

# Opérations sur les corps finis

Jean-Claude Bajard

GRT AI

LIRMM UMR CNRS 5506 - université Montpellier 2

# Présentation

- Koblitz en 1987 [Kob87] introduit l'utilisation des courbes elliptiques sur des corps finis en cryptographie.
- Groupe des Points d'une courbe à coordonnées dans un corps finis
- Addition : Intersection d'une droite passant par deux points de la courbe avec la courbe
- Opérations sur le corps : addition, multiplication, division.

## Des généralités [LN85]

Un corps fini  $F(+, \times)$  est un ensemble fini  $F$  muni de deux lois internes

- $F(+)$  est un groupe abélien
- $F(+, \times)$  est un anneau où tout élément (sauf 0 pour  $\times$ ) admet un inverse

Les corps finis les plus élémentaires sont ceux dont le cardinal est un nombre premier  $p$ .

Le représentant le plus connu est le corps  $\mathbb{Z}/p\mathbb{Z}$

$$\mathbb{Z}/p\mathbb{Z} = \{0, 1, 2, \dots, p - 1\}$$

Le calcul sur ces corps utilise l'arithmétique modulaire classique.

## Extension de corps finis

Les autres corps finis sont de la forme  $GF(p^m)$  avec  $p$  premier.  $p$  est la caractéristique, si  $u \in GF(p^m)$  alors  $p \times u = 0$ .

- soit l'ensemble des restes des polynômes à coefficients dans  $GF(p)$  modulo un polynôme irréductible  $P[X]$  de degré  $m$  (les calculs se traduisent par des opérations modulaires sur les polynômes)
- soit l'ensemble des puissances d'un élément primitif
- base canonique  $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$  ou base normale  $\{\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^{m-1}}\}$  où  $\alpha$  racine du polynôme irréductible

Quelle que soit l'approche un élément de  $GF(p^m)$  est donc représenté par un  $m$ -uplet d'éléments de  $GF(p)$ .

## Exemple dans $GF(4)$

- Représentation polynomiale à coefficient dans  $GF(2)$  :  $0, 1, x, 1 + x$ .
- L'addition se fait simplement sur  $GF(2)$  :  $1 + (1 + x) = x$ .
- Par contre pour le produit nous devons faire une réduction liée au polynôme irréductible utilisé pour la représentation des éléments du corps.
  - \*  $x^2 + x + 1$  est irréductible sur  $GF(2)$ ,  $GF(4)$  peut être assimilé à  $GF(2)[x]/x^2 + x + 1$ .
  - \* multiplication modulo le polynôme irréductible  $x^2 + x + 1$ , exemple :  
 $x * (1 + x) = (x + x^2) \bmod (x^2 + x + 1) = 1$
  - \* Le choix du polynôme irréductible n'est donc pas sans conséquence sur la complexité de la multiplication.

## La multiplication dans $GF(2^m)$

Dans la pratique la grande majorité des implantations utilisent des corps de la forme  $GF(2^m)$  où les opérations sur le corps de base  $GF(2)$  se réduisent à des “et” et des “ou exclusif” .

Le produit sur un corps fini comprend une réduction modulaire qui donne lieu à deux types d’approches :

- celles qui ne dépendent pas du corps choisi
- celles qui au contraire exploitent les particularités de ce dernier

## La multiplication dans $GF(2^m)$

Le calcul de  $C = A \times B \bmod P$  peut se faire en deux temps :

1. un produit de polynômes  $C'[X] = A[X] \times B[X]$ ,

$$\begin{pmatrix} c'_0 \\ c'_1 \\ \dots \\ c'_{m-1} \\ c'_m \\ \dots \\ c'_{2m-2} \end{pmatrix} = \begin{pmatrix} a_0 & 0 & \dots & 0 & 0 \\ a_1 & a_0 & 0 & 0 & 0 \\ & & \dots & & \\ a_{m-1} & & & a_1 & a_0 \\ 0 & a_{m-1} & & & a_1 \\ & & \dots & & \\ 0 & 0 & & 0 & a_{m-1} \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \\ \dots \\ b_{m-1} \end{pmatrix}$$

2. une réduction modulo  $P[X] : C[X] = C'[X] \bmod P[X]$

# Algorithme de Montgomery

L'algorithme de Montgomery évalue le produit  $A(x) * B(x)$  dans  $GF(2^m)$  définie par  $P(x)$  un polynôme irréductible de degré  $m$ ,

autrement dit de calculer  $A(x) * B(x) \text{ mod } P(x)$ .

Une première exécution de cet algorithme évalue  $A(x) * B(x) * R^{-1}(x) \text{ mod } P(x)$  où  $R(x)$  est un élément fixé et  $R^{-1}(x)$  représente son inverse  $\text{ mod } P(x)$ .

Connaissant  $R(x)$ , comme  $P(x)$  est irréductible il est possible de déterminer par avance  $R^{-1}(x)$  et  $P'(x)$  tels que :

$$R^{-1}(x) * R(x) + P'(x) * P(x) = 1$$

## Algorithme de Montgomery (cas général)

Entrées :  $A(x)$ , et  $B(x)$  deux polynômes de degré inférieur à  $m$

Résultat :  $T(x) = A(x) * B(x) * R^{-1}(x) \bmod P(x)$

Données  $P'(x)$ ,  $R(x)$

étape 1 :  $C(x) = A(x) * B(x)$

étape 2 :  $Q(x) = C(x) * P'(x) \bmod R(x)$

étape 3 :  $T(x) = (C(x) + Q(x) * P(x)) \operatorname{div} R(x)$

La complexité de cet algorithme est du au trois produits de polynômes.

La réduction modulo  $R(x)$  et la division par  $R(x)$  sont très simples si  $R(x) = x^m$ .

## Algorithme de Montgomery (cas général)

Nous avons,

$$\begin{aligned}A(X) &= a_0 + a_1X + a_2X^2 + \dots + a_{m-1}X^{m-1} \\B(X) &= b_0 + b_1X + b_2X^2 + \dots + b_{m-1}X^{m-1} \\P(X) &= p_0 + p_1X + p_2X^2 + \dots + p_{m-1}X^{m-1} + X^m \\P'(X) &= p'_0 + p'_1X + p'_2X^2 + \dots + p'_{m-1}X^{m-1}\end{aligned}$$

## Algorithme de Montgomery (cas général)

Décomposition du calcul de  $Q(X)$  :

$$Q(X) = - \begin{pmatrix} p'_0 & 0 & \dots & 0 & 0 \\ p'_1 & p'_0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ p'_{m-2} & p'_{m-3} & \dots & p'_0 & 0 \\ p'_{m-1} & p'_{m-2} & \dots & p'_1 & p'_0 \end{pmatrix} \begin{pmatrix} a_0 & 0 & \dots & 0 & 0 \\ a_1 & a_0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m-2} & a_{m-3} & \dots & a_0 & 0 \\ a_{m-1} & a_{m-2} & \dots & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \dots \\ b_{m-2} \\ b_{m-1} \end{pmatrix}$$

# Algorithme de Montgomery (cas général)

Décomposition du calcul de  $R(X)$  :

$$\begin{pmatrix} 0 & a_{m-1} & \dots & a_2 & a_1 \\ 0 & 0 & \dots & a_2 & a_1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{m-1} & a_{m-2} \\ 0 & 0 & \dots & 0 & a_{m-1} \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \dots \\ b_{m-3} \\ b_{m-2} \\ b_{m-1} \end{pmatrix} + \begin{pmatrix} 1 & p_{m-1} & \dots & p_2 & p_1 \\ 0 & 1 & \dots & n_2 & n_1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & p_{m-1} & p_{m-2} \\ 0 & 0 & \dots & 1 & p_{m-1} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \dots \\ q_{m-3} \\ q_{m-2} \\ q_{m-1} \end{pmatrix}$$

# Algorithme de Montgomery (cas général)

Complexité en nombre d'opérations élémentaires sur  $GF(p)$ :

- $m^2 + (m - 1)^2$  multiplications
- $(m - 1)^2 + (m - 2)^2 + m$  additions.

Inconvénients:

- les matrices dépendent des entrées, ici  $A[X]$
- produits modulaires
- initialisation du circuit

## Algorithme de Montgomery itératif avec $R(x) = x^m$ .

Entrées :  $A(x)$ , et  $B(x)$  deux polynômes de degré inférieur à  $m$

Résultat :  $T(x) = A(x) * B(x) * R^{-1}(x) \text{ mod } P(x)$

Données  $P'(x)$ ,  $R(x) = x^m$

Initialisation  $T(x) = 0$

Boucle pour  $i = 0$  à  $m - 1$

$$T(x) = T(x) + a_i * B(x)$$

$$T(x) = (T(x) + t_0 * P(x)) / x$$

A chaque itération réduction modulo  $x$  d'où au final réduction par  $R(x) = x^m$ .

De plus,  $P(x)$  est irréductible donc son terme de degré 0 vaut 1, idem pour  $P'(x)$ .

La complexité en nombre de portes logiques : soit  $2m$  *xor* (pour les sommes) et  $2m + 1$  *and* (pour les produits).

## Méthode de Mastrovito

$GF(2^m)$  est définie par une racine  $\alpha$  d'un un polynôme irréductible  $P(x)$  de degré  $m$ .

Les éléments de  $GF(2^m)$  sont écrit dans la base canonique  $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$  :

$$A = \sum_{i=0}^{m-1} a_i \alpha^i \quad \text{et} \quad B = \sum_{i=0}^{m-1} b_i \alpha^i.$$

Le produit  $A \times B$  dans  $GF(2^m)$  est noté  $C = \sum_{i=0}^{m-1} c_i \alpha^i$ .

Mastrovito propose dans sa thèse de construire  $Z$  une matrice carrée  $m \times m$  dépendant de  $A$  telle que :

$$C = Z \times B$$

## Méthode de Mastrovito

Mastrovito donne un algorithme de construction de  $Z$  :

1. on construit la matrice  $(m-1) \times m$ ,  $Q$  correspondant à l'écriture des  $X^k$  pour  $k \geq m$  modulo  $P[X]$ :

$$\begin{pmatrix} X^m \\ X^{m+1} \\ \dots \\ X^{2m-2} \end{pmatrix} = Q \times \begin{pmatrix} X^0 \\ X^1 \\ \dots \\ X^{m-1} \end{pmatrix}$$

2. puis la matrice  $Z$  où :

$$z_{i,j} = \begin{cases} a_i \text{ pour } j = 0, i = 0 \dots m-1 \\ u(i-j) * a_{i-j} + \sum_{t=0}^{j-1} q_{j-1-t,i} * a_{m-1-t}, \text{ sinon} \end{cases}, \text{ où } u(t) = \begin{cases} 1 \text{ si } t \geq 0 \\ 0 \text{ sinon} \end{cases}$$

# Méthode de Mastrovito

La complexité de cette méthode vient en partie de la construction de la matrice  $Z$  qui peut nécessiter  $m^3/2$  *And* et *Xor*, le choix du polynôme est donc fondamental.

Avec des polynômes de la forme  $x^m + x + 1$  le produit peut se faire avec  $m^2 - 1$  *Xor* et  $m^2$  *And*.

Il existe des variantes pour des polynômes composés que de 1 (all-one polynomial)  $P(x) = 1 + x + x^2 + \dots + x^m$  ou encore régulièrement espacé  $P(x) = 1 + x^\Delta + x^{2\Delta} + \dots + x^{k\Delta=m}$ .

## Méthode de Mastrovito

Exemple : pour  $P(x) = 1 + x + x^2 + \dots + x^m$  la matrice  $Z$  peut se décomposer sous la forme  $Z = Z_1 + Z_2$  avec :

$$Z_1 = \begin{pmatrix} a_0 & 0 & a_{m-1} & \dots & a_3 & a_2 \\ a_1 & a_0 & 0 & a_{m-1} & a_4 & a_3 \\ & & & \dots & & \\ & & & \dots & & \\ a_{m-2} & a_{m-3} & & & a_0 & 0 \\ a_{m-1} & a_{m-2} & & & a_1 & a_0 \end{pmatrix}$$

et

$$Z_2 = \begin{pmatrix} 0 & a_{m-1} & a_{m-2} & & a_1 \\ 0 & a_{m-1} & a_{m-2} & & a_1 \\ & & & \dots & \\ 0 & a_{m-1} & a_{m-2} & & a_1 \end{pmatrix}$$

## Base normale

Les éléments de  $GF(2^m)$  sont décrits dans une base normale  $\{\alpha, \alpha^2, \alpha^{2^2}, \dots, \alpha^{2^{m-1}}\}$  définie par une racine  $\alpha$  d'un polynôme  $P(x)$  de degré  $m$ .

Soit  $A$  un élément de  $GF(2^m)$  :

$$A = (a_0, a_1, \dots, a_{m-1}) = \sum_{i=0}^{m-1} a_i \alpha^{2^i}.$$

L'élevation au carré sous cette représentation se réduit à une simple permutation circulaire :

$$\text{nous avons } A^2 = \sum_{i=0}^{m-1} a_i \alpha^{2^{i+1}} \text{ or } \alpha^{2^m} = \alpha,$$

$$\text{donc, } A^2 = a_{m-1} \alpha + \sum_{i=1}^{m-1} a_{i-1} \alpha^{2^i} \text{ autrement dit } A^2 = (a_{m-1}, a_0, \dots, a_{m-2}).$$

## Base normale : Massey et Omura 1986

Produit dans  $GF(2^m)$  :  $D = A \times B = A \times M \times B^t$  où  $M$  est la matrice :

$$M = \begin{pmatrix} \alpha^{2^0+2^0} & \alpha^{2^0+2^1} & \dots & \alpha^{2^0+2^j} & \dots & \alpha^{2^0+2^{m-2}} & \alpha^{2^0+2^{m-1}} \\ \alpha^{2^1+2^0} & \alpha^{2^1+2^1} & \dots & \alpha^{2^1+2^j} & \dots & \alpha^{2^1+2^{m-2}} & \alpha^{2^1+2^{m-1}} \\ \alpha^{2^i+2^0} & \alpha^{2^i+2^1} & \dots & \alpha^{2^i+2^j} & \dots & \alpha^{2^i+2^{m-2}} & \alpha^{2^i+2^{m-1}} \\ \alpha^{2^{m-1}+2^0} & \alpha^{2^{m-1}+2^1} & \dots & \alpha^{2^{m-1}+2^j} & \dots & \alpha^{2^{m-1}+2^{m-2}} & \alpha^{2^{m-1}+2^{m-1}} \end{pmatrix}$$

$M = M_0 \alpha + M_1 \alpha^2 + \dots + M_{m-1} \alpha^{2^{m-1}}$  où  $M_i$  composée de 0 et de 1.

Ainsi  $D = A \times B$  s'obtient en évaluant  $d_{m-1-k} = A \times M_{m-1-k} \times B^t$  pour  $k = 0, \dots, m-1$ .

## Base normale : Massey et Omura

$D^{2^k} = A^{2^k} \times B^{2^k}$  et l'élevation à l'exposant  $2^k$  revient à  $k$  décalages circulaires :

$$d_{m-1-k} = A^{2^k} \times M_{m-1} \times (B^{2^k})^t \text{ pour } k = 0, \dots, m-1$$

La complexité dépend du nombre de 1 dans  $M_{m-1}$ , donc du choix de  $m$  et de  $P(x)$ .

Ce nombre est minoré par  $2m - 1$

Lorsque cette borne est atteinte la base est dite optimale

Si  $P(x)$  de degré  $m$  a tous ses coefficients à 1, cette borne est atteinte et la complexité est  $m^2$  *And* et  $2m^2 - 2m$  *Xor*.

## Base normale : Massey et Omura modifié HWB 1993

Cette complexité peut être réduite à  $m^2$  portes *And* et  $m^2 - 1$  portes *Xor*, en décomposant la matrice  $M_{m-1}$

$$M_{m-1} = P + Q(\text{mod } 2) \text{ avec } P_{i,j} = \begin{cases} 1 & \text{si } i = (m/2 + j) \text{ mod } m \\ 0 & \text{sinon} \end{cases}$$

Dans ce cas, si on pose  $T^{(k)}$  telle que :  $B^{2^k} = BT^{(k)}$ , nous remarquons que  $T^{(k)}PT^{(k)} = P$ . Nous obtenons ainsi

$$d_{m-1-k} = A \times P \times B^t + A^{2^k} \times Q \times (B^{2^k})^t \text{ pour } k = 0, \dots, m-1$$

## Bases duales dans $GF(p^m)$

Fonction Trace :  $Tr(u) = \sum_{i=0}^{m-1} u^{p^i}$

Bases duales : deux bases  $\{\lambda_i, i = 0..m - 1\}$  et  $\{\nu_j, j = 0..m - 1\}$  sont duales si  $Tr(\lambda_i \cdot \nu_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

Changement de base :  $Tr(\nu_j \cdot x) = x_j$  où  $x_j$  avec  $x = \sum_{j=0}^{m-1} x_j \lambda_j$

## Bases duales dans $GF(p^m)$ (suite)

Fonction linéaire :  $f(u) = Tr(\beta.u)$  où  $\beta \in GF(p^k)$

Bases duales si  $Tr(\beta.\lambda_i.\nu_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

Changement de base :  $Tr(\beta.\lambda_j.x) = x_j$  où  $x_j$  avec  $x = \sum_{j=0}^{m-1} x_j \nu_j$

## Multiplication avec les Bases duales dans $GF(p^m)$

On considère ici la base canonique  $\{\alpha^i, i = 0..m - 1\}$  et une base duale par  $(f, \beta)$   
 Soient  $a$   $b$  et  $c$  dans  $GF(p^m)$

$$\begin{pmatrix} \text{Tr}(b\beta) & \text{Tr}(b\beta\alpha) & \dots & \text{Tr}(b\beta\alpha^{m-1}) \\ \text{Tr}(b\beta\alpha) & \text{Tr}(b\beta\alpha^2) & \dots & \text{Tr}(b\beta\alpha^m) \\ \dots & \dots & \dots & \dots \\ \text{Tr}(b\beta\alpha^{m-1}) & \text{Tr}(b\beta\alpha^2) & \dots & \text{Tr}(b\beta\alpha^{2m-2}) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \dots \\ c_{m-1} \end{pmatrix} = \begin{pmatrix} \text{Tr}(a\beta) \\ \text{Tr}(a\beta\alpha) \\ \dots \\ \text{Tr}(a\beta\alpha^{m-1}) \end{pmatrix}$$

la première ligne correspond au coordonnées de  $b$  dans la base duale, de même la colonne pour  $a$

but : trouver une fonction  $f$  simple où la base duale est une simple permutation de la base canonique [Ber82, STFT96, HWB98, Gol02]

# la division

Cette opération dans un corps se traduit par le produit par l'inverse.

Nous présentons ici deux façon d'obtenir l'inverse d'un éléments dans un corps fini.

La première méthode est générique pour tous les corps. La seconde est spécifique aux corps finis.

## Avec l'algorithme d'Euclide

Basé sur l'algorithme d'Euclide du calcul du pgcd.

Nous désirons donc calculer l'inverse de  $a$  modulo  $b$ . Nous définissons trois triplets  $U = (u_1, u_2, u_3)$ ,  $V = (v_1, v_2, v_3)$  et  $T = (t_1, t_2, t_3)$  tels que <sup>1</sup>:

$$\begin{aligned}u_1b + u_2a &= u_3 \\v_1b + v_2a &= v_3 \\t_1b + t_2a &= t_3\end{aligned}$$

En réalité les termes d'indice 2 sont inutiles pour le calcul de l'inverse et ne sont donc pas implantés. En terme de complexité cet algorithme est en  $O(k)$ , (à chaque pas le degré le plus haut diminue au moins de un).

---

<sup>1</sup>Soient  $a$  et  $b$  deux entiers relatifs non nul et  $d$  leur PGCD. Il existe deux entiers relatifs  $u$  et  $v$  vérifiant l'égalité de BEZOUT  $au + bv = d$ .

## Initialisation

$$\begin{array}{lll} u_1 \leftarrow 1 & u_2 \leftarrow 0 & u_3 \leftarrow b \\ v_1 \leftarrow 0 & v_2 \leftarrow 1 & v_3 \leftarrow a \end{array}$$

## Boucle principale

while  $v_3 \neq 0$

$$n = \deg(u_3) - \deg(v_3)$$

$$t_1 \leftarrow u_1 - x^n v_1$$

$$t_2 \leftarrow u_2 - x^n v_2 \quad t_3 \leftarrow u_3 - x^n v_3$$

If  $\deg(t_3) \geq \deg(v_3)$

$$u_1 \leftarrow t_1 \quad u_2 \leftarrow t_2 \quad u_3 \leftarrow t_3$$

then

$$u_1 \leftarrow v_1 \quad u_2 \leftarrow v_2 \quad u_3 \leftarrow v_3$$

$$v_1 \leftarrow t_1 \quad v_2 \leftarrow t_2 \quad v_3 \leftarrow t_3$$

Résultat  $u_2 \sim a^{-1} \pmod{b}$

## Par le théorème de Fermat-Euler

L'énoncé du théorème de Fermat est le suivant : tout élément  $\beta$  d'un corps fini d'ordre  $p^m$  satisfait :  $\beta^{p^m} = \beta$ , autrement dit  $\beta$  est racine de  $x^{p^m} = x$  ce qui se traduit aussi par,

$$x^{p^m} - x = \prod_{\beta \in GF(p^m)} (x - \beta)$$

On en déduit ainsi dans  $GF(2^m)$  que l'inverse est égal à  $\beta^{-1} = \beta^{2^m-2}$ .

L'algorithme d'exponentiation utilise une stratégie de produits et carrés liée à l'écriture binaire de l'exposant. Vu la forme de ce dernier dans le cas présent, de nombreuses astuces permettent d'accélérer le calcul [TYT01]. La complexité reste malgré tout en  $O(m)$ .

# Conclusion

Les corps de caractéristique 2 ont été largement étudiés

L'intérêt pour des corps finis de caractéristiques différentes de deux commence à prendre de l'ampleur. [BP00, Sma01].

Une des idées est de refaire ce qui a été fait en caractéristique 2 : par exemple l'exploitation des propriétés du Frobenius ce qui a été fait en caractéristique 3 [PS03] .

[]

## References

- [Ber82] E.R. Berlekamp. Bit-serial reed-solomon encoders. *IEEE Trans. Information Theory*, 1982.
- [BP00] Daniel V. Bailey and Christof Paar. Public-key cryptography with arbitrary finite fields. Submission to IEEE P1363a, February 2000.
- [GG90] Joachim Von Zur Gathen and Mark Giesbrecht. Constructing normal bases in finite fields. *Journal of Symbolic Computation*, 10(6):547–570, December 1990.
- [GL92] Shuhong Gao and Hendrik W. Lenstra, Jr. Optimal normal bases. *Designs, Codes, and Cryptography*, 2(4):315–323, 1992.
- [GL01] Geiselmann and Lukhaub. Redundant representation of finite fields. In *PKC: International Workshop on Practice and Theory in Public Key Cryptography*. LNCS, 2001.

- [Gol02] Dieter Gollmann. Equally spaced polynomials, dual bases, and multiplication in  $\mathbb{F}_{2^n}$ . *IEEE Transactions on Computers*, 2002.
- [GV95] Shuhong Gao and Scott A. Vanstone. On orders of optimal normal basis generators. *Mathematics of Computation*, 64(211):1227–1233, July 1995.
- [HK00] A. Halbutogullari and Cetin Koya Koc. Mastrovito multiplier for general irreducible polynomials. *IEEE Transactions on Computers*, 49(5):503–518, 2000.
- [HWB93] Hasan, Wang, and Bhargava. A modified massey-omura parallel multiplier for a class of finite fields. *IEEETC: IEEE Transactions on Computers*, 42, 1993.
- [HWB98] M. Anwarul Hasan Huapeng Wu and Ian F. Blake. New low-complexity bit-parallel finite field multipliers using weakly dual bases. *IEEE Transactions on Computers*, 1998.
- [KA98] Koc and Acar. Montgomery multiplication in  $GF(2^k)$ . *IJDCC: Designs, Codes and Cryptography*, 14, 1998.

- [KAK96] Cetin Kaya Koc, Golga Acar, and Burton S. Kaliski, Jr. Analyzing and comparing montgomery multiplication algorithms. *IEEE Micro*, June 1996. 26–33.
- [Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.
- [KS98] Koc and Sunar. Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields. *IEEETC: IEEE Transactions on Computers*, 47, 1998.
- [LN85] R. Lidl and H. Niederreiter. *Finite Fields*. Addison-Wesley, Reading, 1985.
- [LN94] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press, revised edition edition, 1994.
- [Mas89] E. D. Mastrovito. VLSI designs for multiplication over finite fields  $GF(2^m)$ . In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 6th International Conference, (AAECC-6)*, pages 297–309, Berlin - Heidelberg - New York, July 1989. Springer.

- [Mas91] E. Mastrovito. *VLSI Architectures for Computation in Galois Fields*. PhD thesis, Linköping University, Dept. Electr. Eng., 1991.
- [Mon85] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, April 1985.
- [PS03] D. Page and N. P. Smart. Hardware implementation of finite fields of characteristic three. In B. S. Kaliski Jr., ?. K. Ko?, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 529–539. Springer-Verlag, February 2003.
- [RMW89] S.A. Vanstone R.C. Mullin, I.M. Onyszchuk and R. Wilson. Optimal normal basis in  $gf(p^m)$ . *Discrete Applied Mathematics*, 1989.
- [Sil99] Silverman. Fast multiplication in finite fields  $GF(2^N)$ . In *CHES: International Workshop on Cryptographic Hardware and Embedded Systems, CHES, LNCS*, 1999.
- [SK99] B. Sunar and C. K. Koc. Mastrovito multiplier for all trinomials. *IEEE Trans. Comput.*, 1999.

- [SK01] B. Sunar and C. K. Koc. An efficient optimal normal basis type ii multiplier. *IEEE Transactions on Computers*, 2001.
- [Sma99] N. P. Smart. Elliptic curves over small fields of odd characteristic. *J. Cryptology*, 12(2):141–151, August 1999.
- [Sma01] N. P. Smart. A comparison of different finite fields for elliptic curve cryptosystems. *Computers and Mathematics with Applications*, 42(1-2):91–100, 2001.
- [STFT96] Mohammed Benaissa Sebastian T.J. Fenn and David Taylor.  $gf(2^m)$  multiplication and division over the dual basis. *IEEE Transactions on Computers*, 1996.
- [STK01] Erkey Savas, Alexandre F. Tenca, and Çetin K. Koç. A scalable and unified multiplier architecture for finite fields  $GF(p)$  and  $GF(2^m)$ . *Lecture Notes in Computer Science*, 1965:277–??, 2001.
- [TYT01] Takagi, Yoshiki, and Takagi. A fast algorithm for multiplicative inversion in

$GF(2^m)$  using normal basis. *IEEETC: IEEE Transactions on Computers*, 50, 2001.

[Wu02] Wu. Bit-parallel finite field multiplier and squarer using polynomial basis. *IEEETC: IEEE Transactions on Computers*, 51, 2002.