UVSQ
université PARIS-SACLAY

Précision numérique en chiffrement homomorphe : focus sur CKKS

Mohamed Jiddou

LI-PARAD, UVSQ – Paris-Saclay

Introduction

Présentation

- Étudiant ingénieur (5e année, double diplôme)
- Master 2 SECRETS (cybersécurité)

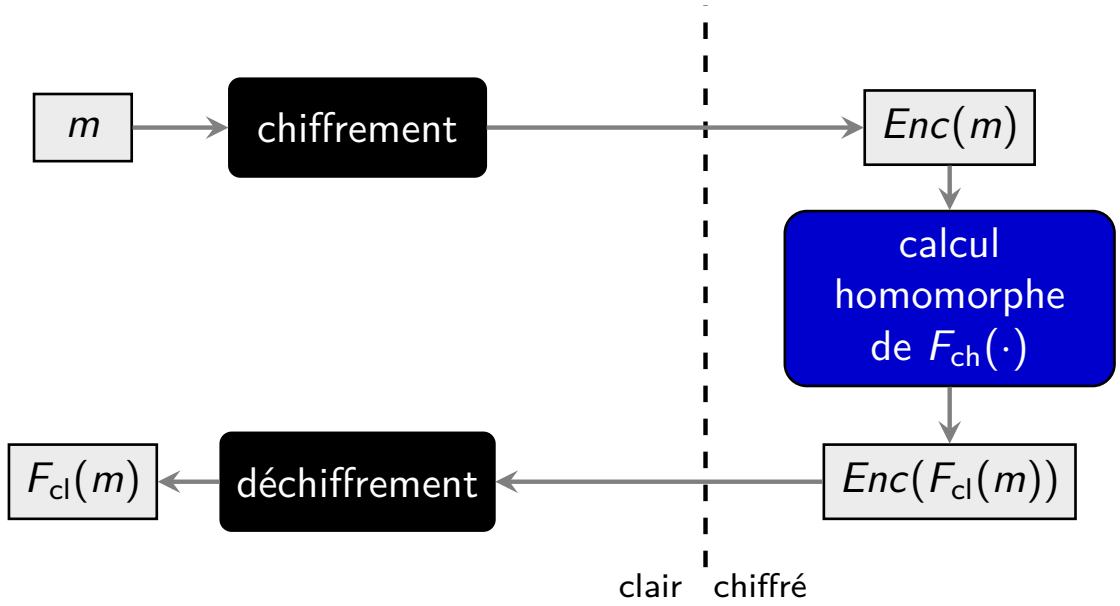
Stage de recherche

- Début : ce lundi
- Sujet : *Précision numérique en chiffrement homomorphe (CKKS)*

Principe du chiffrement homomorphe

- Calculer sur des données chiffrées
- sans jamais manipuler les données en clair pendant le calcul

$$\mathrm{Dec}(F_{\mathsf{ch}}(\mathrm{Enc}(m))) = F_{\mathsf{cl}}(m)$$



m : donnée, $\mathrm{Enc}/\mathrm{Dec}$: chiffrement/déchiffrement,

$F_{\mathsf{cl}}/F_{\mathsf{ch}}$: calcul clair / calcul chiffré

Précision en chiffrement homomorphe

Bruit en chiffrement homomorphe

- La plupart des schémas de chiffrement homomorphes manipulent des données bruitées
- Chaque opération ajoute et propage du bruit

Conséquence

- La précision décroît au fil du calcul
- Limite sur la profondeur des circuits évaluables

Précision en chiffrement homomorphe

Précision et schéma CKKS (Cheon, Kim, Kim and Song)

- Clair : vecteur complexe
- Chiffré : coefficients entiers (approximation à précision fixe via l'échelle Δ)

Deux sources d'erreur

- approximation numérique (comme en flottant classique)
- bruit cryptographique (propre au chiffrement)

Lien avec FPT4

- analyse de la précision des calculs

Espaces de calcul dans CKKS

Deux anneaux

$$R_q = \mathbb{Z}_q[X]/(X^N + 1)$$

Espace des encodés m, et les clés secrètes s

$$R_q \times R_q$$

q premier, $q \equiv 1\ [2N]$,
ou produit de tels entiers distincts

(efficacité)

Espace des chiffrés

Encodage dans CKKS

- **Message** : $\mu \in \mathbb{C}^{N/2}$
- **Extension** : $\mu \rightarrow z \in \mathbb{C}^N$ (symétrie conjuguée $\Rightarrow$ coefficients réels)
- **Pré-encodage** :
  $z \longrightarrow m \in \mathbb{R}[X]/(X^N + 1)$
  (via une transformée de type Fourier inverse)
- **Mise à l'échelle** :
  $m \mapsto \Delta \cdot m$
- **Discrétisation** :
  $\Delta \cdot m \longrightarrow \lfloor \Delta \cdot m \rceil$
  $\Rightarrow$ passage réel $\rightarrow$ entier $\Rightarrow$ erreur d'arrondi

Encodage dans CKKS

- **Encodage** : $\mu \mapsto \lfloor \Delta \cdot m \rceil \bmod q$

Encodage

$$\mu \xrightarrow{1} z \xrightarrow{2} m \xrightarrow{3} \Delta m \xrightarrow{4} \lfloor \cdot \rceil \xrightarrow{5} R_q$$

1 - extension 2 - interpolation 3 - mise à l'échelle 4 - discrétisation 5 - réduction modulo q

Encodage dans CKKS : Exemple

- **Message :** $\mu = (3 + 4i,\ 2 - i), \quad N = 4,\ \Delta = 64$
- **Interpolation :** $m(X) = \frac{5}{2} + \sqrt{2}X + \frac{5}{2}X^2 + \frac{\sqrt{2}}{2}X^3$
- **Mise à l'échelle :**

$$\Delta m(X) = 160 + 64\sqrt{2}X + 160X^2 + 32\sqrt{2}X^3$$

- **Arrondi :**

$$\lfloor \Delta m(X) \rceil = 160 + 91X + 160X^2 + 45X^3$$

- **Décodage :** $\approx (3.0082 + 4.0026i,\ 1.9918 - 0.9974i)$

Chiffrement et déchiffrement

- **Chiffrement :** $ct = (a, b) \in R_q^2$ (détails non essentiels ici, sécurité basée sur RLWE)

- **Déchiffrement :** $\langle ct, sk \rangle = b + a \cdot s = \Delta \cdot m + e \pmod{q}$
  avec $sk = (1, s)$
  a public, s secret à coefficients dans $\{-1, 0, 1\}$
  e : bruit aléatoire (RLWE), à coefficients petits
  $\Rightarrow \langle ct, sk \rangle \approx \Delta \cdot m$

RLWE : Ring Learning With Errors (problème de sécurité sous-jacent)

Addition homomorphe

- **Entrée :** $ct_1 = (a_1, b_1),\ ct_2 = (a_2, b_2)$ avec $b_i + a_i s = \Delta m_i + e_i$
- **Addition :**

$$ct_1 + ct_2 = (a_1 + a_2,\ b_1 + b_2)$$

- **Déchiffrement :**

$$(b_1 + b_2) + (a_1 + a_2)s = \Delta(m_1 + m_2) + (e_1 + e_2)$$

- **Conclusion :** $ct_1 + ct_2 \ \rightarrow\ m_1 + m_2$ (bruit : $e_1 + e_2$)

Multiplication homomorphe

- **Entrée :** $ct_1 = (a_1, b_1),\ ct_2 = (a_2, b_2)$

- **Produit homomorphe :**

  $ct_1 \otimes ct_2 = (c_0, c_1, c_2)$

  $$c_0 = b_1 b_2, \quad c_1 = a_1 b_2 + a_2 b_1$$

  $$c_2 = a_1 a_2$$

  $\langle ct_1 \otimes ct_2,\ sk \otimes sk \rangle = \langle ct_1, sk \rangle \langle ct_2, sk \rangle = \Delta^2 m_1 m_2 + \Delta m_1 e_2 + \Delta m_2 e_1 + e_1 e_2$ (bruit $\approx \Delta(m_1 e_2 + m_2 e_1) + e_1 e_2$)

- **Trois difficultés :**

  1. $ct_1 \otimes ct_2$ est de **dimension 3**
  2. Le **facteur d'échelle** est Δ^2
  3. L'**erreur** est d'**ordre quadratique**

Solution de la première difficulté

- **Idée :** remplacer le terme s^2 par une expression linéaire en s
- **Clé de relinearisation :** $a \leftarrow R_q$, e petit, $\mathrm{rlk} = (a, b)$ avec $b = -as + e + Qs^2$
- **Vérification :** $b + as = Q\,s^2 + e \approx Q\,s^2$ $\rightarrow$ chiffrement RLWE de $Q\,s^2$
- **Application :** $c_2 s^2 \approx (c_2/Q)(b + as)$
- **Nouveau chiffré :** $c_0' = c_0 + (c_2/Q)b$, $c_1' = c_1 + (c_2/Q)a$ $\rightarrow$ $ct' = (c_0', c_1')$

Rescaling (gestion de l'échelle)

Solution de la deuxième difficulté

- **Après multiplication :** $\langle ct, sk \rangle \approx \Delta^2 \cdot m \ [q]$
- **Idée :** ramener l'échelle à Δ
- **Hypothèse :** $q = q_1 q_2, \quad q_2 \approx \Delta$
- **Rescaling :** $ct' = \left\lfloor \frac{1}{q_2} \cdot ct \right\rceil \in R_{q_1}$
- **Effet :** $\langle ct', sk \rangle \approx \Delta \cdot m \ [q_1]$
- **Conséquence :** changement de module $q \to q_1$

Autres opérations

$ct : \langle ct, sk \rangle \approx \Delta \cdot m \ [q]$

avec $sk = (1, s)$

Rotation

$\langle ct(X^k), sk(X^k) \rangle \approx \Delta \cdot m(X^k)$

Pour le clair μ : **permutation des coordonnées**

Conjugaison

$\langle ct(X^{-1}), sk(X^{-1}) \rangle \approx \Delta \cdot m(X^{-1})$

Pour le clair μ : **conjugaison des coordonnées**

chaque automorphisme nécessite un changement de clé (comme pour la multiplication) , ce qui introduit un bruit numérique

Bootstrap

- **OBJECTIF :** $ct\ [q_0] \rightarrow ct'\ [Q]$ pour un module $Q \gg q_0$ et (à peu près) le même clair/encodé

- **ModRaise :** $\langle ct, sk \rangle \approx \Delta \cdot m\ [q_0] \Rightarrow$ $\langle ct, sk \rangle \approx \Delta \cdot m + q_0 \cdot I$ pour un certain $I \in R$

  L'équation est dans R, donc modulo q pour tout q.

EvalMod : chercher une fonction f telle que
$f(\Delta \cdot m + q_0 \cdot I) \approx \Delta \cdot m$

Après passage à l'échelle, on souhaite réduire modulo 1 :

$$\frac{\Delta}{q_0} m + I \quad \Longrightarrow \quad \frac{\Delta}{q_0} m$$

Mais on ne sait qu'évaluer des polynômes.

Idée : chercher une fonction périodique de période 1 qui supprime la partie entière.

Exemple :

$$x \longmapsto \frac{1}{2\pi} \sin(2\pi x)$$

Cette fonction peut ensuite être approchée par un polynôme.

Sources d'erreur dans CKKS

Encodage

Quantification lors du passage vers $R = \mathbb{Z}[X]/(X^N + 1)$:

$$\Delta \cdot m \longrightarrow \lfloor \Delta \cdot m \rceil \Rightarrow \text{erreur d'arrondi initiale.}$$

Addition

$e_{\text{add}} = e_1 + e_2$: croissance linéaire des erreurs.

Sous hypothèse d'erreurs centrées et indépendantes :

$$\|e\| = O(\sqrt{n}), \quad \text{où } n \text{ est le nombre d'additions.}$$

Multiplication

$e_{\text{mult}} \approx m_1 e_2 + m_2 e_1 + e_1 e_2$

Amplification non linéaire des erreurs (dominée par les produits).

Sources d'erreur dans CKKS

Rescale

Division par q suivie d'un arrondi :

$$\frac{c}{q} \longrightarrow \left\lfloor \frac{c}{q} \right\rceil \quad \Rightarrow \quad \text{perte de précision (troncature).}$$

Bootstrap

Réinitialise le niveau de module, mais introduit une nouvelle erreur.

Erreur provenant principalement de :

- l'approximation polynomiale (EvalMod)
- l'accumulation des erreurs précédentes

$\Rightarrow$ erreur plus difficile à contrôler précisément.

Perspectives

- Peut-on modéliser - et si oui, comment - la propagation des erreurs à l'échelle d'un DAG de calcul CKKS ?
- Comment adapter l'arrondi stochastique à l'arithmétique en virgule fixe de CKKS ?
- Sous quelles hypothèses ce SR adapté est-il plus précis que l'arrondi au plus proche ?