6.4 de Minkowski et la proposition 6.5. Donc $2^n \cdot \lambda_1(\mathcal{L})^n \leq 2^n \cdot \sqrt{n}^n det(\mathcal{L})$ d'où $\lambda_1(\mathcal{L}) \leq \sqrt{n} \cdot \sqrt[n]{det(\mathcal{L})}$ □

Le second théorème de Minkowski renforce la borne supérieure obtenue par le théorème 6.6 précédent :

Théorème 6.7. *Pour tout réseau $\mathcal{L} \in \mathbb{R}^n$ de rang plein,*

$$\sqrt[n]{\prod_{i=1}^n \lambda_i(\mathcal{L})} \leq \sqrt{n} \cdot \sqrt[n]{det(\mathcal{L})}$$

Démonstration. Soit $x_1, \dots, x_n \in \mathcal{L}$ des vecteurs linéairement indépendants atteignant les minima successifs : $||x_i|| = \lambda_i$ et $\tilde{x_1}, \dots, \tilde{x_n}$ leur orthogonalisation de Gram-Schmidt. Soit

$$T = \{y\mathbb{R}^n | \sum_{i=1}^n (\frac{< y, \tilde{x_i} >}{||\tilde{x_i}|| \cdot \lambda_i})^2 < 1\}$$

Nous allons montrer qu'alors T ne contient aucun vecteur non-nul de $\mathcal{L}$. Soit $y \in \mathcal{L} \setminus \{0\}$ et soit $1 \leq k \leq n$ le plus petit entier tel que $||y|| \geq \lambda_k$. Alors $y \in Vect(\tilde{x_1}, \dots, \tilde{x_k})$, sinon $x_1, \dots, x_k, y$ forme $k+1$ vecteurs linéairement indépendants de longueur inférieure à λ_{k+1}. Alors,

$$\sum_{i=1}^n (\frac{< y, \tilde{x_i} >}{||\tilde{x_i}|| \cdot \lambda_i})^2 = \sum_{i=1}^k (\frac{< y, \tilde{x_i} >}{||\tilde{x_i}|| \cdot \lambda_i})^2 \geq \frac{1}{\lambda_k^2} \cdot \sum_{i=1}^k (\frac{< y, \tilde{x_i} >}{||\tilde{x_i}||})^2 = \frac{||y||^2}{\lambda_k^2} \geq 1$$

et donc $y \notin T$. D'après le théorème du corps convexe, $vol(T) \leq 2^n \cdot det(\mathcal{L})$, cependant

$$vol(T) = (\prod_{i=1}^n \lambda_i) vol(\mathcal{B}(0^n, 1)) \geq (\prod_{i=1}^n \lambda_i)(\frac{2}{\sqrt{n}})^n$$

Donc $(\prod_{i=1}^{n} \lambda_i)(\frac{2}{\sqrt{n}})^n \leq 2^n det(\mathcal{L})$, d'où

$$\sqrt[n]{\prod_{i=1}^{n} \lambda_i(\mathcal{L})} \leq \sqrt{n} \cdot \sqrt[n]{det(\mathcal{L})}$$

□

6.1.4 LLL

Dans cette section, nous allons discuter de l'algorithme de Lenstra, Lenstra et Lovász présenté dans [35], couramment appelé LLL, et qui permet de réduire une base d'un réseau, afin d'obtenir une base de ce même réseau, dont les vecteurs sont plus courts, et dont le défaut d'orthogonalité est réduit. Cet algorithme permet d'approximer le plus court vecteur en temps polynomial, mais à un facteur d'approximation $(\frac{2}{\sqrt{3}})^n$ exponentiel en la dimension du réseau. Nous allons présenter la notion de base "δ-LLL-réduites", le lecteur pourra trouver une description de l'algorithme LLL, ainsi qu'une analyse de sa complexité dans l'annexe 12.1.

Définition 6.1.6. *Une base $B = (b_1, \ldots, b_n)$ d'un réseau $\mathcal{L}(B)$ est δ-LLL-réduite, pour $\delta \in [\frac{1}{4}, 1]$ si :*

1. $\forall 1 \leq i \leq n$ *et* $j < i$ *on a :* $|\mu_{i,j}| < \frac{1}{2}$
2. $\forall 1 \leq i \leq n$ *on a :* $\delta \cdot ||\tilde{b_i}||^2 < ||\mu_{i+1,i}\tilde{b_i} + \tilde{b_{i+1}}||^2$

En réécrivant la seconde propriété, on obtient :

$$\delta \cdot ||\tilde{b_i}||^2 < ||\mu_{i+1,i}\tilde{b_i} + \tilde{b_{i+1}}||^2 = \mu_{i+1,i}^2||\tilde{b_i}||^2 + ||\tilde{b_{i+1}}||^2 \text{ , car } \tilde{b_i} \perp \tilde{b_{i+1}}$$

Donc

$$||\tilde{b_{i+1}}||^2 \geq (\delta - \mu_{i,j}^2) \cdot ||\tilde{b_i}||^2 \geq (\delta - \frac{1}{4}) \cdot ||\tilde{b_i}||^2,$$

ce qui peut être interprété comme "$\tilde{b_{i+1}}$ n'est pas beaucoup plus court que $\tilde{b_i}$".

Proposition 6.8. *Soient $(b_1, \ldots, b_n) \in \mathbb{R}^{n \times n}$ une base δ-LLL-réduite. Alors $||b_1|| \leq (\frac{2}{\sqrt{4\delta - 1}})^{n-1} \cdot \lambda_1(\mathcal{L})$*

Démonstration. $\lambda_1(\mathcal{L}) \geq \min_i ||\tilde{b_i}||$ d'après le théorème 6.2, donc

$$||\tilde{b_n}||^2 \geq (\delta - \frac{1}{4}) \cdot ||\tilde{b_{n-1}}||^2 \geq \ldots \geq (\delta - \frac{1}{4})^{n-1} \cdot ||\tilde{b_1}||^2 = (\delta - \frac{1}{4})^{n-1} \cdot ||b_1||^2.$$

Alors pour tout i, on a :

$$||b_1|| = ||\tilde{b_1}|| \leq (\delta - \frac{1}{4})^{-(i-1}/2||\tilde{b_i}|| \leq (\frac{4\delta - 1}{4})^{-(n-1}/2||\tilde{b_i}||,$$

et comme cette inégalité est vraie pour tout i, elle l'est aussi pour le $\min_i ||\tilde{b_i}||$:

$$||b_1||(\frac{2}{\sqrt{4\delta - 1}})^{n-1} \min_i ||\tilde{b_i}|| \leq (\frac{2}{\sqrt{4\delta - 1}})^{n-1} \lambda_1(\mathcal{L}).$$

□

La proposition ci-dessus nous fournit donc une approximation sur le plus court vecteur. Pour $\delta = \frac{3}{4}$, une valeur régulièrement utuilisée, on obtient $||b_1|| \leq 2^{(n-1)/2} \cdot \lambda_1(\mathcal{L})$. L'algorithme LLL permettant d'approcher le plus court vecteur d'un réseau est décrite en annexe 12.1 avec une explication sur son fonctionnement.

L'intérêt de cet algorithme est qu'il permet dans bien des cas de résoudre des instances des problèmes décrits section 4.2, en temps polynomial, à un facteur d'approximation exponentiel. Il existe également deux autres algorithmes (HKZ et BKZ) permettant d'obtenir des facteurs d'approximations polynomiaux voire constants, mais en temps exponentiel, ce qui justifie la conjecture de Micciancio et Regev définie section 4.2.9.

6.2 Les tentatives de chiffrement complètement homomorphe

Le chiffrement complètement homomorphe aurait pu dater de 1998, date à laquelle Hoffstein, Pipher et Silverman [29] publient leur cryptosystème "NTRU" basé sur les réseaux, et permettant d'évaluer à la fois des additions et multiplications sur les données chiffrées. Cependant, la taille des chiffrés augmente exponentiellement en la profondeur du circuit à évaluer. Nous présenterons la technique de Brakerski et Vaikuntanathan [12] pour faire face à ce problème. Bien qu'ils ne l'utilisent pas pour NTRU mais pour leur cryptosystème, cette technique reste suffisamment générique pour être appliquée à NTRU.

Différentes approches ont été adoptées dans le but d'obtenir un cryptosystème complètement homomorphe. Il est possible d'obtenir des cryptosystèmes additivement homomorphes à partir de codes correcteurs d'erreurs ou de réseaux, mais la multiplication pose généralement des problèmes [27, 32, 1, 47].

Cependant, les cryptosystèmes [2] d'Aguilar-Melchor, Gaborit et Herranz et [4] d'Armknecht et Sadeghi permettent tous les deux l'évaluation de multiplications, toujours aux prix de chiffrés dont la taille croît exponentiellement en la profondeur du circuit à évaluer. Le premier est basé sur les réseaux, le second sur des codes de Reed-Solomon.

Ces schémas sont intéressant dans la mesure où ils suggèrent dors et déjà de chiffrer les messages en y ajoutant du bruit, comme le fait le schéma de Gentry qui sera publié en 2009.

6.3 La thèse de Gentry : le début d'une ère nouvelle

Dans la suite, on utilisera les notations de [11]. Nous étudierons dans cette section les récents schémas de chiffrement complètement homomorphe, et nous verrons dans quelle mesure chacun d'eux apporte une amélioration par rapport aux précédents. Nous commencerons par la thèse de Craig Gentry, qui fût le premier cryptosystème complètement homomorphe, et qui utilise des idéaux d'anneaux de polynômes. La sécurité de ce schéma repose sur les réseaux idéaux, que nous définirons.

En quelques mots, le schéma de Gentry chiffre des messages en leur ajoutant du bruit. Les opérations homomorphes sont effectuées impactant également ce bruit. Lorsque le bruit devient trop important, Gentry applique une procédure dite de "bootstrap", que nous traduirons pas par réamorçage ou réinitialisation, qui permet de réduire ce bruit à un niveau acceptable. Concrètement, cette procédure consiste à évaluer la fonction de déchiffrement de manière homomorphe. La suite de cette section explique formellement ce que nous venons de décrire en quelques lignes.

Dans sa thèse, Gentry choisit son bruit/erreur e dans un idéal I d'un anneau $\mathcal{R}$: $e = kI \in I \subset \mathcal{R}$. Le message est alors chiffré en ajoutant ce bruit au message, formellement : $c = m + kI$. La procédure de déchiffrement consiste à retirer l'erreur. Les propriétés

homomorphes du systèmes sont quasi-immédiates, pour $c_1 = m_1 + k_1 I$ et $c_2 = m_2 + k_2 I$, on a :

$$c_1 + c_2 = m_1 + m_2 + (k_1 + k_2)I \; et \; c_1 \cdot c_2 = m_1 \cdot m_2 + (m_1 k_2 + m_2 k_1 + k_1 k_2)I$$

On peut déjà remarquer que le bruit est beaucoup plus affecté par une multiplication qu'une addition. Approximativement, une addition double le bruit alors qu'une multiplication l'élève au carré. Si un trop grand nombre d'opérations est effectué, le bruit devient trop grand et la procédure de déchiffrement retourne un message erroné. Cependant, en évaluant régulièrement la procédure de déchiffrement de manière homomorphe, on peut éviter que cela arrive, et c'est exactement ce que fait le bootstrapping : étant donné un chiffré c de m, cette procédure retourne un chiffré c' de m où le bruit k' contenu dans c' est plus petit que le bruit k contenu dans c : $||k'|| < ||k||$.

Cependant, pour pouvoir évaluer la fonction de déchiffrement de façon homomorphe, il est nécessaire que celle-ci soit suffisamment simple, ce qui n'est pas le cas initialement. Pour faire face à ce problème, Gentry réduit la complexité du circuit de déchiffrement [13] en publiant un ensemble de vecteurs dont la somme d'une partie d'entre eux est égale à la clé secrète. Ce problème est connu sous le nom de "[Sparse] Subset Sum Problem", et est prouvé NP-complet.

L'idée de Gentry est de partir d'un schéma dit "somewhat homomorphic encryption scheme" qui peut évaluer des additions et des multiplications tant que le bruit n'est pas trop grand, et de lui appliquer la procédure de bootstrap. Le schéma initial est basé sur le "Ideal Coset Problem", cependant, pour appliquer la procédure de bootstrap en réduisant la complexité du circuit de déchiffrement repose sur le "Sparse Subset Sum Problem".

6.4 Cryptosystèmes complètement homomomorphes

6.4.1 Cryptosystème de van Dijk, Gentry, Halevi et Vaikuntanathan

Ce schéma se veut plus simple qu'efficace comme le précisent les auteurs [56]. L'idée est que la somme de deux nombres proches d'un multiple de p est également proche d'un multiple de p, de même pour le produit.

- `KeyGen` : Soit p un entier impair de η bits. Pour $i = 0, \ldots, \tau$, soit $x_i = pq_i + r_i$ où $q_i \leftarrow [0, \frac{2^\gamma}{p}[$ et $r_i \leftarrow]-2^\rho, 2^\rho[$. Réorganiser les x_i de sorte que x_0 soit le plus grand. $pk = (x_0, \ldots, x_\tau)$.
- `Enc` : Soit $m \in \{0, 1\}$. Choisir un sous-ensemble aléatoire S de $(0, 1, \ldots, \tau)$ et un nombre $r \in]-2^\rho, 2^\rho[$ au hasard, retourner $c = [m + 2r + 2\sum_{i \in S} x_i]_{x_0}$
- `Dec` : Retourner $m = [[c]_p]_2$
- `Add` : Soit $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$. $[c_1 + c_2]_{x_0}$ est un chiffré valide de $m_1 + m_2$ tant que le bruit n'est pas trop grand.
- `Mul` : Soit $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$. $[c_1 \times c_2]_{x_0}$ est un chiffré valide de $m_1 \times m_2$ tant que le bruit n'est pas trop grand.

13. "Squashing the Decryption Circuit : The Encrypter Starts Decryption !" cf. [24] sections 1.5 et 10

Pour que le schéma soit sûr, van Dijk et al. suggèrent de choisir $r \approx 2^{\sqrt{\eta}}$, $q \approx 2^{\eta^3}$, $\rho = \omega(\log \lambda)$, $\eta \geq (\lambda \log^2 \lambda)$, $\gamma = \omega(\eta^2 \log \lambda)$, et $\tau \geq \gamma + \omega(\log \lambda)$. Par exemple, pour $\rho = \lambda$, $\eta = \tilde{O}(\lambda^2)$, $\gamma = \tilde{O}(\lambda^5)$ et $\tau = \gamma + \lambda$, le schéma a une complexité de $\tilde{O}(\lambda^{10})$. Bien que ce schéma ne soit pas efficace, il a le mérite d'être on ne peut plus simple.

La sécurité de ce schéma se réduit au problème "Approximate-GCD" : étant donnés des entiers proches d'un multiple d'un nombre caché, le but est de trouver ce nombre. Ce problème a déjà été étudié par Howgrave-Graham [30] dans le cas où seulement deux multiples sont donnés. Les paramètres (notamment les q_i) sont choisis afin d'éviter des versions généralisées de son attaque.

De façon à être complètement homomorphe, van Dijk et al. appliquent une procédure de bootstrapping, de façon analogue à celle de Gentry, c'est-à-dire en publiant un indice sur la clé secrète dans la clé publique. Ceci est réalisé en assumant en plus que le "Sparse Subset Sum Problem" est dûr.

L'avantage majeur de ce schéma réside en sa simplicité. Cependant, aux regards de la taille des paramètres à adopter, il n'est pas applicable actuellement. Une optimisation naturellement possible serait de prouver que le problème approximate-GCD reste dûr, en réduisant la taille des paramètres.

6.4.2 Cryptosystème de Brakerski et Vaikuntanathan

Dans [12], les auteurs introduisent les processus de relinéarisation (ou changement de clé) et de réduction de module (ou de dimension). La relinéarisation leur permet de baser leur cryptosystème sur LWE plutôt que sur des hypothèses de complexité relatives aux idéaux, tandis que la réduction de dimension réduit naturellement la complexité de la fonction de déchiffrement et permet également de diminuer la taille des chiffrés. Il n'y donc plus de nécessité de passer par une phase de "squashing" comme devait le faire Gentry, et qui lui avait coûté une hypothèse de sécurité supplémentaire.

La relinéarisation est utiliée pour rendre le schéma somewhat homomorphic, et le bruit est géré par le changement de dimension. Nous présentons ici les algorithmes de bases, en omettant les clés d'évaluation dans un premier temps par soucis de clarté :

- `KeyGen` : Soient λ le paramètre de sécurité, $n > 0$ polynômial en λ, $k > 0$ polynômial en λ et n, et q impair sous-exponentiel en n. Soient $\mathbf{sk} = (s[1], \ldots, s[n]) \leftarrow \mathbb{Z}_q^n$ et $\mathbf{A} \leftarrow \mathbb{Z}_q^{k \times n}$ uniformément au hasard et $\mathbf{e} \leftarrow \chi^k$, $\mathbf{pk} = (\mathbf{A} | v = \mathbf{As} + 2\mathbf{e}) \in \mathbb{Z}^{k \times (n+1)}$
- `Enc` : Pour chiffrer un message $m \in \{0, 1\}$, choisir $\mathbf{r} \in \{0, 1\}^k$ aléatoirement, $\mathbf{c} = (\mathbf{a}, b)$ avec $\mathbf{a} = \mathbf{A}^T \mathbf{r}$ et $b = \mathbf{v}^T \mathbf{r} + m$.

– `Dec` : Étant donné $\mathbf{c} = (\mathbf{a}, b)$, calculer

$$\begin{aligned}[[b- < \mathbf{a}, \mathbf{sk} >]_q]_2 &= [[\mathbf{v}^T\mathbf{r} + m - (\mathbf{A}^T\mathbf{r})^T\mathbf{sk}]_q]_2 \\ &= [[\mathbf{v}^T\mathbf{r} + m - \mathbf{r}^T(\mathbf{v} - 2\mathbf{e}]_q]_2 \\ &= [m + 2\mathbf{e}^T\mathbf{r}]_2 \\ &= m\end{aligned}$$

La procédure de déchiffrement est plutôt simple dans le sens où elle consiste en une équation linéaire $\mathcal{L}_{(\mathbf{a},b)}(\mathbf{sk}) = [[b- < \mathbf{a}, \mathbf{sk} >]_q]_2$ en les coefficients de la clé secrète, dépendant du chiffré $\mathbf{c} = (\mathbf{a}, b)$. Pour $\mathbf{c_1} = Enc_{\mathbf{pk}}(\mathbf{m_1}) = (\mathbf{a_1}, b_1)$ et $\mathbf{c_2} = Enc_{\mathbf{pk}}(\mathbf{m_2}) = (\mathbf{a_2}, b_2)$, il est assez simple de voir que

$$[[\mathcal{L}_{(\mathbf{a_1},b_1)}(\mathbf{sk}) + \mathcal{L}_{(\mathbf{a_2},b_2)}(\mathbf{sk})]_q]_2 = [\mathbf{m_1} + \mathbf{m_2}]_2$$

Cependant, $[[\mathcal{L}_{(\mathbf{a_1},b_1)}(\mathbf{sk}) \times \mathcal{L}_{(\mathbf{a_2},b_2)}(\mathbf{sk})]_q]_2$ est une équation quadratique, impliquant des termes en $\mathbf{sk_i sk_j}$. Pour faire face à ce problème, Brakerski et Vaikuntanathan changent de clé :

$$\begin{aligned}[[\mathcal{L}_{(\mathbf{a_1},b_1)}(\mathbf{sk}) \times \mathcal{L}_{(\mathbf{a_2},b_2)}(\mathbf{sk})]_q]_2 &= [[([[b_1- < \mathbf{a_1}, \mathbf{sk} >]_q]_2) \cdot ([[b- < \mathbf{a}, \mathbf{sk} >]_q]_2)]_q]_2 \\ &= [[x_0 + \sum_{i=1}^{n} x_i \mathbf{sk_i} + \sum_{1 \le i \le j \le n} x_{i,j} \mathbf{sk_i sk_j}]_q]_2\end{aligned}$$

La taille du chiffré est maintenant de 1 coefficient x_0, plus n coefficients x_i, plus $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ coefficients $x_{i,j}$ soit $\frac{(n+1)(n+2)}{2} > n$ au total. La relinéarisation est processus surprenant dans le sens où elle permet de réduire la taille du chiffré de $\frac{(n+1)(n+2)}{2}$ à $n + 1$. Ceci est accompli en publiant des "chiffrés" des coefficients linéaires et quadratiques de la clé secrète [14] [12]. Soient $(\mathbf{a_i}, b_i)$ (resp. $(\mathbf{a_{i,j}}, b_{i,j})$) des chiffrés de x_i (resp. $x_{i,j}$) sous une nouvelle clé secrète sk'. Alors l'équation

$$[[x_0 + \sum_{i=1}^{n} x_i \mathbf{sk_i} + \sum_{1 \le i \le j \le n} x_{i,j} \mathbf{sk_i sk_j}]_q]_2$$

peut se réécrire

$$[[x_0 + \sum_{i=1}^{n} x_i (b_i - < \mathbf{a_i}, \mathbf{sk}' >) + \sum_{1 \le i \le j \le n} x_{i,j} (b_{i,j} - < \mathbf{a_{i,j}}, \mathbf{sk}' >)]_q]_2$$

qui est linéaire en sk'. Le nouveau chiffré c' correspond alors aux vecteur des coefficients de l'équation linéaire ainsi obtenue (il y en $n + 1$ car sk' est une clé généré de façon analogue à sk).

14. Bien que ces coefficients soient dans $\mathbb{Z}_q$, Brakerski et Vaikuntanthan les chiffrent bit à bit. Nous ne nous égarerons pas dans ces détails au cours de cette discussion.

Le schéma présenté peut jusque là évaluer des circuits de profondeur au plus $D = n^\epsilon$ pour une constant quelconque $\epsilon < 1$. La procédure de déchiffrement est un polynôme sk, de degré au minimum $max(n, log\ q)$, ce qui est supérieur à ce qui est toléré par ce schéma. En d'autres termes, ce schéma n'est pas encore bootstrappable. Le schéma de Gentry [24] ne l'était pas non plus, chose qu'il a corrigé grâce au squashing qui permet de simplifier la complexité de la fonction de déchiffrement, au prix d'une hypothèse de sécurité supplémentaire. Dans leur papier, Brakerski et Vaikuntanathan parviennent à trouver une astuce pour s'en passer.

Des deux procédures, la réduction de dimension est la plus impressionnante ! Les opérations qui y sont effectuées sont plutôt contre intuitive à ce qui est habituellement effectué à base de réseau... Nous avons vu que la relinéarisation permettait, outre de réduire la taille du chiffré, d'obtenir un chiffré c' sous sk' à partir d'un chiffré c sous sk de sorte que $Dec_{sk'}(c') = Dec_{sk}(c)$. C'est-à-dire, et c'est également la raison de la seconde appellation de cette procédure : elle permet de changer de clé. Les auteurs se sont donc posé la question de savoir s'il était possible que les deux clés n'aient pas la même dimension, voire pas le même module. Il s'avère que la réponse à ces deux question est oui. Ainsi, en choisissant typiquement $n = k^c$, $q = 2^{n^\epsilon}$, et $p = poly(k)$, nous serons capables d'évaluer de manière homomorphe des fonctions de degré $D = n^\epsilon = k^{c\times\epsilon}$ et choisir c suffisamment grand de sorte qu'il sera suffisant d'évaluer la fonction de déchiffrement de paramètres $(k, log\ p)$, formellement $Dec_{sk'}(c') = [[b' - < \mathbf{a}', sk' >]_p]_2$, pour $c' = (\mathbf{a}', b')$ et $sk' \in \mathbb{Z}_p^k$.

L'idée sous-jacente utilisée par Brakzerski et Vaikuntanathan est que $\mathbb{Z}_p$ peut être utilisé comme approximation de $\mathbb{Z}_q$ en multipliant par $\frac{p}{q}$ et en arrondissant, ajoutant cependant une faible erreur. Les paramètres publics à utiliser pour passer de sk à sk' sont notés $(\mathbf{a_i}, \tau, b_{i,\tau}) \in \mathbb{Z}_p^{k+1}$ avec

$$b_{i,\tau} =< \mathbf{a}_{\mathbf{i},\tau}, \mathbf{t} > +e + \lfloor \frac{p}{q} \cdot 2^\tau \cdot sk_i \rceil.$$

Ici, l'erreur ajoutée n'est plus $2e$, mais simplement e car $2 \cdot \frac{q}{p}$ n'est pas entier. Afin de ne pas introduire trop d'erreur supplémentaire, on divise par 2, on effectue la réduction de dimension, puis on remultiplie par 2. Lorsque l'on arrondie, on introduit une erreur supplémentaire de taille au plus $\frac{1}{2}$. En arrondissant ainsi, on obtient :

$$2^\tau \cdot sk_i = \frac{q}{p} \cdot (b_{i,\tau} - < \mathbf{a}_{\mathbf{i},\tau}, \tau >),$$

ce qui permet de convertir une équation linéaire en les coefficients de sk en une équation linéaire en les coefficients de sk'.

Afin d'achever la description de ce cryptosystème, nous allons parler des clés d'évaluation evk. Ce schéma permet d'évaluer tous les circuits de profondeur au plus L (correspondant à des fonctions de degré au plus 2^L). Les clés d'évaluation permettent de passer d'une étape $i + 1$ à i [15]. Elles consistent en les chiffrés des coefficients linéaires et quadratiques de $sk^{(i)}$ sous la clé secrète $sk^{(i+1)}$.

15. On commence au niveau L jusqu'à atteindre le niveau 0

6.4.3 Cryptosystème de Brakerski, Gentry, et Vaikuntanathan

La majorité des idées issues de [11] proviennent de [12] et sont décrites en section 6.4.2. Cependant, la contribution majeure de cet article provient du raffinement du procédé de réduction de la dimension (ou changement de module) de sorte à mieux gérer l'erreur contenu dans les chiffrés. De plus, leur technique ne nécessite plus de clés d'évaluation ce pourquoi, et selon la volonté des auteurs (et dans un soucis de clarté), ce processus sera par la suite nommé gestion du bruit (voire noise management).

Leur principale constatation est qu'en changement "naïvement" de module de q à p, si $p < q$ alors évidemment, l'erreur sera réduite, mais le module sera réduit proportionnellement, et donc le rapport taille de l'erreur / taille du module ne diminuera pas. Or c'est ce rapport qui détermine la quantité d'opérations homomorphes encore possibles. Cependant, les auteurs constatent qu'il n'y a pas que ce rapport qui est important, mais également la taille de l'erreur en elle-même. À titre d'exemple, si $q = x^k$, et qu'on dispose de deux chiffrés mod q dont l'erreur est de taille x, en les multipliant, on obtient un chiffré dont la taille de l'erreur est approximativement x^2. Si on effectue une autre multiplication, cette taille devient x^4, puis x^8, puis x^16, et ainsi de suite. L'erreur croît exponentiellement !

Ainsi, la taille de l'erreur impacte la quantité d'opérations homomorphes encore possibles. Cependant, en choisissant une échelle décroissante de module $q_i \approx q/x^i$, pour $i < k$, après une multiplication, la taille du bruit devient x^2. En réduisant le module à $q_1 = q/x = x^{k-1}$, la taille du bruit repasse à x. En remultipliant les chiffrés (maintenant sous le module q_1), la taille du bruit reddevient x^2. En changeant le module en $q_2 = q_1/x$, le bruit redescend à x, et ainsi de suite. Concrètement, leur technique permet de maintenir la quantité de bruit à un niveau essentiellement constant. Cette technique représente une amélioration exponentielle par rapport à ce qui se faisait précédemment.

Une autre optimisation présentée dans [11] est le batching, qui consiste à empaqueter plusieurs messages dans un même chiffré. Bien que l'idée d'utiliser un espace de textes clairs plus grand ait déjà été abordée dans [12], elle suggérait seulement d'utiliser $\mathbb{Z}_p$ comme espace clairs plutôt que $\mathbb{Z}_2$, avec p et q premiers entre eux. Ici, les auteurs suggèrent d'utiliser un nombre premier p tel que $p = 1 \ mod \ 2d$, où d est une puissance de 2. Ils utilisent le fait que sur le corps cyclotomique $R = \mathbb{Z}[x]/(x^d + 1)$, le polynôme $x^d + 1$ se factorise complètement modulo p, c'est-à-dire qu'on peut l'écrire comme produit de facteurs linéaires :

$$x^d + 1 \ mod \ p = \prod_{i=1}^{d} (x - \alpha_i),$$

où $\alpha_i = \alpha^{2i+1}$ pour α racine primitive $2d$-ième de l'unité.

Autrement dit, le nombre premier p peut se factoriser en un produit d'idéaux $\mathfrak{p}_i$ dans R, où $\mathfrak{p}_i$ est l'idéal engendré par $(p, x-\alpha_i)$. Dans ce cas, le théorème des restes chinois s'applique, et on obtient que

$$R_p = \mathbb{Z}_p[x]/(x^d+1) \cong R_{\mathfrak{p}_1} \times R_{\mathfrak{p}_2} \times \ldots \times R_{\mathfrak{p}_d}.$$

Ainsi, on peut empaqueter jusqu'à d messages $m_i \in R_{\mathfrak{p}_i}$, pour $1 \leq i \leq d$, et le fait d'évaluer un circuit sur R_p avec $x \in R_p^n$ en entrée revient à évaluer ce circuit sur $R_{\mathfrak{p}_i}$ avec le projeté de x sur $R_{\mathfrak{p}_i}^n$ comme entrée.

Pour finir avec ce cryptosystème, malgré ses bonnes performances asymptotiques sans bootstrapping, Brakerski, Gentry et Vaikuntanathan suggèrent de réintroduire la procédure bootstrapping non pas comme une nécessité, mais plutôt comme une optimisation. En effet, bien qu'on puisse évaluer des circuits de profondeur L pour L arbitraire, lorsque L devient grand, la procédure de génération des clés devient longue, et la taille des chiffrés dans les premiers niveaux extrêmement grande. Leur point de vue est qu'il est préférable de fixer L petit, et appliquer la procédure de bootstrapping au besoin, pour les circuits dont la profondeur dépasse cette borne. Les auteurs proposent également une version originale de leur cryptosystème dans laquelle les modules ne sont plus de larges entiers, mais des idéaux, ce qui permet d'obtenir des espaces de textes clairs exponentiellement grands en le paramètre de sécurité, mais nous ne nous attarderons pas sur ce point.

6.4.4 Cryptosystème de Brakerski

Le schéma décrit dans [10] est intéressant, et s'écarte légèrement de la lignée des cryptosystèmes complètement homomorphes précédents dans la mesure où Brakerski propose ici de chiffrer les messages non plus dans le bit de poids faible, mais dans celui de poids fort, comme le faisait Regev dans son cryptosystème [48] basé sur LWE. Ce cryptosystème présente notamment les avantages suivants :

1. il est invariant par mise à l'échelle,
2. il ne nécessite pas de changement de module,
3. il n'y a pas de restriction sur le module (hormis sa taille),
4. il n'y a pas de restriction sur le choix de la distribution de la clé secrète,
5. il existe une réduction classique au problème GapSVP.

Avant de rentrer dans les détails, nous allons décrire les principaux algorithmes de ce cryptosystème :

– `KeyGen` : Soit $n = \lambda$ le paramètre de sécurité, $q = q(\lambda)$ le module, $\chi = \chi(\lambda)$ une distribution bornée par B sur $\mathbb{Z}$, et $N = (n+1)\cdot(log\ q + O(1))$. Générer $\mathbf{s} = (s_1, \ldots, s_n) \leftarrow \mathbb{Z}_q^n$ et $\mathbf{A} \leftarrow \mathbb{Z}_q^{N\times n}$ uniformément, et $\mathbf{e} \leftarrow \chi^N$. Soit $\mathbf{b} = [\mathbf{As}+\mathbf{e}]_q$. Retourner $\mathbf{pk} = (\mathbf{b}|-\mathbf{A}) \in \mathbb{Z}_q^{N\times(n+1)}$, et $\mathbf{sk} = (1|\mathbf{s}) \in \mathbb{Z}_q^{n+1}$.

– `Enc` : Pour chiffrer un message $\mathbf{m} = (m, 0, \ldots, 0) \in \mathbb{Z}_2^{n+1}$, choisir $\mathbf{r} \leftarrow \mathbb{Z}_2^N$ au hasard et calculer

$$\mathbf{c} = [\mathbf{pk}^T\mathbf{r} + \lfloor\frac{q}{2}\rfloor \cdot \mathbf{m}]_q \in \mathbb{Z}_q^{n+1}.$$

– `Dec` :

$$
\begin{aligned}
[\lfloor 2 \cdot \frac{[< \mathbf{c}, \mathbf{sk} >]_q}{q} \rceil]_2 &= [\lfloor 2 \cdot \frac{[< [\mathbf{pk}^T \mathbf{r} + \lfloor \frac{q}{2} \rfloor \cdot \mathbf{m}]_q, (1, \mathbf{s}) >]_q}{q} \rceil]_2 \\
&= [\lfloor 2 \cdot \frac{\lfloor \frac{q}{2} \rfloor \cdot m + \mathbf{r}^T \mathbf{pk} \cdot (1, \mathbf{s})}{q} \rceil]_2 \\
&= [\lfloor 2 \cdot \frac{\lfloor \frac{q}{2} \rfloor \cdot m + \mathbf{r}^T \mathbf{b} - \mathbf{r}^T \mathbf{A} \mathbf{s}}{q} \rceil]_2 \\
&= [\lfloor 2 \cdot \frac{\lfloor \frac{q}{2} \rfloor \cdot m + < \mathbf{r}, \mathbf{e} >}{q} \rceil]_2 \\
&= [\lfloor 2 \cdot \frac{\lfloor \frac{q}{2} \rfloor \cdot m + e}{q} \rceil]_2 \\
&= [\lfloor 2 \cdot (\lfloor \frac{1}{2} \rfloor \cdot m + \frac{e}{q}) \rceil]_2 \\
&= [\lfloor m + 2 \cdot \frac{e}{q} \rceil]_2 \\
&= m
\end{aligned}
$$

lorsque l'erreur e n'est pas trop grande, ce qui est le cas si $e \leftarrow \chi^N$ avec $|\chi| \leq B$. Pour effectuer des opérations homomorphes, Brakerski considère les fractions des chiffrés : $\tilde{\mathbf{c}} = \frac{\mathbf{c}}{q}$. Alors $< \tilde{\mathbf{c}}, \mathbf{sk} >= \frac{1}{2} \cdot m + \tilde{e} + I$ avec $I \in \mathbb{Z}$ et $|\tilde{e}| = |\frac{e}{q}| = \epsilon$. Une addition est réalisée de façon assez simple en calculant

$$\tilde{\mathbf{c_1}} + \tilde{\mathbf{c_2}}_q = \frac{1}{2} \cdot (m_1 + m_2) + (\tilde{e_1} + \tilde{e_2}) + (I_1 + I_2 \ mod \ q$$

alors qu'une multiplication est réalisée en calculant $2 \cdot \mathbf{c_1} \otimes \mathbf{c_2}$, qui peut être déchiffré en utilisant le produit tensoriel de la clé privée :

$$
\begin{aligned}
2 \cdot \tilde{\mathbf{c_1}} \otimes \tilde{\mathbf{c_2}}, \mathbf{sk} \otimes \mathbf{sk} > &= 2 \cdot < \tilde{\mathbf{c_1}}, \mathbf{sk} > \cdot < \tilde{\mathbf{c_2}}, \mathbf{sk} > \\
&= 2 \cdot (\frac{1}{2} m_1 + \tilde{\mathbf{e_1}} + I_1) \cdot (\frac{1}{2} m_2 + \tilde{\mathbf{e_2}} + I_2) \\
&= \frac{1}{2} m_1 m_2 + 2(\tilde{e_1} I_2 + \tilde{e_2} I_1) + \tilde{e_1} m_2 + \tilde{e_2} m_1 + (m_1 I_2 + m_2 I_1 + 2 I_1 I_2).
\end{aligned}
$$

De façon impressionnante, le terme $\tilde{e_1} \tilde{e_2}$ qui provoquait un agrandissement quadratique du bruit ne pose plus problème car $|\tilde{e_i}| = |\frac{e_i}{q}| = \epsilon$, donc $|\tilde{e_1} \tilde{e_2}| = \epsilon^2 << \epsilon$. Le terme le plus grand devient $2(\tilde{e_1} I_2 + \tilde{e_2} I_1)$, dont la valeur absolue est majorée par $O(||\mathbf{s_1}||) \cdot \epsilon$. De façon à ramener le chiffré à sa taille initiale après une multiplication, Brakerski utilise la technique de gestion du bruit de [11].

6.4.5 Cryptosystème de Fan et Vercauteren

Dans [19], Fan et Vercauteren repprennent le schéma de Brakerski décrit dans la section 6.4.4 précédente, et transpose le problème sous-jacent de LWE à Ring-LWE. Si le schéma de Brakerski s'inspirait de celui de Regev [48], celui de Fan et Vercauteren s'inspire de celui de Lyubashevski, Peikert et Regev [36] :

– `KeyGen` : Soit $R = \mathbb{Z}[x]/(x^d + 1)$ avec d puissance de 2, χ une distribution sur R. L'espace des textes clairs sera R_t [16], notons $\Delta = \lfloor \frac{q}{t} \rfloor$. Soient $a \leftarrow R_q$ uniformément et $e \leftarrow \chi$, $sk \in R_q = \mathbb{Z}_q[x]/(x^d + 1)$ générée selon χ et $b = -(a \cdot s + e)\ mod\ q$. Alors $pk = (b, a) \in R_q^2$.
– `Enc` : Pour chiffrer $m \in R_t$, soient $u, e_1, e_2 \leftarrow \chi$. Calculer

$$c = (c_0, c_1) = ([b \cdot u + e_1 + \Delta \cdot m]_q, [a \cdot u + e_2]_q)$$

– `Dec` : Pour déchiffrer, calculer

$$\begin{aligned}
[\lfloor \frac{t \cdot [c_0 + c_1 \cdot sk]_q}{q} \rceil]_t &= [\lfloor \frac{t \cdot [[b \cdot u + e_1 + \Delta \cdot m]_q + [a \cdot u + e_2]_q \cdot sk]_q}{q} \rceil]_t \\
&= [\lfloor \frac{t \cdot [b \cdot u + e_1 + \Delta \cdot m + a \cdot u \cdot sk]_q}{q} \rceil]_t \\
&= [\lfloor \frac{t \cdot [\lfloor \frac{q}{t} \rfloor \cdot m + e \cdot u + e_1 + e_2 \cdot sk]_q}{q} \rceil]_t \\
&= [\lfloor m + \tilde{e} \rceil]_t \\
&= m
\end{aligned}$$

Pour certains e et $\tilde{e} = \frac{e \cdot u + e_1 + e_2 \cdot sk}{q}$, on retrouve bien m tant que l'erreur $\tilde{e}$ n'est pas trop grande.

Dans la suite, nous adopterons la notation suivante : $[c(sk)]_q = [c_0 + c_1 \cdot sk]_q = \Delta \cdot m + v$, où $v = e \cdot u + e_1 + e_2 \cdot s$. Étant donnés deux chiffrés c et c' de m et m' sous sk, on obtient que

$$[c(sk) + c'(sk)]_q = \Delta \cdot [m + m']_t + v_1 + v_2 - \epsilon \cdot t \cdot r$$

où $\epsilon = \frac{q}{t} - \lfloor \frac{q}{t} \rfloor < 1$ et $m + m' = [m + m']_t + t \cdot r$ avec $||r|| \leq 1$. L'algorithme d'addition `Add` consiste donc à ajouter membre à membre les chiffrés : $c_add = ([c_0 + c'_0]_q, [c_1 + c'_1]_q)$.

D'autre part, la multiplication n'est pas triviale, mais elle peut être réalisée en deux étapes. Par soucis de clarté, nous modifions légèrement les notations on écrira $c(sk) = \Delta \cdot m + v + q \cdot r$ où v et r sont obtenus grçace à l'équation $[c_0 + c_1 \cdot sk]_q = \Delta \cdot m + v$. Dans la première, on multiplie les deux polynômes $c_1(sk)$ et $c_2(sk)$ avant de multiplier par $\frac{t}{q}$:

$$\begin{aligned}
(c_1 \cdot c_2)(sk) &= (\Delta \cdot m_1 + v_1 + q \cdot r_1)(\Delta \cdot m_2 + v_2 + q \cdot r_2) \\
&= \Delta^2 m_1 m_2 + \Delta(m_1 v_2 + m_2 v_1) + q(v_1 r_2 + v_2 r_1) \\
&\quad + v_1 v_2 + q\Delta(m_1 r_2 + m_2 r_1) + q^2 r_1 r_2
\end{aligned}$$

Pour obtenir un chiffré valable de $m_1 \cdot m_2$, on doit diviser cette expression par Δ. Cependant, $\Delta = \lfloor \frac{q}{t} \rfloor$ ne divise pas nécessairement q, et on serait obliger d'arrondir,

16. Dans leur article, Fan et Vercauteren prennent $t = 2$

ce pendant l'erreur générée en arrondissant le dernier terme $q^2 r_1 r_2$ serait potentiellement grande. Ce pourquoi, nous ne diviserons pas par Δ, mais par $\frac{q}{t}$. En écrivant $(c_1 \cdot c_2)(x) = \tilde{c_0} + \tilde{c_1} \cdot x + \tilde{c_2} \cdot x^2$, on a

$$\frac{t}{q} \cdot (c_1 \cdot c_2)(sk) = \lfloor \frac{\tilde{c_0}}{q} \rceil + \lfloor \frac{\tilde{c_1}}{q} \rceil \cdot sk + \lfloor \frac{\tilde{c_2}}{q} \rceil \cdot sk^2 + r_a$$

où r_a est l'erreur due à l'approximation, dont la taille est majorée par $(\delta_R \cdot ||sk|| + 1)^2/2$, où $\delta_R = \max\{\frac{||a \cdot b||_\infty}{||a||_\infty \cdot ||b||_\infty} | a, b \in R\}$ est le facteur d'expansion de l'anneau R.

Le problème de la première étape de la multiplication est qu'on obtient un chiffré qui n'est plus sous forme initiale $c = (c_0, c_1)$ mais sous forme étendue $\tilde{c} = (\tilde{c_0}, \tilde{c_1}, \tilde{c_2})$. La deuxième étape de la multiplication consiste donc naturellement en une phase de relinéarisation.

Relinearisation : 1ère version

La première façon de faire est de généraliser le procédé de relinéarisation de [12], en décomposant $\tilde{c_2}$ en base T (où $T = 2$ dans les schémas précédant utilisant la relinéarisation) : $\tilde{c_2} = \sum_{i=0}^{\lceil \log_T q \rceil} T^i \cdot \tilde{c_2}^{(i)} \ mod\ q$. Les clés d'évaluation, ou encore de relinéarisation rlk seront les $T^i sk^2$ auxquels on ajoutera du bruit :

$$rlk = \{([b_i + T^i sk^2]_q, a_i), 0 \leq i \leq \lceil \log_T q \rceil\}$$

où les b_i sont obtenus de façon analogue à `KeyGen` : $b_i = a_i sk + e_i$, pour $a_i \leftarrow R_q$ uniformément et $e_i \leftarrow \chi$.
On obtient alors comme procédé de multiplication $c_{mul} = (c_0, c_1)$ avec

$$c_0 = [\tilde{c_0} + \sum_{i=0}^{\lceil \log_T q \rceil} rlk_{i,0} \cdot \tilde{c_2}^{(i)}]_q, \ et\ c_1 = [\tilde{c_1} + \sum_{i=0}^{\lceil \log_T q \rceil} rlk_{i,1} \cdot \tilde{c_2}^{(i)}]_q$$

Relinearisation : 2ème version

La seconde manière d'accomplir cette relinéarisation ressemble à un changement de module. En masquant sk^2 avec une erreur e, lorsqu'on multiplie $sk^2 + e$ par $\tilde{c_2}$ cette erreur devient beaucoup plus grande. La façon de traiter ce problème consiste à donner une version masquée modulo $p \cdot q$, et un chiffré de p, de sorte que pour retrouver une bonne approximation de $\tilde{c_2} sk^2$, il suffira de de diviser par p :

$$rlk = ([b + p \cdot s^2]_{p \cdot q}, a)$$

où $b = a \cdot sk + e$ de façon analogue à `KeyGen`, mise à part que $a \leftarrow R_{pq}$ et $e \leftarrow \chi'$ [17].
Pour obtenir une bonne approximation de $\tilde{c_2} sk^2$, il suffit alors de calculer

$$(\tilde{c_{2,0}}, \tilde{c_{2,1}}) = ([\lfloor \frac{\tilde{c_2} \cdot rlk_0}{p} \rceil]_q, [\lfloor \frac{\tilde{c_2} \cdot rlk_1}{p} \rceil]_q)$$

On obtient alors $\tilde{c_{2,0}} + \tilde{c_{2,1}} \cdot sk = \tilde{c_2} sk^2 + r$ avec r de taille raisonnablement petite.
On obtient alors comme procédé de multiplication $c_{mul} = (\tilde{c_0} + \tilde{c_{2,0}}, \tilde{c_1} + \tilde{c_{2,1}})$ où $(\tilde{c_{2,0}}, \tilde{c_{2,1}})$

17. Le choix de cette distribution doit être fait rigoureusement afin de ne pas affecter la sécurité du système. Voir [19] pour plus de détails.

6.5 Synthèse et comparaison

Dans cette section, nous reprenons les cryptosystèmes complètement homomorphes décrits ci-dessus et analysons leur complexité. Le tableau ci-dessous permet d'avoir un aperçu des performances de chacun d'eux.

Taille	clé publique	clé secrète	chiffrés
[12]	$O(n^2 \log^2 q)$	$n \cdot \log q$	$(n+1) \cdot \log q$
[11]	$2dn \log q$	$2d \log q$	$2d \log q$
[10]	$O(n^2 \log^2 q)$	$n \log q$	$(n+1) \log q$
[19]	$2d \log q$	d	$2d \log q$

Quatrième partie
Réalisations, implémentations et cas d'usage

7 Implémentations des cryptosystèmes simplement homomorphes

7.1 Langage de programmation

Java, le langage de programmation de l'entreprise
Au cours de mon stage, j'ai implémenté les systèmes de chiffrement homomorphes en Java. Il s'agît d'un langage de programmation orienté objet, fortement typé, qui supporte le polymorphisme. Bien que nous ne définirons pas formellement ces propriétés, nous allons les expliquer en quelques mots.
Programmation orientée objet : cela consiste à décomposer chaque tâche en briques élémentaires, implémenter ces briques, et les faire intéragir.
[Fort] typage : le typage définit la syntaxe et la sémantique de l'objet, c'est-à-dire qu'il caractérise ce que l'objet est sensé réalisé. Java est fortement typé, c'est-à-dire que les conversions implicites de type sont à proscrire. Un entier de type `int` stocké usuellement sur 32 bits sera différent d'un entier de type `short` stocké usuellement sur 16 bits.
Polymorphisme : un même objet peut appartenir à plusieurs types, par exemple un minotaure est à la fois du type humain et du type taureau.

Bien que je n'ai pas beaucoup utilisé ce langage de programmation au cours de ma formation, il reste très abordable, et je n'ai pas eu de mal à m'y adapter. Cependant, derrière ce langage, il a ce qu'on appelle couramment le paradigme objet, qui a mon sens est beaucoup plus difficile à saisir. Ce paradigme dicte la façon dont on doit s'attaquer aux problèmes, en évitant la manière séquentielle à laquelle j'étais très habitué. Je pense que cette phase d'adaptation a été une des plus difficiles du stage : changer de point de vue, de façon de programmer.

Si les langages de programmation auxquels j'étais habitué adoptent une approche séquentielle, comme le C typiquement, j'ai dû apprendre à estimer la pertinence des méthodes ou objets créés au sein du programme global. Une autre difficulté que j'ai rencontrée est la généricité. J'ai dû regrouper des cryptosystèmes tels que Paillier et ElGamal qui ne possèdent pas les mêmes propriétés homomorphes dans une classe mère afin de pouvoir utiliser un cryptosystème ou un autre indifféremment dans les autres classes. Cette façon de coder présente l'avantage majeur de ne pas avoir à implémenter une méthode par cryptosystème, mais une méthode pour tous les cryptosystèmes.

7.2 Java : un langage de programmation clé en main

Bien que se familiariser avec le paradigme objet m'ait pris beaucoup de temps, je dois reconnaître qu'il est vraiment facile d'implémenter des choses complexes à partir des briques de bases que fournies en Java. Prenons un exemple parlant : l'arithmétique multiprécision. En C, la librairie GNU Multi Precision (ou GMP) permet de gérer des entiers de taille arbitraire, mais n'est pas vraiment accessible aux novices. En Java, la classe `BigInteger` permet de gérer l'arithmétique multiprécision avec des méthodes relativement simples. Par exemple, si `a` et `b` du type BigInteger, on peut facilement obtenir leur somme (`a.add(b)`), le produit (`a.multiply(b)`), et bien d'autres choses encore.

Lorsque j'ai commencé à implémenter les cryptosystèmes dont nous avons parlé section 5, je ne connaissais pas encore la librairie BouncyCastle, que nous détaillerons section 8.1.1. Je les ai donc implémenter du point de vue mathématiques, et séquentielle, tout le contraire de ce qui aurait dû être fait d'un point de vue objet.

7.3 Maven

Maven est un puissant outil de automatisation de construction, propre à Java, qui utilise le XML, langage auquel j'ai du me familiariser. La façon dont j'ai eu à l'utiliser n'est certainement pas optimale, mais m'a bien aidé à gérer les dépendances entre les projets, ainsi que les versions à utiliser.

FIGURE 21 – Le logo de Maven

Maven m'a entre autre permis d'intégrer BouncyCastle à mes secondes implémentations des cryptosystèmes décrits plus haut. Afin d'adopter une approche objet, nous décrivons dans la section qui suit les packages de base de la librairie cryptographique de BouncyCastle.

8 Étude des librairies et Réalisations

8.1 Étude des librairies

8.1.1 BouncyCastle

Bouncy Castle est une librairie d'outils cryptographiques libre et open-source pour Java (et C#), comparable à OpenSSL en C.

FIGURE 22 – Les logos de Bouncy Castle et Bouncy Castle pour Java

Concernant les systèmes de chiffrement à clés publique, Bouncy Castle possède déjà le package `org.bouncycastle.crypto` dans lequel nous étudierons les trois packages : `params`, `generators`, et `engine`, auxquels j'ai dû m'adapter lorque j'ai implémenter les cryptosystèmes, afin d'être compatible avec Bouncy Castle. Nous allons décrire de façon générique ces trois packages en implémentant le cryptosystème `CryptoSystem`, qui possède comme opération homomorphe `Operation`.

`params`
En tout premier lieu, le cryptosystème devra posséder une classe `CryptoSystemKeyPairGeneratorParameters` qui étendra la classe `KeyGenerationParameters` présente dans Bouncy Castle. Cette classe a pour objectif d'initier un random, et choisir le niveau de sécurité désiré. Une classe `CryptoSystemParameters` devra également être créée. Elle possèdera comme attributs tous les paramètres du cryptosystème [18]. La classe `CryptoSystemKeyParameters` étend la classe `AsymmetricKeyParameter` qui implémente elle même l'interface *`CipherParameters`*. Les classes `CryptoSystemPublicKeyParameter`, `CryptoSystemSecretKeyParameter`, et `CryptoSystemOperationKeyParameter` étendent toutes les trois la classe précédente `CryptoSystemKeyParameters`, et sont utilisées toutes les trois pour respectivement chiffrer, déchiffrer, ou réaliser une opération homomorphe.

`generators`
La classe `CryptoSystemKeyPairGeneratorParametersGenerator` possède comme

18. Dans le cas où `CryptoSystem` est `RSA`, cette classe contiendra $p, q, N, \ldots$

attributs le random et le niveau de sécurité désiré plus éventuellement des informations propres au cryptosystème [19] et une méthode publique `generate()` qui retourne une instance de `CryptoSystemKeyPairGeneratorParameters` initiée grâce aux attributs. La classe `CryptoSystemKeyPairGenerator` qui implémente l'interface `AsymmetricCipherKeyPairGenerator` possède deux méthodes publiques : `init(KeyGenerationParameters param)` qui initialise le générateur clés, et `generateKeyPair()`, qui retourne une `AsymmetricCipherKeyPair`.

`engine`
Enfin, nous devrons utiliser la classe `CryptoSystemEngine` qui implémente l'interface *`AsymmetricBlockCipher`*. Elle possède comme attributs un `CipherParameters`, et deux `int` représentant les tailles de blocs d'entrées et de sorties. Les méthodes à implémenter sont `getInputBlockSize()` et `getOutputBlockSize()` qui ne font que retourner respectivement les tailles d'entrée et de sortie, `init(boolean forEncryption, CipherParameters param)` qui initialise le processus [20], et enfin `processBlock(byte[] in, int inOff, int len)` qui prend le tableau d'octets in, et applique le processus sur les octets de `inOff` à `len-1`.

D'autre part, lorsque nous avons utilisé les implémentations de [12, 11], nous avons choisi la librairie JLBC open source. En étudiant cette librairie, je me suis rendu compte que les packages `params`, `generators`, et `engine` constituaient un standard, dont j'ai dû tenir compte lorsque j'ai réécris les cryptosystèmes.

8.1.2 JLBC

JLBC, pour Java Lattice Based Cryptography est une librairie de calcul permettant d'utiliser les réseaux euclidiens. La librairie JLBC est composée de deux modules principaux : `jlbc-api`, qui contient les interfaces de programmation, et `jlbc-plaf` qui les implémente. Ces deux modules permettent déjà d'utiliser les réseaux euclidiens. Le dernier module `jlbc-crypto` contient les implémentations de [12, 11], conformément à BouncyCastle. Le lecteur pourra disposer de plus ample informations à l'adresse `http://gas.dia.unisa.it/projects/jlbc/`.

8.2 Réalisations

Au cours de ce stage, j'ai eu l'occasion de m'intéresser à certaines démonstrations utilisées dans [24, 12, 11, 56]. Ceci m'a permis de comprendre les mécanismes qui entrent en jeu lors de certaines astuces d'optimisation. D'autre part, ceci m'a permis de réaliser que certaines de ces optimisations sont suffisamment générique pour être transportées à d'autres cryptosystèmes.

À titre d'exemple, la technique de `batching` introduite par [11], peut a priori être appliquée au cryptosystème de [12, 19]. Cette constatation nous est apparue lorsque

19. Comme la taille r des clairs pour Benaloh
20. Si `forEncryption` est vrai, le processus consistera en un chiffrement, un déchiffrement sinon

nous avons dresser un état de l'art de la façon de créer un système de chiffrement complètement homomorphe, et pourra être d'avantage étudiée lors de la thèse.

D'autre part, afin de mieux comprendre comment paramétriser les cryptosystèmes de sorte à atteindre le niveau de sécurité 2^{λ} attendu, j'ai dû m'intéresser aux réductions existantes entre les différents problèmes. Ces réductions étant en temps polynomial, jusqu'à un facteur d'approximation polynomial, je n'ai pas été en mesure de déterminer de manière précise la manière précise de déterminer les paramètres, je reviendrai donc sur ce point durant la thèse.

Dans la section suivante, nous décrivons les cas d'usage que nous avons retenu. Le premier utilise un système de chiffrement simplement homomorphe, tandis que le second utilise un somewhat.

9 Cas d'usage mis en avant à l'aide des cryptosytèmes homomorphes

9.1 Cas d'usage possibles

Les systèmes de chiffrement complètement homomorphes permettent d'évaluer des circuits, de profondeur arbitraire, de manière sécurisée. Face à cette observation, nous nous sommes demandés quelles pourraient être les applications pratiques de ce résultat théorique, et avons abouti au schéma suivant.

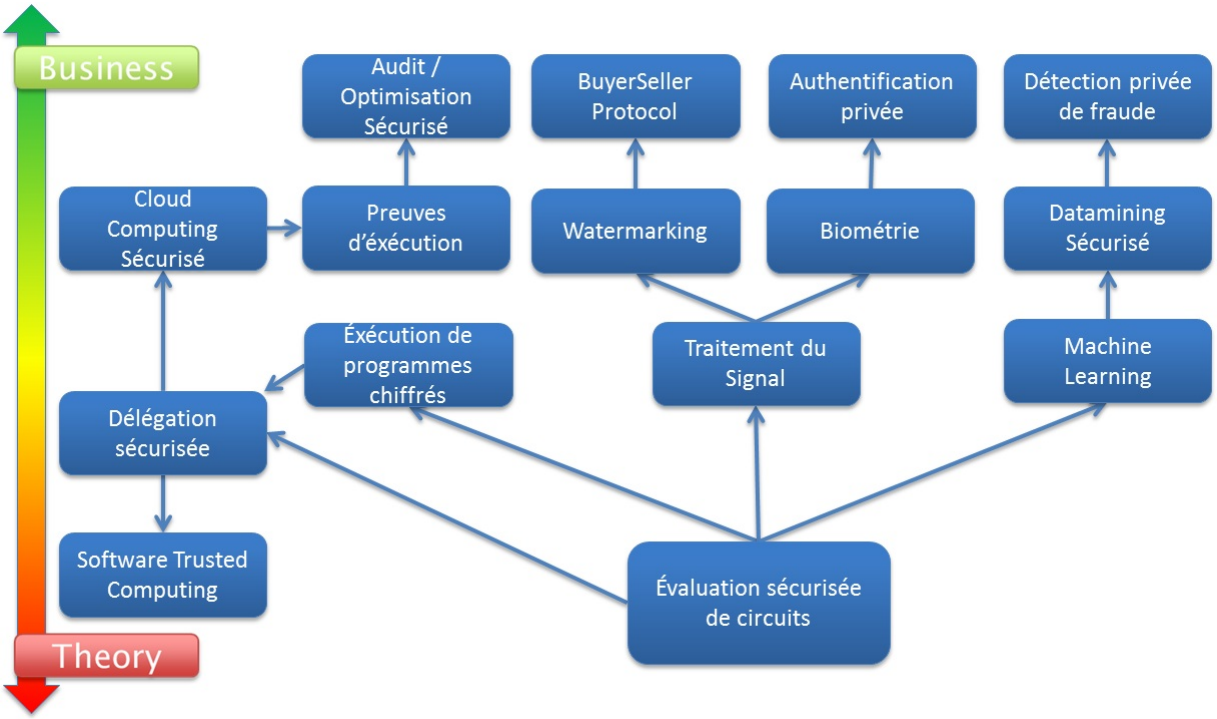


FIGURE 23 – Panel des cas d'usage du chiffrement complètement homomorphe

9.2 Exécution d'un programme sur des données chiffrées

Dans cette section, nous nous intéressons aux cas d'usage qui pourraient profiter à Atos Worldline. Le premier est assez conceptuel et nécessite une grande quantité de travail en amont du chiffrement homomorphe. Supposons qu'Atos Worldline travaille dûr pour résoudre un problème difficile, comme celui du voyageur de commerce. Dans ce problème, un commercial doit parcourir un ensemble de villes et revenir à sa ville de départ en minimisant la distance parcourue. Ce problème consiste à trouver le plus court cycle hamiltonien dans un graphe complet [21] généralement non-orienté.

Atos Worldline aura certainement investit une somme considérable et n'aura pas envie de disséminer des copies de son logiciel résolvant ce problème. D'un autre côté,

21. C'est-à-dire que tous les sommets du graphe sont reliés entre eux

les potentiels clients d'Atos Worldline, qui pourraient être des sociétés de transport ou de courrier, ont probablement des donnés confidentiels qu'ils ne souhaitent pas divulguer. Le chiffrement complètement homomorphe apparaît comme la solution appropriée face à ce problème.

En effet, supposons que la société cliente chiffre ses données confidentielles et les envoie à Atos, Atos peut alors exécuter son logiciel en interne sur les données chiffrées de son client, et renvoyer le résultat chiffré à son client, moyennant paiement, qui n'a alors plus qu'à déchiffrer. La confidentialité des données est préservée, Atos n'a rien appris des données du client, et le ce dernier a obtenu la solution à son problème.

9.3 Watermarking

Au cours de mon stage, je me suis également intéressé au watermarking utilisant le chiffrement homomorphe. Le watermarking est un domaine du traitement du signal où on cherche à protéger un contenu multimédia en y insérant une watermark de sorte que l'acheteur ne puisse la retirer ou la modifier sans détériorer de façon considérable le contenu. Le watermarking est utilisé dans le protocole du Buyer-Seller que j'ai implémenté. J'ai utilisé le protocole proposé par Memon et Wah Wong [37]. Sous réserve que toutes les parties soient honnêtes, les garanties couvertes par le watermarking sont :

- `Exactitude` : Le protocole se termine correctement.
- `Traçabilité` : Lorsqu'une copie piratée est trouvée, le vendeur est capable d'identifier l'acheteur qui l'a diffusé.
- `Non-forgeabilité` : Un acheteur honnête ne peut être accusé d'avoir diffusé des copies piratées du contenu qu'il a acheté.
- `Non-répudiation directe` : Un acheteur coupable ne peut renier avoir diffusé des copies piratées du contenu qu'il a acheté. De plus, le vendeur peut convaincre une tierce personne de la culpabilité de l'acheteur sans intéragire avec ce dernier.
- `Respect de la vie privée` : Le vendeur n'apprend rien sur le contenu acheté.
- `Anti-fraude` : Le vendeur est sûr que l'acheteur paye le montant correct.

Cependant, la majorité des protocoles de watermarking homomorphes, [37] y compris s'avèrent lents et limités. Ces défauts sont inhérents au fait que dans ces protocoles, la watermark est inséré dans le contenu de manière homomorphe, cela signifie que la watermark est chiffrée, mais que le contenu doit l'être aussi, et c'est exactement là la source des problèmes. La taille des fichiers à watermarker est limitée par la taille des clés, et la watermark est insérée par l'intermédiaire d'un Xor (dans [37]), à l'aide du cryptosystème de Goldwasser et Micali. Ce qui signifie que l'image sera chiffrée bit par bit, ce prend un certain temps, mais permet de contrer le premier défaut.

Pour une petite image de taille 300×300 pixels au format JPEG, le fichier qui en résulte est stocké sur environ 40 kilo octets soit 320000 bits à chiffrer. Si cela peut éventuellement être accepté en tolérant que cette opération prenne quelques

minutes, il est cependant inimaginable de watermarker un film de quelques giga octets avec cette technique !

Pour conclure, ce protocole fonctionne, mais insérer une watermark juste avec un Xor ne semble pas vraiment une technique robuste. La récente arrivée du chiffrement complètement homomorphe incitera peut-être les personnes de ce domaine à créer des protocoles de watermarking homomorphes plus complexes, notamment avec des ondelettes, des transformées de Fourier et/ou des transformées en cosinus discrètes.

9.4 Placement sous Surveillance Électronique Mobile

Dans le contexte d'une exécution de peine, il arrive que certains condamnés aient la possibilité d'effectuer une partie, voire la totalité de leur peine sous surveillance électronique.

9.4.1 Description du système [25] du Ministère de la Justice

Les sociétés sollicitées proposent des matériels sensiblement différents qui reposent néanmoins sur un système identique.

Les solutions présentées font appel au réseau satellitaire GPS (" global positioning system "), fondé et contrôlé par le Ministère de la Défense des États-Unis. Le système GPS permet de localiser tout individu porteur de l'équipement nécessaire avec une précision d'environ 10 mètres. Pour des raisons stratégiques, le Pentagone est seul bénéficiaire d'une précision supérieure pour des applications militaires.

Dans certains cas, les ondes GSM (" global service mobil ") seront utilisées. Les ondes GPS ayant un rayonnement plus faible que les ondes GSM, dans les situations où les premières ne pourront plus être captées (à l'intérieur d'un bâtiment, en souterrain), les secondes seront utilisées à titre de relais.

Dans cette hypothèse, elles offrent toutefois un degré de précision moindre (50 m en zone urbaine, 500 m à 1 km en zone rurale).

Le principe utilisé pour la localisation est celui d'un double mode de positionnement GPS et GSM fondé sur le principe de la triangulation des signaux émis par les antennes des satellites pour le réseau GPS et/ou des antennes téléphoniques pour le réseau GSM.

Le récepteur GPS-GSM calcule la distance qui le sépare de satellites ou d'antennes en se basant sur le temps de transmission des signaux. La localisation est ensuite calculée à partir de la distance d'éloignement de trois satellites. Un quatrième satellite permet éventuellement de déterminer l'altitude.

Ces informations sont ensuite transmises à un logiciel de surveillance par le réseau de téléphonie mobile GSM, soit par son fonctionnement classique avec facturation du coût de la communication, soit par le mode GPRS avec facturation du coût, plus économique, de la quantité d'informations transmise.

Le logiciel de surveillance, géré par un prestataire de service privé [22], intègre un fichier nominatif des personnes placées sous surveillance électronique mobile. Il permet de déterminer et de contrôler tous les paramètres du programme de surveillance propre à chaque placé.

Ce programme peut comprendre des horaires d'assignation à domicile (zones d'inclusion) et des zones d'exclusion associant éventuellement des horaires pour ces exclusions. Le logiciel permet de surveiller jusqu'à 50 zones d'exclusion. Ces zones d'exclusion, qui sont définies par l'autorité judiciaire, en fonction des faits commis par le condamné, peuvent par exemple concerner des lieux accueillant des enfants (écoles, centres de loisirs...), des lieux fréquentés par la victime (domicile, lieu de travail, centres commerciaux,...), des lieux sensibles en matière de délinquance (trafic de stupéfiants, prostitution...).

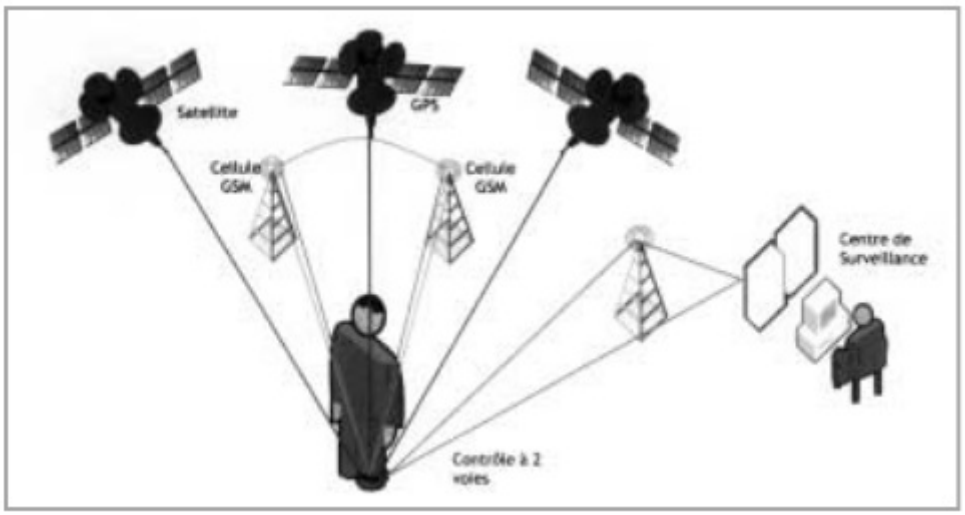


FIGURE 24 – Fonctionnement du PSEM

Les solutions techniques proposées par les différents prestataires se distinguent quant à la composition du matériel de surveillance.

22. Atos Worldline pourrait être celui-ci...

FIGURE 25 – Les différents appareils disponibles sur le marché : de gauche à droite : un boîtier récepteur portable de PSEM, une montre, un bracelet PSE statique et un bracelet de PSEM

Le premier système (Elmo Tech et Premier Geografix), représenté ci-dessus est composé de 3 éléments :
– Un bracelet émetteur à porter à la cheville ou au poignet
– Un boîtier récepteur portable
– Un boîtier récepteur statique
Le deuxième système disponible (On Guard) est composé d'un seul appareil, comprenant à la fois l'émetteur, le récepteur et une batterie.

Le lecteur pourra trouver de plus amples informations dans le rapport [25] du Ministère de la Justice, nous expliquons notre contribution au système.

9.4.2 Contribution dans le système de Placement sous Surveillance Électronique Mobile

Notre idée a été de fournir aux prisonniers une garantie du respect de leur vie privée tout en utilisant un cryptosystème homomorphe dont les performances sont acceptables. Dans cet objectif, nous avons étudié les différents outils cryptographiques à notre disposition, et avons décidé d'utiliser conjointement un système de signature de groupe et un système de chiffremment somewhat homomorphic. Dans la suite, nous commençons par décrire en quoi consiste une signature de groupe, ainsi que le modèle de sécurité associé, puis nous décrivons notre "Proof of Concept", qui implique un établissement pénitentiaire, un prestataire de service [23], et des prisonniers.

23. Qui jouera le rôle de manager du groupe

9.4.3 Signature de groupe

Le concept de signature de groupe a été introduit par David Chaum et Eugène Van Heyst en 1991 [14]. Il s'agît d'une architecture cryptographique assurant l'anonymat d'un signataire au sein d'un groupe. Plus précisément, un membre peut prouver à un vérificateur son appartenance à un groupe sans que ce dernier ne puisse identifier de façon unique le signataire. Un mécanisme exécutable uniquement par un tiers de confiance (le GroupManager) permet néanmoins de révoquer l'anonymat et de révéler l'identité du membre. Les entités impliquées dans un schéma de signature de groupe peuvent donc être séparées en trois catégories :

- `Manageur du groupe` : il s'agît de l'entité la plus importante dans l'architecture. Il est responsable de la délivrance et de la révocation des certificats de membre. Il est également le seul à pouvoir "ouvrir" les signatures produites par les membres et ainsi désanonymiser le signataire.
- `Membres du groupe` : ils peuvent produire des signatures valides à partir d'une clé cryptographique secrète et de leur certificat d'appartenance au groupe après s'être enregistré auprès du GroupManager.
- `Vérificateur` : est une entité responsable de vérifier la validité d'une signature de groupe. Plus précisément, il peut s'assurer qu'elle a effectivement été produite par un membre légitime, grâce à la clé publique du groupe.

En terme de modèle de sécurité, un schéma de signature de groupe doit avoir les propriétés suivantes :

- `Consistance et solidité` : Une signature légitime doit toujours passer la procédure de vérification, tandis qu'une signature falsifiée échoue avec une grande probabilité.
- `Anonymat` : Il est impossible d'identifier le membre qui a produit une signature à partir d'un message signé (sauf pour le GroupManager préposé à l'ouverture des signatures).
- `Non-chaînabilité` : Il est impossible de décider si deux messages signés différents proviennent du même signataire.
- `Traçabilité totale` : Cette propriété regroupe plusieurs notions
  - `Non-forgeabilité` : Seuls les membres du groupe peuvent produire des signatures valides.
  - `Traçabilité` : L'entité de confiance est toujours capable d'ouvrir une signature valide et d'identifier le signataire.
  - `Disculpabilité` : Ni un membre du groupe ni le GroupManager ne peut signer un message au nom d'un autre membre.
  - `Résistance aux coalitions` : Une coalition d'un sous-ensemble des membres du groupe ne peut générer une signature valide que le GroupManager ne peut lier à un membre.

Nous avons utilisé l'implémentation java du schéma de signature de David Schönfeld qui est décrite dans [5]. Nous l'avons légèrement modifié de sorte à l'adapter à notre protocole.

9.4.4 Choix du cryptosystème homomorphe

Dans cette section, nous discutons du choix du cryptosystème que nous avons finalement retenu pour notre protocole. Ce choix a notamment été fortement impacté par deux problèmes :

1. Comme spécifier ci-dessus, le cryptosystème devra avoir des performances acceptables. C'est-à-dire que les procédures impliquant le cryptosystème ne devront pas représenter la majeure partie des calculs effectués dans notre protocole. Étant donné les performances des systèmes de chiffrement complètement homomorphes actuels, il devient alors hors de question d'y recourir...
2. D'autre part, le premier problème en entraine implicitement un second : si on ne peut pas utiliser de cryptosystème complètement homomorphe, il est nécessaire de savoir quels types de fonctions nous devrons évaluer. Par "type de fonction", nous entendons linéaire, quadratique et autres.

La résolution du second problème est prioritaire par rapport au premier. Si les fonctions que nous devons évaluer ont un degré trop élevé, certains cryptosystèmes seront également à proscrire. Nous avons essayé de créer un nouveau cryptosystème complètement homomorphe à partir de cryptosystèmes simplement homomorphes, comme Paillier qui permet de faire des additions et El Gamal qui permet de faire des multiplications, cependant cela n'a pas été fructueux.

Avant de savoir quelles fonctions nous devrons évaluer sur les données chiffrées, nous devons définir quelles informations doivent être protégées. Dans l'optique de garantir le respect de la vie privée des prisonniers, nous souhaitons, outre leur garantir de l'anonymat via la signature de groupe, nous souhaitons également que l'établissement pénitentiaire ne connaissent pas la position exacte de ses prisonniers. Cela peut paraître paradoxal lorsqu'on parle de placement sous surveillance électronique, mais cela est possible par l'intermédiaire du prestataire de services que pourrait être Atos Worldline. Par mesure de sécurité, ce dernier conservera tout de même de manière chiffrée un historique des déplacements des prisonniers, qui pourra être révélé en cas de problème.

Afin de définir les déplacements des prisonniers, nous avons dû réfléchir à la façon de modéliser les zones autorisées, et celles interdites. Le problème était de définir suffisamment simplement ces zones afin que la fonction qui permet de vérifier si un prisonnier est dans une zone autorisée ou non soit relativement simple. Ses observations nous ont amenés à définir des zones circulaires : un prisonnier possède un bracelet, qui calcule la position (x, y) du prisonnier. L'ensemble des zones autorisées du prisonnier est un ensemble de $n \leq 50$ [24] cercles centrés en $(x_{0,i}, y_{0,i})$, de rayon r_i. Ainsi, vérifier si un prisonnier est dans une zone règlementaire revient à déterminer si

$$(x_{0,i} - x)^2 + (y_{0,i} - y)^2 \leq r_i^2, 1 \leq i \leq n.$$

24. De sorte à respecter les restrictions du logiciel [25]

Pour chaque zone, le bracelet envoie un chiffré randomisé de (x, y) à l'établissement pénitentiaire qui vérifie la signature et calcule $(x_{0,i}-x)^2+(y_{0,i}-y)^2-r_i^2$ de manière homomorphe, et fais appelle au prestataire de service - qui est également le Group-Manager - qui déchiffre et vérifie que le résultat est négatif ou nul. Si ce n'est pas le cas, le prestataire retourne un message d'alerte au centre carcéral, avec l'identité du/des prisonnier(s) sortis de leur zone, avec l'historique de leurs déplacements déchiffré.

Ceci étant fixé, les fonctions que nous auront à évaluer seront donc au plus quadratiques. Le cryptosystème de Boneh, Goh, Nissim semble alors idéalement adapté à nos besoins. Cependant, la discussion ci-dessus n'est valable que pour un prisonnier. En effet, le choix que le calcul $(x_{0,i}-x)^2+(y_{0,i}-y)^2-r_i^2$ soit réalisé côté prison a été fait dans le but d'optimiser la gestion et la mise à jour des zones autorisées. Cependant, si un prisonnier envoie seulement sa position chiffrée, la prison ne saura pas quelle(s) zone(s) choisir pour effectuer le calcul, et si le prisonnier envoie le numéro de sa/ses zone(s), il n'est plus anonyme.

Face à ce problème, nous avons décidé d'indexer les prisonniers et d'utiliser un filtre, qui sera un vecteur de taille le nombre total de prisonnier, nul partout sauf en l'indice du prisonnier et qui sera multiplié coordonnée par coordonnée aux zones autorisés afin de sélectionner automatiquement la/les zones , qui présente l'avantage de résoudre ce problème, mais suscite deux inconvénients :

1. Le coût des communications est multiplié par le nombre total de prisonnier sous surveillance.
2. Les fonctions à évaluer deviennent cubiques.

Le second problème devient très gênant, car même le cryptosystème de Boneh, Goh, et Nissim ne peut évaluer qu'une seule multiplication. Nous devons donc abandonner l'idée d'utiliser les cryptosystèmes simplement homomorphes. Cependant nous ne pouvons pas non plus utiliser du complètement homomorphe, qui serait trop lent pour notre application. Notre choix s'est donc tourné vers du somewhat homomorphic, rapide, et pouvant évaluer quelques multiplications. Nous aurions également pu choisir un cryptosystème "leveled fully homomorphic", d'ailleurs l'implémentation que j'ai réalisée est suffisamment générique pour supporter ce changement.

9.4.5 Description du protocole

À chaque fois que l'établissement pénitentiaire place un détenu sous surveillance électronique, le bracelet est ajouté au groupe. Puis à intervalle de temps régulier, le bracelet envoie la position chiffrée du prisonnier au centre carcéral, qui effectue son calcul dans les données chiffrés et envoie le tout au prestataire pour déchiffrement et validation.

Dans la section précédente, nous avons abordé le fait que le prisonnier devait appliquer un filtre, nous allons décrire précisément comment est créé ce filtre, qui consistera en un vecteur de n coordonnées. Chacun des n prisonniers se voit affecter un numéro $i \in \{1, \ldots, n\}$. Le bracelet connaît le nombre total de prisonniers n qui est mis à jour à chaque fois qu'un nouveau détenu est placé sous surveillance. Pour le prisonnier i, pour $1 \leq j \leq n$ si $j = i$ alors $v_j = v_i = Enc_{pk}(1)$ sinon, $v_j = Enc_{pk}(0)$. Ici, pk désigne la clé publique du GroupManager. Lorsque toutes les coordonnées ont été chiffrées, le bracelet chiffre la position du prisonnier et l'envoie à la prison, qui effectue ses calculs sur toutes les zones autorisées dont elle dispose, et les envoie au GroupManager qui effectue la vérification et répond à la prison de façon adéquate.

Afin de réaliser son calcul, pour chaque zone autorisée, la prison chiffre les coordonnées du centre de la zone et du rayon : $Enc_{pk}(x_0)$, $Enc_{pk}(y_0)$ et $Enc_{pk}(-r^2)$ [25], calcule de façon homomorphe $Enc_{pk}(x_0 - x)$ et $Enc_{pk}(y_0 - y)$, multiplie ses chiffrés avec eux-mêmes pour obtenir $Enc_{pk}((x_0 - x)^2)$ et $Enc_{pk}((y_0 - y)^2)$ avant de les ajouter ensemble avec $Enc_{pk}(-r^2)$ et obtenir ainsi $Enc_{pk}((x_0 - x)^2 + (y_0 - y)^2 - r^2)$. Si E est une borne supérieure sur la taille de l'erreur, alors la taille de l'erreur contenue dans le chiffré final est bornée par $8e^2 + e$ [26]. Une fois ce chiffré final obtenu, il est envoyé au GroupManager.

Le GroupManager vérifie alors que le prisonnier concerné est bien dans sa zone, en s'assurant que le message déchiffré est négatif, ce qui correspond au cas où

$$(x_0 - x)^2 + (y_0 - y)^2 \leq r^2$$

. Si toutefois ce n'était pas le cas, le prestataire peut lever l'anonymat du prisonnier, avertir le centre carcéral qui demande l'historique des déplacements au bracelet, l'envoie au prestataire qui le déchiffre et le renvoie au centre. Le centre peut alors décider de révoquer ou non le prisonnier, c'est-à-dire l'exclure du groupe.

Par soucis de sécurité, les messages et l'historique sont signés de sorte que même un centre carcéral qui essai de connaître l'identité ou l'historique d'un détenu à la place d'un autre se voit refuser cette demande par le prestataire. Le problème de notre protocole est un problème inhérent à la signature de groupe que nous n'avons pas réussi à contrer : si le prestataire est malhonnête, tout peut se passer...

9.4.6 Optimisations

Comme on utilise un système de chiffrement somewhat homomorphe, il est primordial de chercher à ne pas faire grandir l'erreur trop vite, afin que le déchiffrement continue à être correct. Encore une fois, en remplaçant facilement ce somewhat homomorphic par un leveled homomorphic de 3 niveaux, le déchiffrement sera correct et il sera alors inutile de se soucier de cela.

25. Chiffrer directement $-r^2$ permet d'obtenir moins de bruit qu'en chiffrant r et en calculant un chiffré de $-r^2$ de manière homomorphe

26. Nous présentons deux optimisations légèrement plus loin

Pour rappel, le but est de calculer $Enc_{pk}((x_0 - x)^2 + (y_0 - y)^2 - r^2)$ étant donnés $x, y, r, Enc_{pk}(x_0)$ et $Enc_{pk}(y_0)$. Dans la suite, nous décrivons deux optimisations pour diminuer la borne sur la taille de l'erreur du chiffré final. Cette erreur était $8E^2 + E$ initialement, où E est une borne sur la taille des erreurs initiales.

Optimisation 1
Une première optimisation consisterait à calculer la formule obtenue en ouvrant les parenthèses, c'est-à-dire $Enc_{pk}(x_0^2 - 2xx_0 + x^2 + y_0^2 - 2yy_0 + y^2 - r^2)$, en chiffrant $Enc_{pk}(x_0^2)$, $Enc_{pk}(-2x_0)$, $Enc_{pk}(y_0^2)$, $Enc_{pk}(-2y_0)$, et $Enc_{pk}(-r^2)$. Ainsi la taille de l'erreur du chiffré final devient majorée par $4E^2 + 5E$.

Optimisation 2
La seconde optimisation fait intervenir le bracelet, qui doit en plus d'envoyer des chiffrés de x et y, envoyer également des chiffrés de x^2 et y^2. Le calcul de $Enc_{pk}((x_0 - x)^2 + (y_0 - y)^2 - r^2)$ se fait comme dans la première optimisation, et on arrive à atteindre une erreur finale dont la taille est majorée par $2E^2 + 5E$. Cependant, cette optimisation double la taille des communications.

9.4.7 Problèmes rencontrés

Au cours de l'élaboration et l'implémentation de ce protocole, le premier problème rencontré a été le choix du cryptosystème. En effet, comme discuté ci-dessus, l'apparition de fonctions de degré trois nous a poussé à utiliser un cryptosystème qui soit au minimum somewhat homomorphic.

D'autre part, nous avons également évoqué deux optimisations, cependant, par manque de temps, nous n'avons pu implémenter que la version de base, où afin de diminuer la taille de l'erreur présente dans le chiffré, le GroupManager aide pour en calculant les multiplications présentes dans la formule, en déchiffrant. Dans notre modèle, il est absolument nécessaire que le GroupManager soit honnête, ce pourquoi lui confier cette tâche n'entrave pas la sécurité du système. D'autre part, cette tâche peut être facilement et simplement réaffectée au bracelet en implémentant ce protocole avec un système de chiffrement complètement homomorphe nivelé avec $L = 3$.

Nous souhaitions également réaliser une implémentation Android de ce protocole, dans laquelle un smartphone aurait joué le rôle du bracelet électronique, et la prison aurait été représentée par un serveur. Cependant, encore par manque de temps, nous n'avons pu le faire.

Conclusions et Perspectives

Dans ce qui précède, nous avons décrit les différentes phases du stage, à commencer par l'état de l'art. Celui-ci a débuté par un résumé de tous les cryptosystèmes simplement homomorphes. En parallèle, j'ai dû les implémenter dans le but de comparer leurs performances respectives. Au cours de cette période, je n'avais pas le temps d'étudier les cryptosystèmes complètement homomorphes, nous avons donc décidé que cela constituerait la prochaine phase. Durant celle-ci, nous avons utilisé la librairie JLBC que j'ai dû préalablement étudier, ce qui m'a amené à m'intéresser aux modules de cryptographie dans Bouncy Castle, et nous avons décider de modifier les implémentations des cryptosystèmes simplement homomorphes, afin d'être compatible avec ces modules.

L'étude des cryptosystèmes complètement homomorphes nous a alors poussé à dresser une liste des problèmes classiques en cryptographie basée sur les réseaux, et de leur réduction. Bien que la liste établie ne soit pas exhaustive, je possède dorénavant une bonne connaissance de ces problèmes. Cependant, je n'arrive pas à comprendre certaines des réductions entre ces problèmes, et je pense que cela constituera un des premiers axes que j'explorerai au cours de la thèse.

J'ai été amené à présenter mes travaux aux cours d'événements internes, ainsi que les potentiels cas d'usage qu'il était possible d'explorer, nous avons alors retenu le placement sous surveillance électronique, et décidé d'en réaliser une preuve de concept. J'ai alors étudié la signature de groupe, dont je connaissais uniquement le concept.

D'un point de vue informatique, ce stage m'aura permis de me perfectionner en développement, notamment en découvrant de nouveaux outils, ainsi que certaines façons de penser son code. J'ai pu réaliser à quel point la modélisation était importante afin d'avoir un code portable et maintenable.

D'un point de vue mathématique, ce stage m'aura permis de développer mes connaissances des réseaux euclidiens et de comprendre la plupart des mécanismes utilisés dans les cryptosystèmes complètement homomorphes. J'ai également appris qu'il existait deux autres algorithmes pour SVP : HKZ et BKZ, que je n'ai pas eu le temps d'étudier en détail. Le premier a été utiliser pour cryptanalyser les challenges publiés à l'adresse `http://www.latticechallenge.org/`, mais je ne sais pas dans quelles mesures est utilisé le second. Ceci entrera également dans le cadre de ma thèse. Nous avons constaté que certains cryptosystèmes comme [19] ne profite pas des optimisations proposées précédemment, comme le batching par exemple. Je réfléchirai donc à la manière d'intégrer ces optimisations dans les cryptosystèmes existants.

Références

[1] Aguilar-Melchor, C., Castagnos, G., and Gaborit, P. Lattice-based homomorphic encryption of vector spaces. In *ISIT* (2008), pp. 1858–1862.

[2] Aguilar-Melchor, C., Gaborit, P., and Herranz, J. Additively homomorphic encryption with d-operand multiplications. *IACR Cryptology ePrint Archive 2008* (2008), 378.

[3] Ajtai, M. The shortest vector problem in l2 is np-hard for randomized reductions (extended abstract). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (New York, NY, USA, 1998), STOC '98, ACM, pp. 10–19.

[4] Armknecht, F., and Sadeghi, A.-R. A new approach for algebraically homomorphic encryption. Cryptology ePrint Archive, Report 2008/422, 2008. http://eprint.iacr.org/.

[5] Aur, E., and Schőnfeld, D. P3ers : Privacy-preserving peer review system, 2011.

[6] Bellare, M., and Rogaway, P. Optimal asymmetric encryption how to encrypt with rsa. Springer-Verlag, pp. 92–111.

[7] Benaloh, J. Dense probabilistic encryption. In *In Proceedings of the Workshop on Selected Areas of Cryptography* (1994), pp. 120–128.

[8] Boneh, D., and Freeman, D. M. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. Cryptology ePrint Archive, Report 2010/453, 2010. http://eprint.iacr.org/.

[9] Boneh, D., Goh, E.-J., and Nissim, K. Evaluating 2-dnf formulas on ciphertexts. In *Proceedings of Theory of Cryptography Conference 2005* (2005), J. Killian, Ed., vol. 3378 of *LNCS*, Springer, pp. 325–342.

[10] Brakerski, Z. Fully homomorphic encryption without modulus switching from classical gapsvp. Cryptology ePrint Archive, Report 2012/078, 2012. http://eprint.iacr.org/.

[11] Brakerski, Z., Gentry, C., and Vaikuntanathan, V. Fully homomorphic encryption without bootstrapping. *IACR Cryptology ePrint Archive 2011* (2011), 277.

[12] Brakerski, Z., and Vaikuntanathan, V. Efficient fully homomorphic encryption from (standard) lwe. Cryptology ePrint Archive, Report 2011/344, 2011. http://eprint.iacr.org/.

[13] Catalano, D., Gennaro, R., Howgrave-Graham, N., and Nguyen, P. Q. Paillier's cryptosystem revisited. In *ACM Conference on Computer and Communications Security* (2001), pp. 206–214.

[14] Chaum, D., and Van Heyst, E. Group signatures. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques* (Berlin, Heidelberg, 1991), EUROCRYPT'91, Springer-Verlag, pp. 257–265.

[15] Choi, D.-H., Choi, S., and Won, D. Improvement of probabilistic public key cryptosystems using discrete logarithm. In *Proceedings of the 4th International Conference Seoul on Information Security and Cryptology* (London, UK, UK, 2002), ICISC '01, Springer-Verlag, pp. 72–80.

[16] Coron, J.-S., Naccache, D., Paillier, P., and International, G. C. Accelerating okamoto-uchiyama's public-key cryptosystem, 1999.

[17] Damgård, I., and Jurik, M. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography : Public Key Cryptography* (London, UK, UK, 2001), PKC '01, Springer-Verlag, pp. 119–136.

[18] Diffie, W., and Hellman, M. E. New directions in cryptography. *IEEE Transactions on Information Theory 22*, 6 (1976), 644–654.

[19] Fan, J., and Vercauteren, F. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. http://eprint.iacr.org/.

[20] Fellows, M., and Koblitz, N. Combinatorial cryptosystems galore! In *Finite Fields : Theory, Applications, and Algorithms*, G. L. Mullen and P. J.-S. Shiue, Eds., vol. 168 of *Contemporary Mathematics*. 1994, pp. 51–61.

[21] Fousse, L., Lafourcade, P., and Alnuaimi, M. Benaloh's dense probabilistic encryption revisited. In *Progress in Cryptology - AFRICACRYPT 2011 - 4th International Conference on Cryptology in Africa, Dakar, Senegal, July 5-7, 2011. Proceedings* (2011), vol. 6737 of *Lecture Notes in Computer Science*, Springer, pp. 348–362.

[22] Galbraith, S. D. Elliptic curve paillier schemes, 2001.

[23] Gamal, T. E. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO* (1984), pp. 10–18.

[24] Gentry, C. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[25] Georges FENECH, D. d. R. Le placement sous surveillance ctronique mobile - rapport de la mission confiar le premier ministre orges fenech, Avril 2005.

[26] Goldreich, O., Micciancio, D., Safra, S., and Seifert, J.-P. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters 71*, 2 (1999), 55–61.

[27] Goldwasser, S., and Kharchenko, D. Proof of plaintext knowledge for the ajtai-dwork cryptosystem. In *Proceedings of the Second international conference on Theory of Cryptography* (Berlin, Heidelberg, 2005), TCC'05, Springer-Verlag, pp. 529–555.

[28] Goldwasser, S., and Micali, S. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC* (1982), pp. 365–377.

[29] Hoffstein, J., Pipher, J., and Silverman, J. H. Ntru : A ring-based public key cryptosystem. In *Lecture Notes in Computer Science* (1998), Springer-Verlag, pp. 267–288.

[30] Howgrave-Graham, N. Approximate integer common divisors. In *CaLC* (2001), pp. 51–66.

[31] Karp, R. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[32] Kawachi, A., Tanaka, K., and Xagawa, K. Multi-bit cryptosystems based on lattice problems. In *Proceedings of the 10th international conference on Practice and theory in public-key cryptography* (Berlin, Heidelberg, 2007), PKC'07, Springer-Verlag, pp. 315–329.

[33] Koblitz, N. Elliptic Curve Cryptosystems. *Mathematics of Computation 48*, 177 (1987), 203–209.

[34] Lee, M. S. On the sparse subset sum problem from gentry-halevi's implementation of fully homomorphic encryption. Cryptology ePrint Archive, Report 2011/567, 2011. http://eprint.iacr.org/.

[35] Lenstra, A. K., Lenstra, H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Math. Ann. 261* (1982), 515–534.

[36] Lyubashevsky, V., Peikert, C., and Regev, O. On ideal lattices and learning with errors over rings. In *EUROCRYPT* (2010), pp. 1–23.

[37] Memon, N. D., and Wong, P. W. A buyer-seller watermarking protocol. *IEEE Transactions on Image Processing 10*, 4 (2001), 643–649.

[38] Menezes, A. J., Vanstone, S. A., and Oorschot, P. C. V. *Handbook of Applied Cryptography*, 1st ed. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[39] Micciancio, D., and Regev, O. Lattice-based cryptography, 2008.

[40] Miller, V. S. Use of elliptic curves in cryptography. In *Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85* (New York, NY, USA, 1986), Springer-Verlag New York, Inc., pp. 417–426.

[41] Naccache, D., and Stern, J. A new public-key cryptosystem. In *EUROCRYPT* (1997), pp. 27–36.

[42] Naccache, D., and Stern, J. A new public key cryptosystem based on higher residues. In *ACM Conference on Computer and Communications Security* (1998), pp. 59–66.

[43] of Excellence in Cryptology II (ECRYPT II), E. N. Yearly report on algorithms and keysizes (2011), June 2011. D.SPA.17 Rev. 1.0, ICT-2007-216676.

[44] Okamoto, T., and Uchiyama, S. A new public-key cryptosystem as secure as factoring. In *In Eurocrypt '98, LNCS 1403* (1998), Springer-Verlag, pp. 308–318.

[45] Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *IN ADVANCES IN CRYPTOLOGY EUROCRYPT 1999* (1999), Springer-Verlag, pp. 223–238.

[46] Paillier, P. Trapdooring discrete logarithms on elliptic curves over rings, 2000.

[47] Peikert, C., and Waters, B. Lossy trapdoor functions and their applications, 2007.

[48] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In *STOC* (2005), pp. 84–93.

[49] Rivest, R., Adleman, L., and Dertouzos, M. On data banks and privacy homomorphisms. In *Foundations on Secure Computation, Academia Press* (1978), pp. 169–179.

[50] Rivest, R. L., Shamir, A., and Adleman, L. M. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 2 (1978), 120–126.

[51] Sakurai, K., and Takagi, T. On the security of a modified paillier public-key primitive. In *Proceedings of the 7th Australian Conference on Information Security and Privacy* (London, UK, UK, 2002), ACISP '02, Springer-Verlag, pp. 436–448.

[52] Sander, T., Young, A. L., and Yung, M. Non-interactive cryptocomputing for nc^1. In *FOCS* (1999), pp. 554–567.

[53] Schmidt-Samoa, K., and Takagi, T. Paillier's cryptosystem modulo p2q and its applications to trapdoor commitment schemes. In *Proceedings of the 1st international conference on Progress in Cryptology in Malaysia* (Berlin, Heidelberg, 2005), Mycrypt'05, Springer-Verlag, pp. 296–313.

[54] Shanks, D. Five number-theoretic algorithms. In *Proceedings of the Second Manitoba Conference on Numerical Mathematics (Univ. Manitoba, Winnipeg, Man., 1972)* (Winnipeg, Man., 1973), Utilitas Math.

[55] Shoup, V. Oaep reconsidered. In *CRYPTO* (2001), pp. 239–259.

[56] van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2009/616, 2009. http://eprint.iacr.org/.

[57] van Emde-Boas, P. *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Report. Department of Mathematics. University of Amsterdam. Department, Univ., 1981.

[58] Xiao, L., Bastani, O., and Yen, I.-L. An efficient homomorphic encryption protocol for multi-user systems. Cryptology ePrint Archive, Report 2012/193, 2012. http://eprint.iacr.org/.

Annexes

10 Cryptosystèmes simplement homomorphes

10.1 RSA

Auteurs : Léonard ADLEMAN - Ronald RIVEST - Adi SHAMIR [50]
Année : 1978
Génération des clés : Alice génère deux grands nombres premiers p et q et calcule $N = pq$. Elle choisit aléatoirement e [27] premier avec $\phi(N) = (p-1)(q-1)$ et calcule $d = e^{-1}\ mod\ \phi(N)$
Clé publique : (N, e)
Clé privée : $(p, q, d, \phi(N))$
Chiffrement : Bob chiffre son message $m \in \mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$ en calculant $c = m^e\ mod\ N$, et l'envoie à Alice.
Déchiffrement : Alice déchiffre le message reçu en calculant $c^d = m^{ed} = m\ modN$
Problème sous-jacent : Problème RSA (hypothèse plus forte que la factorisation)
Avantages : A l'époque, c'est le premier cryptosystème à clé publique !
Inconvénients : Le chiffrement est déterministe, donc non sémantiquement sûr. Des versions probabilistes existent (OAEP/OAEP+), mais elles font perdre le caractère homomorphe du système...
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2)\ mod\ N) = (m_1 \times m_2)\ mod\ N$$

10.2 Goldwasser-Micali

Auteurs : Shafi GOLDWASSER - Silvio MICALI [28]
Année : 1983
Génération des clés : Alice génère deux grands nombres premiers p et q et calcule $N = pq$. Elle choisit aléatoirement un non-résidu quadratique y tel que son symbole de Jacobi soit $+1$, c'est-à-dire $\forall z \in \mathbb{Z}_N, z^2\ mod\ N \neq y$ et $(\frac{y}{N}) = +1$
Clé publique : (N, y)
Clé privée : (p, q)
Chiffrement : Bob décompose son message en binaire : $m = (m_0, \ldots, m_{k-1}) \in \mathbb{Z}_2^k$ et le chiffre en $c = (c_0, \ldots, c_{k-1}) \in \mathbb{Z}_N^k$ de telle sorte que $\forall 0 \leq i \leq k-1, c_i = x^2 z^{m_i}\ mod\ N$, où $x \in \mathbb{Z}_\mathbb{N}$ est choisi uniformément au hasard.
Déchiffrement : $\forall 0 \leq i \leq k-1, m_i = 1$ si c_i est un carré modulo N, 0 sinon.
Problème sous-jacent : Résidualité quadratique (hypothèse plus forte que la factorisation)
Avantages : Pas de fuite d'information
Inconvénients : Expansion du message ($\times N$)

27. Il peut être judicieux de choisir e de faible poids de Hamming pour accélérer le chiffrement

Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2) \ mod \ N) = m_1 \oplus m_2$$

où $\oplus$ correspond à un OU exclusif.

10.3 ElGamal

Auteur : Taher ELGAMAL [23]
Année : 1983
Génération des clés : Alice génère un grand nombre premier p tel que $p-1$ ait au moins un large facteur premier. Elle choisit un générateur g de $\mathbb{Z}_p$. Elle choisit également un x uniformément au hasard dans $\{0, \ldots, p-1\}$ et calcule $h = g^x \ mod \ p$.
Clé publique : (g, h, p)
Clé privée : x
Chiffrement : Pour chiffrer son message $m \in \mathbb{Z}_p$, Bob choisit uniformément au hasard $y \in \{0, \ldots, p-1\}$, calcule $c_1 = g^y \ mod \ p$ et $c_2 = mh^y \ mod \ p$ et envoie (c_1, c_2) à Alice.
Déchiffrement : Alice calcule $K = c_1^x \ mod \ p$ et déchiffre son message en calculant $m = c_2 K^{-1} \ mod \ p$
Problème sous-jacent : Problème décisionnel de Diffie-Hellmann (hypothèse plus forte que le logarithme discret)
Avantages : Permet de faire de l'échange de clé simplement
Inconvénients : Expansion du message ($\times 2$)
Homomorphie : Si $(c_{1,1}, c_{1,2}) = Enc_{pk}(m_1)$ et $(c_{2,1}, c_{2,2}) = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_{1,1} \times c_{2,1} \ mod \ p, c_{1,2} \times c_{2,2} \ mod \ p)) = (m_1 \times m_2) \ mod \ p$$

10.4 Benaloh / Fousse - Lafourcade - Alnuaimi

Auteurs : Josh BENALOH [7] puis [28] Laurent FOUSSE - Pascal LAFOURCADE - Mohamed ALNUAIMI [21]
Année : 1994 puis 2010
Génération des clés : Alice choisit la taille r des blocs, et deux grands premiers p et q de telle sorte que $r|p-1$, $pgcd(r, \frac{p-1}{r}) = 1$, et $pgcd(r, q-1) = 1$. Elle calcule $N = pq$, et choisit $y \in \mathbb{Z}_N^*$ uniformément au hasard tel que $y^{\frac{(p-1)(q-1)}{r}} \ mod \ N \neq 1$. Dans la version corrigée, $y = g^\alpha \ mod \ p$ où g est un générateur de $\mathbb{Z}_p^*$ avec $pgcd(\alpha, r) = 1$, et on exige de plus que r soit premier.
Clé publique : (y, r, N)
Clé privée : (p, q)

28. Ce cryptosystème dans sa version originale présentait un défaut sur les choix des paramètres

Chiffrement : Pour chiffrer son message $m \in \mathbb{Z}_r$, Bob choisit uniformément au hasard $u \in \mathbb{Z}_N^*$, calcule et envoie $c = y^m u^r \ mod \ N$ à Alice.
Déchiffrement : Alice doit faire une recherche exhaustive pour trouver quel $i \in \{0, \ldots, r-1\}$ vérifie $(y^{-i} c \ mod \ N)^{\frac{(p-1)(q-1)}{r}} \ mod \ N = 1$
Problème sous-jacent : Problème de résidualité d'ordre supérieur (hypothèse plus forte que la factorisation)
Avantages : "Choix" de la taille des blocs
Inconvénients : Temps de déchiffrement potentiellement long lorsque r est grand, expansion ($\times \frac{N}{r}$) potentiellement grande lorsque r est petit par rapport à N
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$Dec_{sk}((c_1 \times c_2) \ mod \ N) = (m_1 + m_2) \ mod \ r \ et \ Dec_{sk}(c_2^{m_1} \ mod \ N) = (m_1 \times m_2) \ mod \ r$

10.5 Naccache-Stern knapsack cryptosystem

Auteurs : David NACCACHE - Jacques STERN [41]
Année : 1997
Génération des clés : Alice choisit un grand nombre premier p et établit la liste des n premiers nombres premiers p_i tels que $p > \prod_{i=0}^{n} p_i$ et $p < \prod_{i=0}^{n+1} p_i$ (à partir de $p_0 = 2$). Elle génère $s < p-1$ aléatoirement tel que $pgcd(p-1, s) = 1$. Les clés publiques sont les $n+1$ racines $v_i = \sqrt[s]{p_i} \ mod \ p$
Clé publique : $(v_0, \ldots, v_n, p)$
Clé privée : s
Chiffrement : Pour chiffrer son message $m = (m_0, \ldots, m_n) \in \mathbb{Z}_2^{n+1}$, Bob calcule $c = \prod_{i=0}^{n} v_i^{m_i} \ mod \ p$ et l'envoie à Alice.
Déchiffrement : Alice déchiffre le message reçu en calculant

$$m = \sum_{i=0}^{n} \frac{2^i}{p_i - 1} \times (pgcd(p_i, c^s \ mod \ p) - 1)$$

Problème sous-jacent : Problème du sac à dos
Avantages : Original, taux d'expansion presque nul.
Inconvénients : Déterministe, donc non sémantiquement sûr. Pas de sécurité prouvable.
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2) \ mod \ p) = (m_1 + m_2) \ mod \ 2 = m1 \oplus m_2$$

10.6 Naccache-Stern

Auteurs : David NACCACHE - Jacques STERN [42]
Année : 1998
Génération des clés : Alice choisit une famille de k nombres premiers distincts p_i, calcule $u = \prod_{i=1}^{\frac{k}{2}} p_i$ et $v = \prod_{i=\frac{k}{2}+1}^{k} p_i$, $\sigma = uv$, et choisit deux grands nombres premiers a et b tels que $p = 2au + 1$ et $q = 2bv + 1$ soient premiers, et g uniformément

au hasard d'ordre $aubv = \frac{\phi(N)}{4}$ où $N = pq$.
Clé publique : (g, σ, N)
Clé privée : (p, q)
Chiffrement : Pour chiffrer son message $m \in \mathbb{Z}_\sigma$, Bob choisit uniformément au hasard $x \in \mathbb{Z}_N^*$, calcule et envoie $c = x^\sigma g^m \ mod \ N$ à Alice.
Déchiffrement : Alice détermine d'abord $m_i = m \ mod \ p_i$ en calculant $c_i = c^{\frac{\phi(n)}{p_i}} \ mod \ N$ et en déterminant exhaustivement quel $j \in \{1, \ldots, p_i - 1\}$ vérifie $c_i = c^{j\frac{\phi(N)}{p_i}} \ mod \ N$. Cette dernière égalité est vérifiée pour $j = m_i$. Lorsqu'Alice a tous les m_i pour $i \in \{1, \ldots, k\}$, elle calcule par restes chinois $m \ mod \ \sigma$.
Problème sous-jacent : Problème de résidualité d'ordre supérieur (hypothèse plus forte que la factorisation)
Avantages : Généralisation du crytposystème de Benaloh
Inconvénients : Temps de déchiffrement potentiellement long lorsque les p_i sont grands
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2) \ mod \ N) = (m_1 + m_2) \ mod \ \sigma \ et \ Dec_{sk}((c_1^k) \ mod \ N) = k \cdot m_1 \ mod \ \sigma$$

10.7 Okamoto - Uchiyama

Auteurs : Tatsuaki OKAMOTO - Shigenori UCHIYAMA [44]
Année : 1998
Génération des clés : Alice génère deux grands nombres premiers p et q de k bits, calcule $N = p^2 q$, choisit $g \in \mathbb{Z}_N^*$ au hasard tel que $g^p \ mod \ p^2 \neq 1$, et calcule $h = g^N \ mod \ N$.
Clé publique : (n, g, h, k)
Clé privée : (p, q)
Chiffrement : Pour chiffrer son message $m \in \{1, \ldots, 2^{k-1} - 1\}$, Bob choisit uniformément au hasard $r \in \mathbb{Z}_N^*$, calcule et envoie $c = g^m h^r \ mod \ N$ à Alice.
Déchiffrement : Alice déchiffre le message reçu en calculant $m = \frac{c^{p-1} \ mod \ p^2}{g^{p-1} \ mod \ p^2} \ mod \ p$.
Problème sous-jacent : Factorisation
Avantages : Hypothèse de sécurité "assez faible"
Inconvénients : Sécurité sémantique sous l'hypothèse du sous-groupe d'ordre p
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2) \ mod \ N) = (m_1 + m_2) \ mod \ N$$

Improved Okamoto - Uchiyama
Auteurs : Jean-Sébastien CORON - David NACCACHE - Pascal PAILLIER [16]
Année : 1999
Génération des clés : Alice génère deux grands nombres premiers p et q de k bits de telle sorte que $p - 1$ ait un grand facteur premier t (environ 160 bits pour un module RSA de 1024 bits), elle calcule $N = p^2 q$, choisit $g \in \mathbb{Z}_N^*$ au hasard tel que $g^p \ mod \ p^2 \neq 1$, et calcule $h = g^{\frac{N \cdot (p-1)}{t}} \ mod \ N$.
Clé publique : (n, g, h, k)

Clé privée : (p, q)
Chiffrement : Pour chiffrer son message $m \in \{1, \ldots, 2^{k-1} - 1\}$, Bob choisit uniformément au hasard $r \in \mathbb{Z}_N^*$, calcule et envoie $c = g^m h^r \ mod \ N$ à Alice, comme dans le cryptosystème initial.
Déchiffrement : Alice déchiffre le message reçu en calculant $m = \frac{c^t \ mod \ p^2}{g^{p-1} \ mod \ p^2} \ mod \ p$.
Avantages : Le chiffré est élevé à la puissance t dans le déchiffrement, plutôt que $p - 1$, qui est beaucoup plus grand.

Modified Okamoto - Uchiyama
Auteurs : Dug-Hwan CHOI - Seungbok CHOI - Dongho WON [15]
Année : 2001
Génération des clés : $N = p^2 q$ est généré comme précédemment. Soient $c \in \mathbb{Z}_{p^2}$, $b = (\frac{c^{p-1}-1}{p} - 1) \cdot c \ mod \ p$, et $g = bp + c \ mod \ p^2$. Alors $\frac{(g^{p-1} \ mod \ p^2)-1}{p} = 1 \ mod \ p$.
Chiffrement : Comme dans le système initial.
Déchiffrement : $m = \frac{(c^{p-1} \ mod \ p^2)-1}{p} \ mod \ p$
Avantage : Déchiffrement rapide
Inconvénient : Sakurai and Takagi ont souligné qu'il n'était toujours pas prouvé que la fonction de chiffrement ne pouvait pas être inversée lorsque le choix de g était restreint à cette forme [51].

10.8 Paillier

Auteur : Pascal PAILLIER [45]
Année : 1999
Génération des clés : Alice choisit deux grands nombres premiers p et q et calcule $N = pq$ et $\lambda = \lambda(N) = ppcm(p - 1, q - 1)$. Elle choisit g aléatoirement tel que $pgcd(\frac{(g^\lambda \ mod \ n^2)-1}{N}, N) = 1$.
Clé publique : (g, N)
Clé privée : (p, q)
Chiffrement : Pour chiffrer son message $m \in \mathbb{Z}_N$, Bob choisit uniformément au hasard $r \in \mathbb{Z}_N^*$, calcule et envoie $c = g^m r^N \ mod \ N^2$ à Alice.
Déchiffrement : Alice déchiffre le message reçu en calculant $m = \frac{c^\lambda \ mod \ N^2}{g^\lambda \ mod \ N^2} \ mod \ N$.
Problème sous-jacent : Problème décisionnel de résidualité composée (hypothèse plus forte que la factorisation)
Avantages : Homomorphie
Inconvénients : Expansion du message ($\times 2$)
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2) \ mod \ N^2) = (m_1 + m_2) \ mod \ N \ et \ Dec_{sk}(c_2^{m_1} \ mod \ N^2) = (m_1 \times m_2) \ mod \ N$$

Fast Decryption Paillier
Génération des clés : $N = pq$ et $\lambda = \lambda(N) = ppcm(p - 1, q - 1)$ comme précédemment. Soit $g \in \mathbb{Z}_{N^2} *$ aléatoire d'ordre $\alpha \in \{1, \ldots, \lambda\}$.

Clé publique : (g, n)
Clé privée : (p, q, α)
Chiffrement : Pour chiffrer $m \in \mathbb{Z}_N$, Bob choisit uniformément au hasard $r \in \mathbb{Z}_N^*$, et calcule $c = g^{m+nr} \ mod \ N^2$.
Déchiffrement : Alice déchiffre en calculant $m = \frac{c^\alpha \ mod \ N^2}{g^\alpha \ mod \ N^2} \ mod \ N$.
Avantages : Chiffrement et déchiffrement plus rapides, une seule exponentiation modulaire pour le chiffrement contre deux dans le système initial.

Small Exponent Paillier
De façon analogue à RSA 10.1, Catalano, Gennaro, Howgrave-Graham, et Nguyen [13] ont suggéré d'utiliser un exposant public e de faible poids binaire, afin d'accélérer le processus de chiffrement. D'autre part, ce dernier peut encore être optimisé en choisissant $g = N + 1$ de sorte que $g^m = (1 + mN) \ mod \ N^2$. Le chiffrement devient ainsi $c = g^m y^e = y^e \cdot (1 + mN) \ mod \ n^2$, une faible exponentiation et une multiplication modulaire. Cependant, la sécurité du système se rapproche plus du problème RSA défini section **??**, consistant à extraire une racine e-ième modulo N.

Modified Paillier
Paillier a également proposé une variante à son schéma initial dans [45] :
Génération des clés : p, q premiers, $N = pq$ et $\lambda = \lambda(N) = ppcm(p - 1, q - 1)$, g aléatoire tel que $pgcd(\frac{(g^\lambda \ mod \ n^2)-1}{N}, N) = 1$ comme précédemment. Les clés publique et privée sont les mêmes que dans le schéma initial.
Chiffrement : Étant donné un message $m \in \mathbb{Z}_{N^2}$, déterminer $(m_1, m_2) \in \mathbb{Z}_N^2$ tel que $m = m_1 + Nm_2$. Calculer $c = g^{m_1} m_2^N \ mod \ N^2$.
Déchiffrement : Calculer $m_1 = \frac{(c^\lambda \ mod \ N^2)-1}{(g^\lambda \ mod \ N^2)-1} \ mod \ N$, $c' = cg^{-m_1} \ mod \ N$, et $m_2 = c' g^{N^{-1}}$. $m = m_1 + Nm_2$.
De façon étonnante, cette modification n'altère pas le caractère homomorphe du cryptosystème, mais le chiffré résultant d'une opération homomorphe n'est plus même : si $c = Enc_{pk}(m_1 + Nm_2)$ et $c' = Enc_{pk}(m'_1 + Nm'_2)$, alors

$$Dec_{sk}((c \times c') \ mod \ N^2) = (m_1 + m'_1 + N(m_2 + m'_2) \ mod \ N^2$$

$$et \ Dec_{sk}(c^k \ mod \ N^2) = (m_1 + k + N(m_2^k)) \ mod \ N^2$$

Avantages : Plus rapide, pas d'expansion
Inconvénient : Déterministe, donc non IND-CPA

10.9 Sander - Young - Yung

Auteurs : Tomas SANDER - Adam L. YOUNG - Moti YUNG [52]
Année : 1999
Génération des clés : De façon analogue au cryptosystème de Goldwasser-Micali 10.2, Alice génère deux grands premiers p et q, calcule $N = pq$, choisit aléatoirement un non-résidu quadratique y tel que $\frac{y}{N} = +1$

Clé publique : (N, y)
Clé privée : (p, q)
Chiffrement : Pour chiffrer un bit $b \in \mathbb{Z}_2$, Bob choisit un entier $l \geq 1$. Si $b = 0$, Bob choisit un vecteur aléatoire $b' \in (\mathbb{Z}_2)^l$ non nul. Si $b = 1$, $b' = 0^l$. Bob utilise la fonction de déchiffrement de Goldwasser-Micali pour chiffrer : $c = (GM.Enc(b'_1), \ldots, GM.Enc(b'_l))$
Déchiffrement : $\forall 1 \leq i \leq l, d_i = 1$ si c_i est un carré modulo N, 0 sinon. Si $d = 0^l$ alors $m = 1$, sinon $m = 0$.
Problème sous-jacent : Résidualité quadratique (hypothèse plus forte que la factorisation)
Avantages : Pas de fuite d'information
Inconvénients : Expansion du message ($\times l \cdot N$)
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \otimes c_2) \ mod \ N) = m_1 \ and \ m_2 = m_1 \times m_2 \ mod \ 2$$

où $\otimes$ correspond à une multiplication coordonnée par coordonnée.

10.10 Damgård - Jurik

Auteur : Ivan Damgård - Mads Jurik [17]
Année : 2001
Génération des clés : Alice choisit une dimension s, deux grands nombres premiers p et q et calcule $n = pq$ et $\lambda = \lambda(n) = ppcm(p-1, q-1)$. Elle choisit $g \in \mathbb{Z}^*_{n^{s+1}}$ et $x \in \mathbb{Z}^*_n$ aléatoirement tels que $g = (1+n)^j x \ mod \ n^{s+1}$ pour un j connu premier avec n et x. Elle choisit également d tel que $d \ mod \ n \in \mathbb{Z}^*_n$ et $\lambda | d$.
Clé publique : (g, n)
Clé privée : (p, q, j, d)
Chiffrement : Pour chiffrer son message $m \in \mathbb{Z}_{n^s}$, Bob choisit uniformément au hasard $r \in \mathbb{Z}^*_{n^{s+1}}$, calcule et envoie $c = g^m r^{n^s} \ mod \ n^{s+1}$ à Alice.
Déchiffrement : Alice déchiffre le message reçu en calculant $c^d \ mod \ n^{s+1} = (1+n)^{jmd \ mod \ n^s}$ et applique une version récursive du déchiffrement de Paillier pour obtenir jmd. Elle peut alors calculer $m = (jmd)(jd)^{-1} \ mod \ n^{s+1}$
Problème sous-jacent : Problème décisionnel de résidualité composée (hypothèse plus forte que la factorisation)
Simplification : En acceptant de ne plus généraliser le cryptosystème de Paillier, ce système peut être simplifier en fixant $g = n + 1$, $\lambda | d$ et $d \ mod \ n^s = 1$. Alors le déchiffrement se résume à calculer $c^d = (1+n)^{jmd} \ mod \ n^{s+1}$ et à appliquer la version récursive du déchiffrement de Paillier
Avantages : Généralisation du chiffrement de Paillier, expansion du message arbitrairement faible
Inconvénients : Procédure de déchiffrement potentiellement longue
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$$Dec_{sk}((c_1 \times c_2) \ mod \ n^{s+1}) = (m_1 + m_2) \ mod \ n^s \ et \ Dec_{sk}(c_2^{m_1} \ mod \ n^{s+1}) = (m_1 \times m_2) \ mod \ n^s$$

Length - Flexible Damgård - Jurik
Le cryptosystème précédent peut être modifié de sorte à ce que le choix de la taille s des blocs revienne à la personne qui chiffre les messages :
Clé publique : $N = pq$
Clé privée : (p, q, j, d)
Chiffrement : Bob choisit s de sorte que son message $m \in \mathbb{Z}_{N^s}$, et $r \in \mathbb{Z}^*_{N^{s+1}}$ et calcule $c = g^m r^{N^s} \ mod \ N^{s+1}$.
Déchiffrement : Alice détermine s de sorte que $c \in \mathbb{Z}_{N^{s+1}}$, et calcule $c^d \ mod \ N^{s+1} = (1+n)^{jmd \ mod \ n^s}$. En applicant la version récursive du déchiffrement de Paillier, Alice obtient jmd et peut alors calculer $m = (jmd) \cdot (jd)^{-1} \ mod \ N^{s+1}$.

Modied Length - Flexible Damgård - Jurik
La version décrite à l'instant peut encore être modifiée de sorte que plusieurs utilisateurs partage un même module public, tout en conservant les propriétés homomorphes du cryptosystème de Paillier 10.8 :
Génération des clés : Soient $p = 2p' + 1$ et $q = 2q' + 1$ des nombres premiers tels que p' et q' soient premiers. Soient $N = pq$, $\lambda = p'q'$, et $g \in \mathbb{Z}_N$ et $\alpha \in \mathbb{Z}_\lambda$ aléatoires, et $h = g^\alpha \ mod \ N$.
Clé publique : (N, g, h)
Clé privée : $(p, q, p', q', \alpha, \lambda)$
Chiffrement : Bob choisit s de sorte que son message $m \in \mathbb{Z}_{N^s}$ et $r \in \mathbb{Z}^*_N$ et calcule $c = (G, H) := (g^r \ mod \ N, (h^r \ mod \ n)^{n^s}(1+N)^m \ mod \ N^{s+1}$.
Déchiffrement : Alice détermine s de sorte que $H \in \mathbb{Z}_{N^{s+1}}$, et calcule
$m' = (H(G^\alpha \ mod \ N)^{-N^s}) \ mod \ N^{s+1} = (1+N)^m \ mod \ N^{s+1}$. En appliquant la version récursive du déchiffrement de Paillier, Alice obtient m.
Avantage : Une fois les paramètres générés, on peut créer d'autres instances utilisant le même module en choisissant aléatoirement $\alpha \in \mathbb{Z}_\lambda$ et en publiant $h = g^\alpha \ mod \ N$.

10.11 Elliptic Curve Paillier

Paillier a fourni un exemple d'application de son cryptosystème 10.8 aux courbes elliptiques, mais Galbraith a montré qu'il existait une faille dans cet exemple. Tout comme dans la version elliptique d'Okamoto et Uchiyama, le calcul du logarithme discret sur les courbes utilisées était simple, ce qui permettait de retrouver la clé privée à partir des informations publiques. Galbraith a tout de même fourni un exemple concret d'une version de Paillier sur les courbes elliptiques [22] :
Génération des clés : Alice choisit p et q, calcule $N = pq$, et choisit aléatoirement une courbe elliptique $E : y^2z = x^3 + axz^2 + bz^3$ sur $\mathbb{Z}_N$ telle que $pgcd(6(4a^3 + 27b^2), N) = 1$. Soit $M = ppcm(|E(\mathbb{F}_p)|, |E(\mathbb{F}_q)|)$, Q' un point choisi au hasard sur la courbe et $Q = NQ'$
Clé publique : (a, b, N, Q)
Clé privée : (p, q, M)
Chiffrement : Pour chiffrer un message $m \in \mathbb{Z}_N$, Bob choisit uniformément au hasard $r \in \mathbb{Z}^*_N$, et calcule $c = rQ + P_m$, où $P_m = (mN, 1, 0)$ est le point sur la courbe $E(\mathbb{Z}_{N^2}$ correspondant à $m \in \mathbb{Z}_N$.

Déchiffrement : Pour déchiffrer, Alice calcule $Mc = MrQ + MP_m = P_{mM} = (mM, 1, 0)$, puis $m = mMM^{-1}$
Problème sous-jacent : Étant donnés $Q \in E(\mathbb{Z}_{N^2}$ d'ordre divisant $|E(\mathbb{Z}_N)|$ et un poin au hasard $S \in E(\mathbb{Z}_{N^2}$, déterminer si S appartient au sous groupe engendré par Q (hypothèse plus forte que la factorisation)
Avantages : Hypothèse de sécurité "apparemment" assez forte
Inconvénients : Bien que basé sur les courbes elliptiques, la meilleure attaque contre ce système utilise la factorisation, le module N doit donc être assez large, on ne bénéficie pas des tailles de clés réduites comme c'est en général le cas avec la cryptographie sur les courbes elliptiques.
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$Dec_{sk}((c_1+c_2)\ mod\ N^2) = (m_1+m_2)\ mod\ N$ *et* $Dec_{sk}(kc_1\ mod\ N^2) = (km_1)\ mod\ N$

10.12 Schmidt-Samoa - Takagi

Auteur : Katja SCHMIDT-SAMOA - Tsuyoshi TAKAGI [53]
Année : 2005
Génération des clés : Alice choisit p et q tels que $p-1$ et $q-1$ aient un grand facteur premier, $p \nmid q-1$, $q \nmid p-1$ et l tel que $2^l < pq < 2^{l+1}$. Soit $N = p^2q$, et $d = N^{-1}\ mod\ (p-1)(q-1)$
Clé publique : (N, l)
Clé privée : (p, q, d)
Chiffrement : Pour chiffrer un message $m \in \mathbb{Z}_{2^l}$, Bob choisit uniformément au hasard $r \in \mathbb{Z}_N^*$, calcule $c = r^N(1+mN)\ mod\ N^2$.
Déchiffrement : Pour déchiffrer, Alice calcule $r = c^d\ mod\ pq$, puis $\frac{(r^{-N}c\ mod\ N^2)-1}{N}\ mod\ pq$.
Problème sous-jacent : Factorisation de $N = p^2q$
Avantages : Prend les avantages des cryptosystèmes de Paillier 10.8 et d'Okamoto-Uchiyama 10.7
Inconvénients : Expansion du message ($> \times 3$)
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$, alors

$Dec_{sk}((c_1 \times c_2)\ mod\ N^2) = (m_1+m_2)\ mod\ pq$ *et* $Dec_{sk}(c_1^k\ mod\ N^2) = (km_1)\ mod\ pq$

10.13 Boneh-Goh-Nissim

Auteurs : Dan BONEH - Eu-Jin GOH - Kobbi NISSIM [9]
Année : 2005
Génération des clés : Alice choisit deux grands nombres premiers q_1 et q_2 et calcule $n = q_1q_2$. Elle choisit un groupe $\mathbb{G}$ cyclique d'ordre n pour lequel il existe un groupe $\mathbb{G}_1$ cyclique d'ordre n et une application $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ telle que $\forall u, v \in \mathbb{G}, \forall a, b \in \mathbb{Z}, e(u^a, v^b) = e(u, v)^{ab}$. Elle choisit un générateur g de $\mathbb{G}$ tel que $e(g, g)$ soit un générateur de $\mathbb{G}_1$. Elle choisit également au hasard un générateur u de $\mathbb{G}$ et pose $h = u^{q_2}$. h est un générateur du sous-groupe d'ordre q_1 de $\mathbb{G}$.

Clé publique : $(n, \mathbb{G}, \mathbb{G}_1, e, g, h)$
Clé privée : q_1
Chiffrement : Pour chiffrer son message $m \in \{0, 1, \ldots, T)$, où $T < q_2$, Bob choisit uniformément au hasard $r \in \mathbb{Z}_n$, calcule et envoie $c = g^m h^r \ mod \ N^{s+1} \in \mathbb{G}$ à Alice.
Déchiffrement : Alice déchiffre le message reçu en calculant $c^{q_1} \ mod \ N^{s+1} = (g^{q_1})^m$ et calcule le logarithme discret de c^{q_1} en base g^{q_1}. Comme $0 \leq m \leq T$, cela nécessite $O(\sqrt{T})$ opérations dans $\mathbb{G}$ ainsi que le stockage de $O(log(T))$ éléments de $\mathbb{G}$ en utilisant la méthode lambda de Pollard [38].
Problème sous-jacent : Problème décisionnel d'appartenance à un sous-groupe
Avantages : Doublement homomorphe.
Inconvénients : Forte expansion du message pour accélérer le déchiffrement ou peu d'expansion mais déchiffrement très long.
Homomorphie : Si $c_1 = Enc_{pk}(m_1)$ et $c_2 = Enc_{pk}(m_2)$ avec $m_1, m_2 \in \{0, \ldots, T\}$, alors pour $r \in \{0, \ldots, n-1\}$ aléatoire,

$$Dec_{sk}(c_1 \times c_2 \times h^r) = (m_1 + m_2) \ mod \ n \ et \ Dec_{sk}(e(c_1, c_2) \times e(g, h)^r) = (m_1 \times m_2) \ mod \ n$$

11 Crible algébrique sur les corps de nombres

ECRYPT II, le réseau d'excellence européen en cryptologie II publie chaque année un rapport sur les algorithmes cryptographiques ainsi que la taille des clés à utiliser [43]. Concernant le problème de la factorisation et les autres problèmes dont la meilleure approche connue est la factorisation, la sécurité des cryptosystèmes qui en résultent est soumise à la complexité du crible algébrique.

ECRYPT II estime qu'actuellement un module RSA $N = pq$ de n bits (p et q sont deux nombres premiers de $\frac{n}{2}$ bits) a une sécurité équivalente à une clé symétrique de $s(n)$ bits, avec

$$s(n) = \sqrt[3]{\frac{64}{9}}(\log_2(\exp)) \sqrt[3]{(nln(2))(ln(ln(2)))^2} - 14$$

ce qui donne par exemple une équivalence entre un module RSA de 1024 et une clé symétrique de 72 bits.

12 Description de l'algorithme LLL et analyse

12.1 L'algorithme

ALGORITHM

`Input` : $B = (b_1, \ldots, b_n) \in \mathbb{Z}^{n \times n}$ base du réseau $\mathcal{L}(B)$, et $\delta \in [\frac{1}{4}, 1]$
`Output` : Base δ-LLL-réduite de $\mathcal{L}(B)$
    Calculer $\tilde{B} = (\tilde{b_1}, \ldots, \tilde{b_n})$ par Gram-Schmidt
    Étape de réduction :
        Pour $i = 2$ à n, faire :

Pour $j = i-1$ à 1, faire :

$b_i \leftarrow b_i - \lfloor \mu_{i,j} \rceil \cdot b_j$ avec $\mu_{i,j} = \frac{<\tilde{b_j}, b_i>}{<\tilde{b_j}, \tilde{b_j}>}$

Étape d'échange :

Si $\exists i \in \{1, \ldots, n\} / \delta ||\tilde{b_i}||^2 > ||\mu_{i+1,i}\tilde{b_i} + \tilde{b_{i+1}}||^2$, alors

$b_i \leftrightarrow b_{i+1}$

Recommencer du début.

Retourner $(b_1, \ldots, b_n)$.

12.2 Analyse de l'algorithme

Il est facile de voire que la condition d'arrêt de l'algorithme alors on ne doit pas avoir $\exists i \in \{1, \ldots, n\} / \delta ||\tilde{b_i}||^2 > ||\mu_{i+1,i}\tilde{b_i} + \tilde{b_{i+1}}||^2$. Autrement dit, l'algorithme se terminera lorsque $\forall 1 \leq i \leq n$ on a : $\delta \cdot ||\tilde{b_i}||^2 < ||\mu_{i+1,i}\tilde{b_i} + \tilde{b_{i+1}}||^2$, ce qui permet de respecter la seconde condition d'une base δ-LLL-réduite.

La première condition est assurée par la phase de réduction, qui permet sans changer le réseau engendré par les b_i, d'assurer cette condition, du fait que $||\lfloor x \rceil - x| \leq \frac{1}{2}, \forall x \in \mathbb{Z}$. Dans l'étape de réduction, Pour la i^{eme} itération dans la première boucle, et la j^{eme} dans la seconde, soit B la matrice dans la base orthonormale des vecteurs de Gram-Schmidt $\frac{\tilde{b_i}}{||\tilde{b_i}||}$, alors :

$$B = \begin{pmatrix} ||\tilde{b_1}|| & \leq \frac{1}{2}||\tilde{b_1}|| & \leq \frac{1}{2}||\tilde{b_1}|| & \ldots & * & * & \ldots & * \\ 0 & ||\tilde{b_2}|| & \leq \frac{1}{2}||\tilde{b_2}|| & \ldots & * & * & \ldots & * \\ 0 & 0 & ||\tilde{b_3}|| & \ldots & * & * & \ldots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & \underline{todo : \leq \frac{1}{2}||\tilde{b_j}||} & * & \ldots & * \\ 0 & 0 & 0 & \ldots & \leq \frac{1}{2}||b_{j+1}|| & * & \ldots & * \\ 0 & 0 & 0 & \ldots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & ||\tilde{b_i}|| & * & \ldots & * \\ \vdots & \vdots & \vdots & \ldots & \vdots & ||\tilde{b_{i+1}}|| & \ldots & \vdots \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & ||\tilde{b_n}|| \end{pmatrix}$$

Afin de s'assurer que la première condition est réalisée, il suffit de remarquer que les coefficients de Gram-Schmidt deviennent :

$$\begin{aligned} |\mu'_{i,j}| &= |\frac{< b_i - \lfloor \mu_{i,j} \rceil b_j, \tilde{b_j} >}{< \tilde{b_j}, \tilde{b_j} >}| \\ &= |\frac{< b_i, \tilde{b_j} >}{< \tilde{b_j}, \tilde{b_j} >} - \lfloor \mu_{i,j} \rceil \cdot \frac{< b_j, \tilde{b_j} >}{< \tilde{b_j}, \tilde{b_j} >}| \\ &= |\underbrace{\frac{< b_i, \tilde{b_j} >}{< \tilde{b_j}, \tilde{b_j} >}}_{x} - \underbrace{\lfloor \frac{< b_i, \tilde{b_j} >}{< \tilde{b_j}, \tilde{b_j} >} \rceil}_{\lfloor x \rceil}| \leq \frac{1}{2} \end{aligned}$$

car $< b_j, \tilde{b_j} >=< \tilde{b_j}, \tilde{b_j} >$.

Une étude détaillée [29] montre que le nombre d'itérations est polynomial en $M = \max(n, \log(\max_i ||b_i||)$, et que le temps d'exécution de chaque itération en M, ce qui au total, est polynomial en M. Plus précisément, la complexité de LLL est $O(n^6 \log \max_i ||b_i||)$.

29. Que le lecteur pourra consulter dans le second cours de Regev à http://www.cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/lll.pdf