Représentation de Nombres pour les Algorithmes

Choix de l'Arithmétique pour une Implémentation Réaliste de FV

Vincent Zucca

Année 2016-2017

Problématique

- Les éléments de base appartiennent à $\mathcal{R}_q \cong (\mathbb{Z}/q\mathbb{Z})[X]/(X^n + 1)$
- Paramètres réalistes de sécurité $\implies$ q et n très grand.
- Temps produit naïf de 2 éléments dans $\mathcal{R}_q$
  - $n = 2048$ et $\log_2(q) = 95 \rightarrow 1.824$ s.
  - $n = 4096$ et $\log_2(q) = 190 \rightarrow 7{,}872$ s.
  - $n = 8192$ et $\log_2(q) = 390 \rightarrow 35{,}419$ s.
  - $n = 16384$ et $\log_2(q) = 790 \rightarrow 188.657$ s.
- Imaginez pour une multiplication homomorphique !

$\implies$ Utiliser des représentations adaptées pour accélérer les calculs

Arithmétique sur les coefficients

- Pas de contraintes sur la forme du module q.
  $\Longrightarrow$ choisir q comme produit de k premiers entre eux $q_1, \cdots, q_k$.
- Les q_i sont choisis de même taille (idéalement un mot machine).
- Le CRT donne l'isomorphisme d'anneau $\mathcal{R}_q \cong \mathcal{R}_{q_1} \times \cdots \times \mathcal{R}_{q_k}$
  $\Longrightarrow$ Arithmétique en parallèle sur k corps de taille $\log_2(q)/k$.
- Coût multiplication $\mathcal{O}(\log_2(q)^2) \rightarrow \mathcal{O}((\log_2(q)/k)^2)$.

Remarque : au lieu de représenter $a \in \mathcal{R}_q$ comme un tableau de n coefficients, on le représente comme une matrice avec $n \times k$ coefficients.

Application

$q = 221 = 13 \times 17$, donner la représentation en RNS du chiffré précédent $\boldsymbol{c} = (49X - 30, 13X + 73)$.

Solution

$\boldsymbol{c} = ((-3X - 4, -5), (-2X + 4, -4X + 5))$

Grâce au RNS on obtient les timings suivants pour des moduli de 32 bits :

- $n = 2048$ et $\log_2(q) = 95$: 1.824 s $\rightarrow$ 1.634 s.
- $n = 4096$ et $\log_2(q) = 190$: 7,872 s $\rightarrow$ 6,608 s.
- $n = 8192$ et $\log_2(q) = 390$: 35,419 s $\rightarrow$ 26,894 s.
- $n = 16384$ et $\log_2(q) = 790$: 188.657 s $\rightarrow$ 105,651 s.

Le passage au RNS n'est pas suffisant, autre chose domine le calcul.

Transformée de Fourier Discrète

La transformée de Fourier Discrète d'un polynôme $\boldsymbol{a} \in \mathbb{C}[X]$ de degré n est l'évaluation de $\boldsymbol{a}$ en les racines n-ième de l'unité $(\omega_i)_{i=1}^{n}$.

$$\begin{array}{rccc} TFD : & \mathbb{C}_n[X] & \longrightarrow & \mathbb{C}^n \\ & (a_0, \cdots, a_n) & \mapsto & \left(\sum_{j=0}^{n-1} a_j \omega_i^j \right)_{i=1}^{n} \end{array}$$

- C'est une application bijective, que l'on peut calculer (ainsi que son inverse) en $\mathcal{O}(n \log_2(n))$ grâce à l'algorithme de Cooley-Tukey.

  $\rightarrow$ n doit être une puissance de 2.
- Polynômes représentés en TFD $\Longrightarrow$ multiplication linéaire en n.
- Une fois en TFD, l'arithmétique se parallélise indépendemment sur les n composantes.

Transformée de Fourier Discrète

Multiplier deux polynômes $\boldsymbol{a}$ et $\boldsymbol{b}$ de degré $n-1$:

- Représenter les deux polynômes avec $2n$ coefficents (avec n zéros).
- Calculer la TFD de $\boldsymbol{a}$ et $\boldsymbol{b}$ avec les racines $2n$-ièmes de l'unité.
- Calculer $TFD(\boldsymbol{c}) = TFD(\boldsymbol{a}) \times TFD(\boldsymbol{b})$ coordonnées par coordonnées.
- Calculer la TFD inverse de $TFD(\boldsymbol{c})$.

$$\text{Côut total : } TFD + \text{Multiplication} + TFD^{-1} \rightarrow \mathcal{O}(n \log_2(n))$$

Et dans les corps de nombres ?

Number Theoretic Transform (NTT)

Le principe de la transformée de Fourier Discrète peut se généraliser à tout corps de nombres admettant des racines de l'unité.

Exercice

Soit q un nombre premier, montrer que $q \equiv 1\ [n]$ si et seulement si $\mathbb{F}_q$ admet des racines n-ièmes de l'unité.

Pour utiliser la NTT dans $\mathcal{R}_q \cong \mathcal{R}_{q_1} \times \cdots \times \mathcal{R}_{q_k}$ on doit :

- Utiliser des modulis q_i premiers.
- Choisir chaque $q_i \equiv 1\ [2n]$.

Problème

On se retrouve avec un polynôme de degré $2n-2$. Il faut donc utiliser un algorithme de réduction pour le réduire modulo $X^n + 1$.

Algorithme de Réduction Polynômial

Que connaissez-vous ?

- Barrett $\rightarrow \mathcal{O}(n \log_2(n))$
- Montgomerry $\rightarrow \mathcal{O}(n \log_2(n))$

La complexité de la multiplication dans $\mathcal{R}_q$ reste en $\mathcal{O}(n \log_2(n))$.

Mais la constante cachée dans le $\mathcal{O}$ augmente...

En fait, dans notre cas, on peut faire la réduction directement...

Remarquons qu'en évaluant les polynômes en les racines $2n$-ièmes de l'unité on calcule le résultat modulo $X^{2n} - 1$.

Retour sur la NTT

Il s'agit de trouver des racines n-ièmes de -1 dans $\mathbb{F}_q$.

Exercice

Soit $\psi \in \mathbb{F}_q$ une racine primitive $2n$-ième de l'unité :

1. Montrer que ψ est une racine primitive n-ième de -1 et que $\omega = \psi^2$ est une racine primitive n-ième de 1.
2. Décrire l'ensemble des racines n-ièmes de -1 de $\mathbb{F}_q$ à l'aide de ψ et ω.
3. Lorsque $q \equiv 1\ [2n]$ montrez, utilisant des NTTs de taille n, comment multiplier deux éléments $\boldsymbol{a}$ et $\boldsymbol{b}$ de $\mathcal{R}_q$.

Application

Soit $\boldsymbol{a} = -X + 2$ et $\boldsymbol{b} = 5X - 3$ deux éléments de $(\mathbb{Z}/13\mathbb{Z})[X]/(X^2 + 1)$. Calculer le produit de $\boldsymbol{a}$ et $\boldsymbol{b}$ en utilisant l'algorithme présenté précedemment ($\psi = 5$, $\psi^{-1} = -5$, $\omega = -1$ et $\omega^{-1} = -1$, $2^{-1} = -6$).

Retour sur la NTT

Il s'agit de trouver des racines n-ièmes de -1 dans $\mathbb{F}_q$.

Exercice

Soit $\psi \in \mathbb{F}_q$ une racine primitive $2n$-ième de l'unité :

1. Montrer que ψ est une racine primitive n-ième de -1 et que $\omega = \psi^2$ est une racine primitive n-ième de 1.
2. Décrire l'ensemble des racines n-ièmes de -1 de $\mathbb{F}_q$ à l'aide de ψ et ω.
3. Lorsque $q \equiv 1\ [2n]$ montrez, utilisant des NTTs de taille n, comment multiplier deux éléments $\boldsymbol{a}$ et $\boldsymbol{b}$ de $\mathcal{R}_q$.

Solution

- $f_\psi(\boldsymbol{a}) = (2, -5)$ et $f_\psi(\boldsymbol{b}) = (-3, -1)$
- $NTT(f_\psi(\boldsymbol{a})) = (-3, -6)$ et $NTT(f_\psi(\boldsymbol{a})) = (-4, -2)$
- $NTT(f_\psi(\boldsymbol{c})) = (-1, -1)$
- $f_\psi(\boldsymbol{c}) = (-1, 0) \implies f_{\psi^{-1}}(f_\psi(\boldsymbol{c})) = \boldsymbol{c} = (-1, 0) = -1$

Pour résumer

- Accélérer l'arithmétique au niveau des coefficients $\Longrightarrow$ RNS.
  $\rightarrow$ $q = q_1 \cdots q_k$, avec les q_i premiers entre eux de même taille.
- Accélérer l'arithmétique au niveau polynômial $\Longrightarrow$ NTT.
  $\rightarrow$ prendre tous les q_i premiers avec $q_i \equiv 1 \ [2n]$.

Attention !

Certaines opérations ne peuvent pas être effectuées en RNS ou en NTT, pour FV par exemple :

- Déchiffrement : division non exacte par q et arrondis.
- Multiplication : multiplication dans $\mathbb{Z}$, division non exacte par q, arrondis, décomposition en base ω.

$\rightarrow$ Il va falloir naviguer entre les différentes représentations durant les algorithmes...

Conclusion

Tableau des représentations