D34: Méthodes de calcul efficaces et sécurisées

Nicolas Méloni

Master 2: 1er semestre
(2014/2015)

Le corps $\mathbb{F}_{2^n}$

Définition

- $\mathbb{F}_{2^n}$ est le corps à 2^n éléments
- $\mathbb{F}_{2^n} \neq \mathbb{Z}/2^n\mathbb{Z}$
- ex: dans $\mathbb{Z}/2^2\mathbb{Z}$, 4 n'est pas inversible

Représentation

- $\mathbb{F}_{2^n}$ est l'ensemble des polynomes $A[X] \in \mathbb{F}_2[X]$ réduits modulo un certain $P[X]$ irréductible dans $\mathbb{F}_2$
- Un mot machine représente ainsi simplement un polynome: $A[X] = a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \cdots + a_1X + a_0$ par

$$a_{n-1}a_{n-2}\ldots a_1a_0$$

Addition dans $\mathbb{F}_{2^n}$

- Addition bit à bit, sans passage de retenue
- Aucune réduction modulaire

Algorithm 1 Addition dans $\mathbb{F}_{2^n}$

Require: $A[X], B[X] \in \mathbb{F}_{2^n}$
Ensure: $S[X] = A[X] + B[X] \mod P[X]$

1: **for** $i = 0 \dots n-1$ **do**
2: $\quad s_i \leftarrow a_i \oplus b_i$
3: **end for**
4: **return** S

Réduction modulaire dans $\mathbb{F}_{2^n}$

Algorithme d'Euclide

- On suppose que $C[X] = A[X]B[X]$ est un polynome de degré $2n-2$.
- On précalcule les $X_i = X^i \mod P[X]$ pour $n \leq i \leq 2n-2$
- On considère $C_L[X]$ la partie de degré $n-1$ du polynome $C[X]$ et on calcule $C_L[X] \oplus c_n X_n \oplus \cdots \oplus c_{2n-2} X_{2n-2}$

Réduction modulaire dans $\mathbb{F}_{2^n}$

Algorithm 2 Réduction dans $\mathbb{F}_{2^n}$

Require: $C[X]$ de degré $2n-2$, et $\{X_i = X^i \mod P[X], n \leq i \leq 2n-2\}$

Ensure: $R[X] = C[X] \mod P[X]$

1: $R \leftarrow C_L$
2: **for** $i = n \ldots 2n-2$ **do**
3: $\quad R \leftarrow R \oplus c_i X_i$
4: **end for**
5: **return** R

- Possibilité de diminuer le nombre de Xor par plus de précalculs

Multiplication dans $\mathbb{F}_{2^n}$

Algorithm 3 Shift-and-add

Require: $A[X], B[X] \in \mathbb{F}_{2^n}$
Ensure: $C[X] = A[X]B[X] \mod P[X]$

1: $C \leftarrow a_o B$
2: **for** $i = 1 \ldots n-1$ **do**
3: $\quad B \leftarrow BX \mod P$
4: $\quad$ **if** $a_i = 1$ **then**
5: $\quad\quad C \leftarrow C + B$
6: $\quad$ **end if**
7: **end for**
8: **return** C

- L'opération $\mod P$ est un simple Xor

Multiplication dans $\mathbb{F}_{2^n}$

Version matricielle

- $\mathbb{F}_{2^n}$ peut être vu comme un espace vectoriel sur $\mathbb{F}_2$
- Tout élément se décompose alors dans la base $(1, X, \ldots, X^{n-1})$
- On peut décomposer le produit AB de la même manière, pour cela il suffit de connaitre la valeur des $X^i X^j$ dans la base.

- $X^i X^j = \sum_{s=0}^{n-1} \lambda_{ij}(s) X^s$
- $A[X]B[X] = \sum_{s=0}^{n-1} \left(\sum_{0 \leq i,j \leq n-1} a_i b_j \lambda_{ij}(s) \right) X^s$

Multiplication dans $\mathbb{F}_{2^n}$

- Soit c_s la coordonnée d'indice s du produit:

$$c_s = (a_0, a_1, \ldots, a_{n-1}) \underbrace{\begin{pmatrix} \lambda_{00}(s) & \ldots & \lambda_{0,n-1}(s) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \lambda_{n-1,0}(s) & \ldots & \lambda_{n-1,n-1}(s) \end{pmatrix}}_{\text{Matrice de Structure } M_s} \begin{pmatrix} b_0 \\ \vdots \\ \vdots \\ \vdots \\ b_{n-1} \end{pmatrix}$$

- On a ainsi $c_s = A^T M_s B$

Multiplication dans $\mathbb{F}_{2^n}$

- Posons maintenant

$$C = \begin{pmatrix} c_0 \\ \vdots \\ \vdots \\ \vdots \\ c_{n-1} \end{pmatrix} = M_A B \text{ où } M_A = \begin{pmatrix} A^T M_0 \\ \vdots \\ \vdots \\ \vdots \\ A^T M_{n-1} \end{pmatrix}$$

- Le produit entre A et B revient au final à effectuer un produit matrice-vecteur.
- On cherche des bases pour lesquelles la matrice M_A est simple (creuse).

Matrice de Toeplitz

Définition

Une matrice de Toeplitz est une matrice carrée $M = (m_{ij})$ tq: $m_{ij} = m_{i+1,j+1} = m_{i-j}$.

$$M = \begin{pmatrix} m_0 & m_{-1} & \dots & \dots & m_{-(n-1)} \\ m_1 & m_0 & \ddots & \ddots & \vdots \\ m_2 & m_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & m_{-1} \\ m_{n-1} & \dots & m_2 & m_1 & m_0 \end{pmatrix}$$

Matrice de Toeplitz

- Il est toujours possible de transformer une matrice M_A de multiplication dans $\mathbb{F}_{2^n}$ en une matrice de Toeplitz par un changement de base. Par exemple dans $F_2[X]/(X^3+X+1)$:

$$\begin{aligned} M_A &= \begin{pmatrix} a_0 & a_2 & a_1 \\ a_1 & a_0+a_2 & a_1+a_0 \\ a_2 & a_1 & a_0+a_2 \end{pmatrix} \\ &\simeq \begin{pmatrix} a_1 & a_0+a_2 & a_1+a_0 \\ a_2 & a_1 & a_0+a_2 \\ a_0 & a_2 & a_1 \end{pmatrix} \\ &= M'_A \end{aligned}$$

Matrice de Toeplitz

- Lorsque n est divisible par 2 il est possible de tirer partie de la structure pour obtenir un algorithme de multiplication de type Karatsuba. On suppose que M_A est une matrice de Toeplitz:

$$\begin{pmatrix} C_0 \\ C_1 \end{pmatrix} = \begin{pmatrix} A_0 & A_1 \\ A_2 & A_0 \end{pmatrix} \begin{pmatrix} B_0 \\ B_1 \end{pmatrix}$$

$$= \begin{pmatrix} A_0B_0 + A_1B_1 \\ A_2B_0 + A_0B_1 \end{pmatrix}$$

Matrice de Toeplitz

- Posons maintenant:

$$\begin{aligned} T_0 &= A_0(B_0 + B_1) \\ T_1 &= (A_0 + A1)B_1 \\ T_2 &= (A_2 + A_0)B_0 \end{aligned}$$

On a alors:

$$\begin{aligned} C_0 &= T_0 + T_1 \\ C_1 &= T_0 + T_2 \end{aligned}$$

- On a remplacé le produit matrice-vecteur $n^2 \times n$ initial par 3 produits $\left(\frac{n}{2}\right)^2 \times \frac{n}{2}$ où les matrices en jeu sont toujours des matrices de Toeplitz. On obtient au final un algorithme de compléxité sous quadratique ($n^{\log_2(3)}$).

Bases normales

Définition

- Soit $\gamma \in \mathbb{F}_{2^n}, \gamma$ est dit normal si le système $\mathcal{B}(\gamma) = (\gamma, \gamma^2, \ldots, \gamma^{2^{n-1}}$ forme une base de $\mathbb{F}_2^n$ sur $\mathbb{F}_2$. Une telle base est dite base normale.

Intérêt des bases normales

- Soit $A = a_0\gamma + a_1\gamma^2 + \cdots + a_{n-1}\gamma^{2^{n-1}}$ alors $A^2 = a_{n-1}\gamma + a_0\gamma^2 + a_1\gamma^3 + \cdots + a_{n-2}\gamma^{2^{n-1}}$.
- L'élévation au carré est une simple permutation circulaire des coordonnées.

Matrices de structures

- Elles s'obtiennent par décalage des colonnes et des lignes. Si $M(s) = (\lambda_{ij}(s))$ alors: $\lambda_{ij}(s) = \lambda_{i-s,j-s}(0)$.
- Il suffit de calculer la matrice M_0 pour obtenir toutes les matrices de structures.

Algorithm 4 Multiplication en base normale

Require: $A = (a_0, \ldots, a_{n-1}), B = (b_0, \ldots, b_{n-1}), M_0 = (\lambda_{i,j}(0))_{i,j=0..n-1}$

1: **for** $s = 0 \ldots n-1$ **do**
2: $\quad c_s \leftarrow (A^T)^{2^{n-s}} M_0 B^{2^{n-s}}$
3: **end for**
4: **return** $C = (c_0, c_1, \ldots, c_{n-1})$