Conception et sécurisation d'unités arithmétiques hautes performances pour courbes elliptiques

Soutenance de thèse de Doctorat d'Informatique

Julien Francq

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

LIRMM Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

16 *décembre* 2009

Cryptographie

- La cryptographie est l'étude des techniques pouvant permettre de sécuriser des opérations en présence de personnes potentiellement malveillantes (attaquants) :
  - Chiffrement
  - Signature numérique
  - Authentification
  - Vérification d'intégrité
  - Autres (Internet)

Contexte de la thèse

- Projet : Briques Technologiques pour le Renforcement de la Sécurité (BTRS)
- Partenaires :
  - CEA-LETI
  - Gemalto
  - Smart Packaging Solutions (SPS)
- Thèse effectuée conjointement au :
  - Centre de Microélectronique de Provence-Georges Charpak (CMP-GC)
    - Équipe SAS (Systèmes et Architectures Sécurisées)
  - Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM)
    - Équipe ARITH
- Financement : Fonds Social Européen (FSE)

Cryptographie et courbes elliptiques

- Cryptographie symétrique
  - Même clé pour le chiffrement et le déchiffrement
  - Performante mais nécessite un échange de clé avant utilisation
- Cryptographie asymétrique
  - Paire clé privée / clé publique
  - Pas besoin de partager un secret avant utilisation
- Historique :
  - Diffie et Hellman, 1976
  - Rivest, Shamir et Adleman (RSA), 1977
  - Koblitz et Miller, 1985
    - Cryptographie sur courbes elliptiques (*Elliptic Curve Cryptography*, ECC)
    - ECC-160 $\simeq$ RSA-1024

    $\Longrightarrow$ ECC alternative crédible au RSA (standards NIST et Certicom)

Du point de vue du concepteur de circuits cryptographiques

- Un opérateur (ou unité) arithmétique pour l'ECC doit être :
  - performant
    - temps d'exécution court, débit important, petite surface, consommation électrique faible, etc.
  - sécurisé
    - face à d'éventuelles attaques théoriques et physiques (par observation, par perturbation)
    - implantation de parades (contre-mesures)...
    - ...pouvant diminuer les performances

$\Longrightarrow$ Trouver le meilleur compromis performances/sécurité

Travaux réalisés

- Nouvelle architecture d'unité arithmétique pour l'ECC sur $\mathbb{F}_p$
  - Performances meilleures que la plupart de celles de la littérature
- Protection de cette unité contre les attaques par observation à l'aide de l'état de l'art
  - Solution la plus performante de la littérature
- Protection de cette unité contre les attaques par perturbation...
  - ...à l'aide du principe de la préservation de la parité

Sommaire

Sommaire

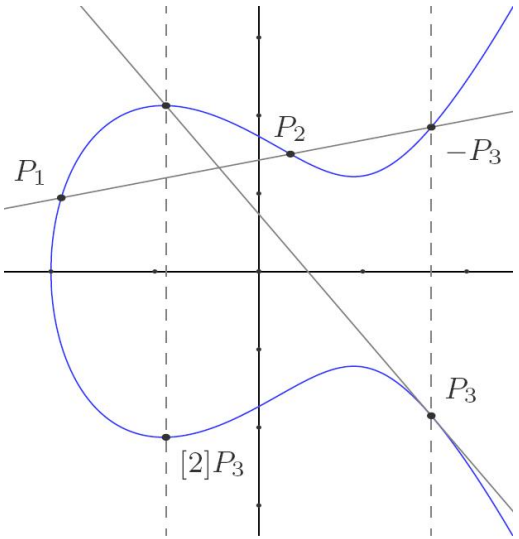
Sommaire

Introduction générale

- Une courbe elliptique E sur $\mathbb{F}_p$ est l'ensemble des points (x, y) obéissant à l'équation simplifiée de Weierstrass :

$$E : y^2 = x^3 + ax + b \cup \{\infty\}$$

- $E(\mathbb{F}_p)$: groupe additif
  - Élément neutre : ∞ (point à l'infini)
  - Opération de groupe : addition de points (+)
    - loi « corde et tangente »



Formules de doublement et d'addition de points

- $P_1(x_1, y_1) + P_2(x_2, y_2)$ donne le point $P_3(x_3, y_3)$, où :

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = (x_1 - x_3)\lambda - y_1 \end{cases}, \text{avec}$$

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{si } x_1 \neq x_2 \text{ [Addition, A]} \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{si } x_1 = x_2 \text{ [Doublement, D]} \end{cases}$$

- Formulation de λ différente pour l'addition et le doublement
- L'addition et le doublement entraînent donc le calcul d'additions, de soustractions, de multiplications, de carrés et d'inversions sur $\mathbb{F}_p$

Multiplication scalaire et ECDLP

- La sécurité de l'ECC repose notamment sur un sous-groupe cyclique de $E(\mathbb{F}_p)$ noté $\mathbb{G}$ généré par le point de base P et d'ordre n grand :

$$\mathbb{G} = \langle P \rangle = \{\infty, P, [2]P, \cdots, [n-1]P\} \subseteq E(\mathbb{F}_p), \text{ avec } [n]P = \infty$$

- Multiplication scalaire de P par un grand entier k (la clé) :

$$Q = [k]P = \underbrace{P + P + \cdots + P}_{k \text{ fois}}$$

  - Opération importante dans les protocoles ECC
  - Opération « à sens unique » : connaissant k et $P \in \mathbb{G}$, facile de calculer $Q = [k]P \in \mathbb{G}$, mais connaissant P et $[k]P$, difficile de retrouver k

→ Problème du logarithme discret sur courbes elliptiques (*Elliptic Curve Discrete Logarithm Problem*, ECDLP)

Multiplication scalaire avec l'algorithme « doublement-et-addition »

Entrées : $P \in E$, $k = \sum_{i=0}^{\ell} k_i 2^i = (k_{\ell-1} \cdots k_0)_2$.

Sortie : $Q = [k]P$.

1. $Q \leftarrow \infty$
2. **pour** $i = \ell - 1$ à 0
3. $\quad Q \leftarrow [2]Q$ [Doublement]
4. $\quad$ **si** $k_i = 1$
5. $\quad\quad Q \leftarrow Q + P$ [Addition]
6. $\quad$ **fin si**
7. **fin pour**
8. **retourner** Q

- Nombre moyen de bits non-nuls dans k : $\ell/2$

$\Longrightarrow$ Nombre d'opérations de points : ℓD + $(\ell/2)$A

Exemples d'optimisations classiques

- Recodage de k sous une forme non-adjacente (NAF)

$$\mathrm{NAF}_w(k) = \sum_{i=0}^{\ell} k_i 2^i, \text{ avec } |k_i| < 2^{w-1}$$

  - Exemple : $k = 763$

$$\begin{array}{rcccccccccc} k_2 & = & (& 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1) \\ \mathrm{NAF}_2(k) & = & (1 & 0 & \overline{1} & 0 & 0 & 0 & 0 & 0 & \overline{1} & 0 & \overline{1}) \end{array}$$

  - Nombre moyen de bits non-nuls dans $\mathrm{NAF}_w(k)$: $\ell/(w+1)$

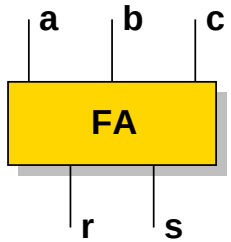
$\Longrightarrow$ Nombre d'opérations de points : $(\ell-1)\mathrm{D} + (\ell/(w+1))\mathrm{A}$

- Autres voies d'amélioration : chaînes d'additions spéciales, courbes particulières (Montgomery, Edwards), etc.

Sommaire

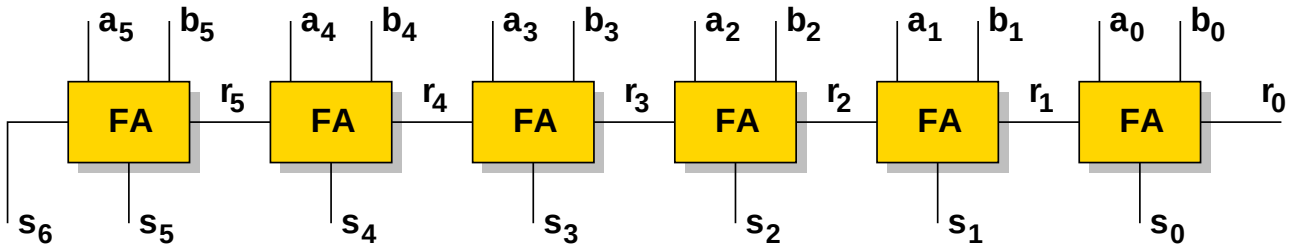
Addition classique

- Cellule d'addition complète (*Full-Adder*, FA) :



$$\begin{cases} s = a \oplus b \oplus c \\ r = ab + ac + bc \end{cases}$$

- Additionneur à propagation séquentielle de la retenue

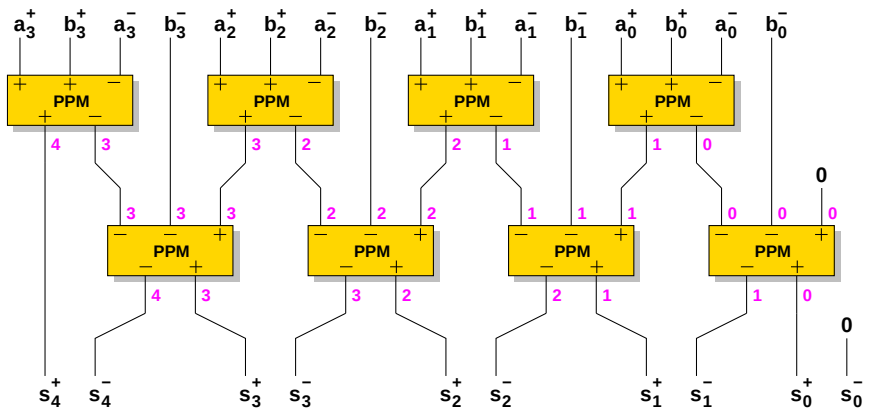


- Si opérandes sur n bits

$\Longrightarrow$ Temps de calcul pire cas $= n \times \mathrm{T(FA)}$

Addition en représentation redondante

- Représentation redondante à retenues conservées : *Carry-Save* (CS), *Borrow-Save* (BS)
- BS : $X = (x_{n-1} \cdots x_1 x_0)_{\mathrm{BS}} = \sum_{i=0}^{n-1} x_i 2^i = \sum_{i=0}^{n-1} (x_i^+ - x_i^-) 2^i$



$\Longrightarrow$ Temps de calcul de l'addition BS (BSA) indépendant de n ($\mathrm{T}(\mathrm{BSA}(n)) = 2 \times \mathrm{T}(\mathrm{PPM}) \approx 2 \times \mathrm{T}(\mathrm{FA})$)

- Mais comparaison et signe difficiles, coût mémoire $2\times$ grand

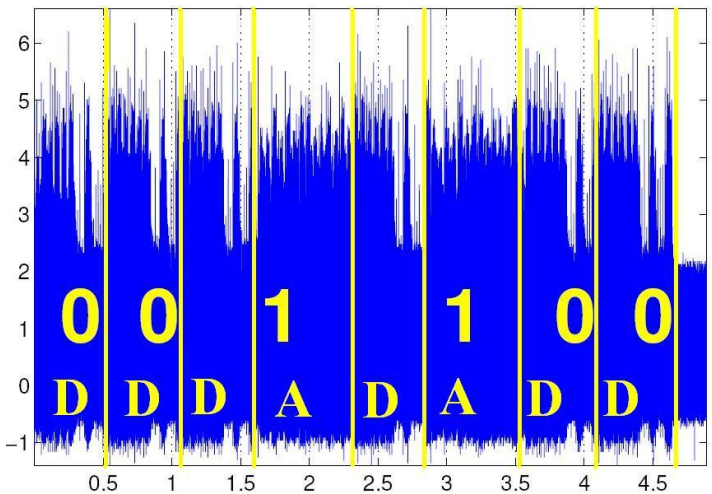
Calculs sur $\mathbb{F}_p$

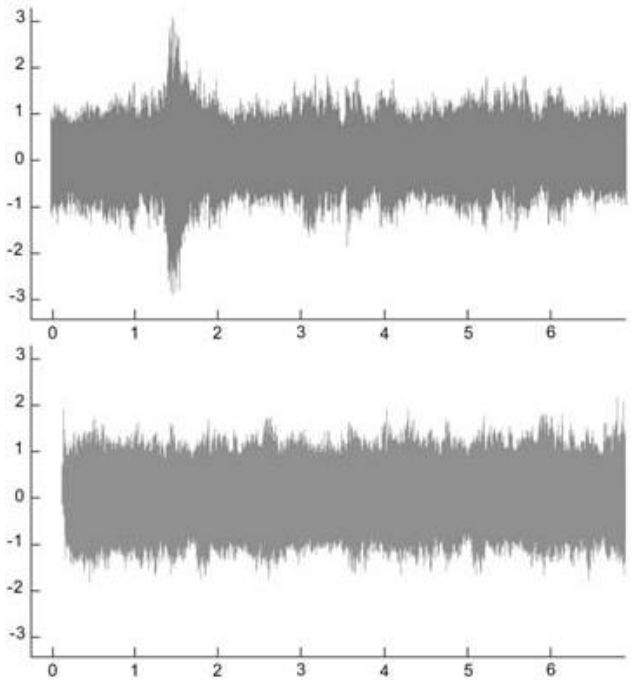
- L'ECC a besoin de trois opérations modulaires :
  - Addition
    - Méthode classique
    - Méthode d'Omura
    - Méthode de Takagi
  - Multiplication
    - Multiplication puis réduction
    - Multiplication et réduction croisées (Montgomery)
  - Inversion
    - Petit théorème de Fermat
    - PGCD
    - Montgomery

Sommaire

Attaques par observation

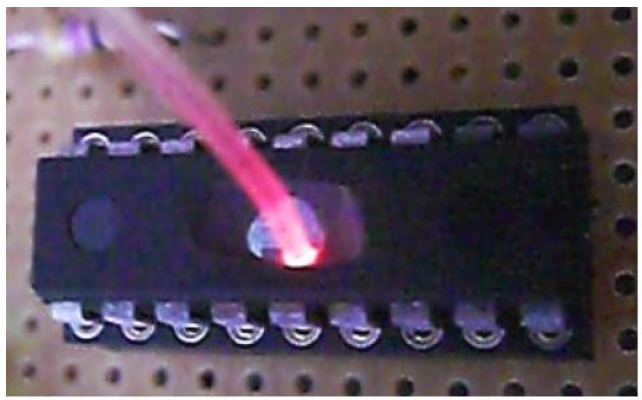
- Mesurer des paramètres externes (canaux cachés : consommation électrique, rayonnement électromagnétique (EM), temps de calcul, etc.) pour récupérer des informations internes
- Mesures globales (consommation électrique) / locales (EM)
- Attaques simples / différentielles





Attaques par perturbation

- Modifier l'environnement du cryptosystème de telle sorte que son fonctionnement soit altéré et qu'il apparaisse des fautes
  - Pics de courant sur :
    - l'alimentation
    - l'horloge
  - Variations :
    - de la température
    - ou du champ EM environnant
  - Injection de particules :
    - rayons X
    - ions lourds
    - photons (laser)



Perturbations : modèles et conséquences sur l'ECC

- Degré d'invasivité
- Degré de destructivité (Fautes permanentes / transitoires)
- Modèles de perturbation :
  - Localisation temporelle/spatiale
  - Nombre de bits fautés
  - Modèle de faute injectée : collage, inversion, etc.
- Pour l'ECC :
  - Forcer le calcul de la multiplication scalaire sur une autre courbe où l'ECDLP est plus facile à calculer
  - Obtenir un point résultat faux mais appartenant à la courbe initiale
  - Attaques initialement mises en évidence sur le RSA

Sommaire

Contre-mesures vis-à-vis des attaques par observation simples

- Un cryptosystème ECC est vulnérable vis-à-vis des attaques par observation simples si :
  - les formules utilisées pour l'addition et le doublement sont différentes
  - l'exécution de la multiplication scalaire dépend de la clé
- D'où 2 contre-mesures possibles :
  - Rendre le doublement et l'addition de points indistinguables (formules unifiées d'addition de points, atomicité)
  - Utiliser un algorithme de multiplication scalaire régulier (« Doublement-et-toujours-addition », échelle de Montgomery, chaînes d'additions, atomicité, recodage de k, « doublement-addition », « addition-seulement »)

Détecter une perturbation

- Vérifier que le résultat est bien sur E

$$(y^2 - x^3 - ax) \stackrel{?}{=} b \pmod{p}$$

- Attaques conservant la courbe initiale
  - = Attaques différentielles

  $\Longrightarrow$ Changer la représentation de k à chaque exécution de la multiplication scalaire
    - $Q = [k^*]P = [k + rn]P$ (avec $[n]P = \infty$)
    - $Q = [k]P = [k - r]P + [r]P$
  - [Blömer, 2006] : effectuer $[k]P$ modulo p_0 ($<p$) et modulo $p_0 p$, puis vérifier que les deux points résultats sont égaux modulo p_0
- La politique adoptée en cas de détection ne doit pas donner d'information à l'attaquant

Sommaire

Sommaire

Modulo choisi

- Version initiale de la multiplication modulaire de Montgomery contient une soustraction conditionnelle
- Vulnérable face à des attaques par observation

$\Longrightarrow$ Algorithme de Montgomery sans soustraction conditionnelle

Entrées : $p = (p_{n-1} \cdots p_1 p_0)_2$, $\text{pgcd}(p, 2) = 1$,
$A = (a_n a_{n-1} \cdots a_1 a_0)_2 < 2p$, $B = (b_n b_{n-1} \cdots b_1 b_0)_2 < 2p$.
Sortie : $S = \text{MMM}(A, B, 2p) = A.B.2^{-(n+2)} \pmod{2p}$.

1. $S \leftarrow 0$
2. **pour** $i = 0$ à $n + 1$
3. $\quad m_i \leftarrow s_0 \oplus a_i.b_0$
4. $\quad S \leftarrow (S + a_i.B + m_i.p)/2$
5. **fin pour**
6. **retourner** S

$\Longrightarrow$ Modulo $2p$

Choix de la représentation des nombres utilisée

- Multiplication modulaire de Montgomery conduit au calcul d'additions/décalages
- Inversion modulaire conduit au calcul de multiplications et/ou d'additions
- Besoin d'une addition rapide

$\Longrightarrow$ Représentation redondante pour une addition rapide (CS ou BS)

- Or, CS très utilisée dans la littérature et BS peu (voir pas du tout)

$\Longrightarrow$ Explorer les possibilités offertes par BS

Opérations que l'unité arithmétique doit pouvoir effectuer

- Opérations pour les calculs en coordonnées affines
  - Addition modulaire : $A + B \pmod{2p}$, avec $A \neq B$
  - Soustraction modulaire : $A - B \pmod{2p}$
  - Multiplication modulaire : $A \times B \pmod{2p}$, avec $A \neq B$
  - Carré modulaire : $A^2 \pmod{2p}$
  - Inversion modulaire : $1/X \pmod{2p}$
- Opérations pour les calculs en coordonnées projectives
  - Multiplication modulaire par 2 : $2 \times A \pmod{2p}$
  - Multiplication modulaire par -3 : $-3 \times A \pmod{2p}$
- Opérations pour les calculs de formules unifiées d'additions
  - Cube modulaire : $A^3 \pmod{2p}$

Sommaire

Addition modulaire en BS

- Méthode de Takagi modulo $2p$

Entrées : $2^n \leq 2p < 2^{n+1}$, $-2p < A = (a_{n+1}a_n \cdots a_0)_{BS} < 2p$, $-2p < B = (b_{n+1}b_n \cdots b_0)_{BS} < 2p$.
Sortie : $S = A + B \pmod{2p}$, avec $S = (s_{n+1}s_n \cdots s_0)_{BS}$.

1. $(T^+, T^-) \leftarrow \mathrm{BSA}[(A^+, A^-), (B^+, B^-)]$
2. **si** $tv = 4t_{n+2} + 2t_{n+1} + t_n < 0$
3. $\quad (S^+, S^-) \leftarrow \mathrm{BSA}[(T^+, T^-), (2p, 0)]$
4. **sinon si** $tv > 0$
5. $\quad (S^+, S^-) \leftarrow \mathrm{BSA}[(T^+, T^-), (0, 2p)]$
6. **sinon si** $tv = 0$
7. $\quad (S^+, S^-) \leftarrow \mathrm{BSA}[(T^+, T^-), (0, 0)]$
8. **fin si**
9. **retourner** S

Multiplication de Montgomery en BS

Entrées : $p = (p_{n-1} \cdots p_1 p_0)_2$, $\text{pgcd}(p, 2) = 1$,
$-2p < A = (a_{n+1} a_n \cdots a_0)_{BS} < 2p$,
$-2p < B = (b_{n+1} b_n \cdots b_0)_{BS} < 2p$.
Sortie : $(S^+, S^-) = (A^+, A^-).(B^+, B^-).2^{-(n+2)} \pmod{2p}$.

1. $(U^+, U^-) \leftarrow \text{BSA}[(B^+, B^-), (p, 0)]$
2. $(V^+, V^-) \leftarrow \text{BSA}[(B^-, B^+), (p, 0)]$
3. $(S^+, S^-) \leftarrow \text{BSA}[(0, 0), (0, 0)]$
4. **pour** $i = 0$ à $n + 1$
5. $\quad m_i \leftarrow (a_i^+ \oplus a_i^-).(b_0^+ \oplus b_0^-)$
6. $\quad$ **si** $(a_i^+, a_i^-, m_i) = (0, 0, 0)$ ou $(1, 1, 0)$
7. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (0, 0)]$
8. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (1, 0, 0)$
9. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (B^+, B^-)]$
10. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (0, 1, 0)$
11. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (B^-, B^+)]$
12. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (0, 0, 1)$ ou $(1, 1, 1)$
13. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (p, 0)]$
14. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (1, 0, 1)$
15. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (U^+, U^-)]$
16. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (0, 1, 1)$
17. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (V^+, V^-)]$
18. $\quad$ **fin si**
19. $\quad (S^+, S^-) \leftarrow (S^+/2, S^-/2)$
20. **fin pour**
21. **retourner** (S^+, S^-)

Sommaire

Une observation et une optimisation possible

- Toutes les opérations modulaires pourront être effectuées à l'aide d'un seul BSA
  - $(C^+, C^-) \leftarrow \text{BSA}[(A^+, A^-), (B^+, B^-)]$
- Deux multiplexeurs en amont du BSA
  - MUX 1 pour sélectionner les valeurs de (A^+, A^-)
  - MUX 2 pour sélectionner les valeurs de (B^+, B^-)
- Or, 3 valeurs initialement possibles pour B^+ et B^- : 0, p et $2p$

$\Longrightarrow$ Se servir du caractère redondant de BS pour n'avoir que p comme entrée du MUX2

Addition modulaire en BS (version optimisée)

- Méthode de Takagi modulo $2p$

Entrées : $2^n \leq 2p < 2^{n+1}$, $-2p < A = (a_{n+1}a_n \cdots a_0)_{BS} < 2p$, $-2p < B = (b_{n+1}b_n \cdots b_0)_{BS} < 2p$.
Sortie : $S = A + B \pmod{2p}$, avec $S = (s_{n+1}s_n \cdots s_0)_{BS}$.

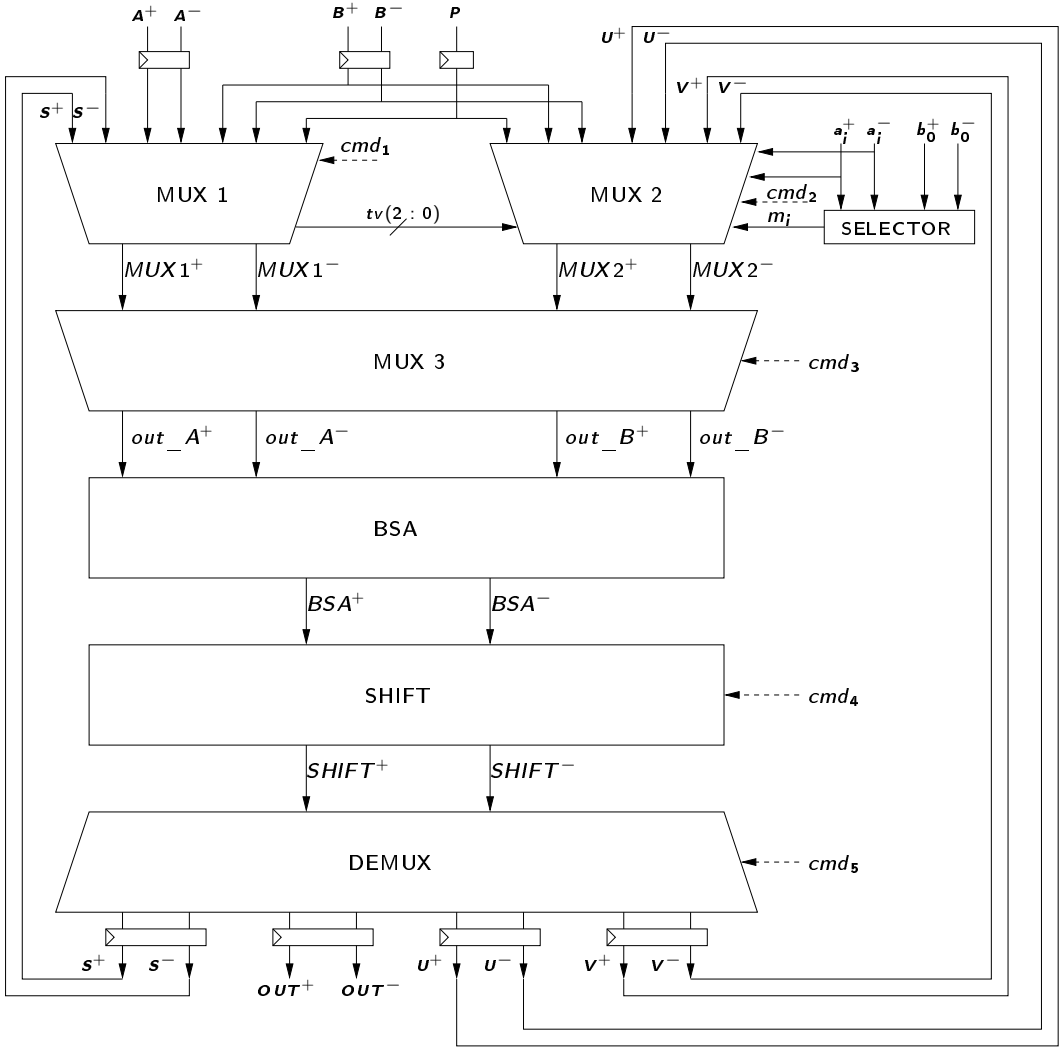
1. $(T^+, T^-) \leftarrow \mathrm{BSA}[(A^+, A^-), (B^+, B^-)]$
2. **si** $tv = 4t_{n+2} + 2t_{n+1} + t_n < 0$
3. $\quad (S^+, S^-) \leftarrow \mathrm{BSA}[(T^+, T^-), (4p, 2p)]$
4. **sinon si** $tv > 0$
5. $\quad (S^+, S^-) \leftarrow \mathrm{BSA}[(T^+, T^-), (2p, 4p)]$
6. **sinon si** $tv = 0$
7. $\quad (S^+, S^-) \leftarrow \mathrm{BSA}[(T^+, T^-), (p, p)]$
8. **fin si**
9. **retourner** S

Multiplication de Montgomery en BS (version optimisée)

Entrées : $p = (p_{n-1} \cdots p_1 p_0)_2$, $\text{pgcd}(p, 2) = 1$,
$-2p < A = (a_{n+1} a_n \cdots a_0)_{BS} < 2p$,
$-2p < B = (b_{n+1} b_n \cdots b_0)_{BS} < 2p$.
Sortie : $(S^+, S^-) = (A^+, A^-).(B^+, B^-).2^{-(n+2)} \pmod{2p}$.

1. $(U^+, U^-) \leftarrow \text{BSA}[(B^+, B^-), (2p, p)]$
2. $(V^+, V^-) \leftarrow \text{BSA}[(B^-, B^+), (2p, p)]$
3. $(S^+, S^-) \leftarrow \text{BSA}[(p, p), (p, p)]$
4. **pour** $i = 0$ à $n + 1$
5. $\quad m_i \leftarrow (a_i^+ \oplus a_i^-).(b_0^+ \oplus b_0^-)$
6. $\quad$ **si** $(a_i^+, a_i^-, m_i) = (0, 0, 0)$ ou $(1, 1, 0)$
7. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (p, p)]$
8. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (1, 0, 0)$
9. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (B^+, B^-)]$
10. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (0, 1, 0)$
11. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (B^-, B^+)]$
12. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (0, 0, 1)$ ou $(1, 1, 1)$
13. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (2p, p)]$
14. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (1, 0, 1)$
15. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (U^+, U^-)]$
16. $\quad$ **sinon si** $(a_i^+, a_i^-, m_i) = (0, 1, 1)$
17. $\quad\quad (S^+, S^-) \leftarrow \text{BSA}[(S^+, S^-), (V^+, V^-)]$
18. $\quad$ **fin si**
19. $\quad (S^+, S^-) \leftarrow (S^+/2, S^-/2)$
20. **fin pour**
21. **retourner** (S^+, S^-)

Architecture de l'unité arithmétique

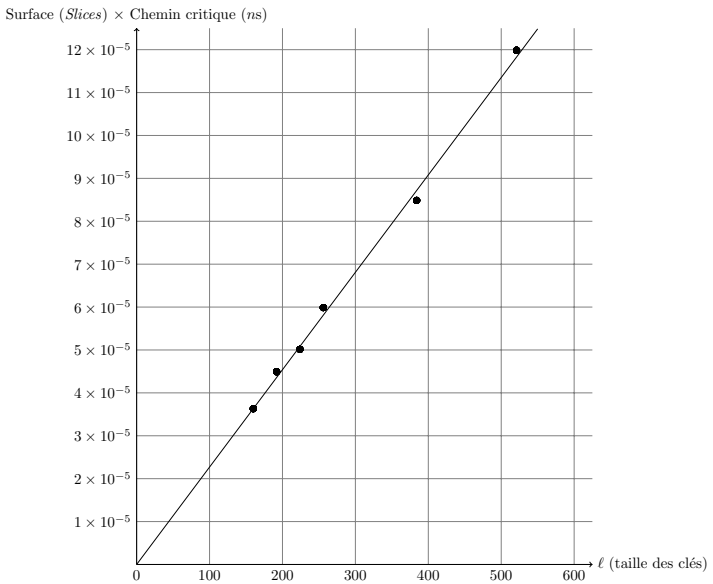


Sommaire

Propriétés de l'unité arithmétique

- FPGA Xilinx XCV2000
- Fréquence de fonctionnement quasi-constante quelque soit ℓ (longueur de k, en bits)

ℓ	160	192	224	256	384	521
Fréquence (MHz)	93	87	91	87	91	88



- Mise à l'échelle facilitée

Résultats comparatifs

Référence	Surface occupée	Temps [k]P	Opérateurs FPGA ?
[Orlando, 2001]	−	−	Non
[Byrne, 2007]	−	−	Non
[Mentens, 2007]	−	−	Non
[Örs, 2008]	−	−	Non
[Crowe, 2005]	+	−	Non
[Daly, 2005]	+	−	Non
[McIvor, 2004]	−	+	Oui
[Sakiyama, 2006]	−	+	Non
[Güneysu, 2008]	+	+	Oui

- Notre unité arithmétique supporte bien la comparaison avec l'état de l'art

Sommaire

Conclusion

- L'unité arithmétique proposée est très performante :
  - Fréquence de fonctionnement élevée
  - Mise à l'échelle facilitée
  - Temps de calcul de $[k]P$ souvent plus court que l'état de l'art
  - Architecture compacte
    - $\longrightarrow$ intégration dans des milieux fortement contraints (ex. : cartes à puce)
- Nouvelle implantation de l'algorithme de Montgomery en BS
- Apport décisif de la représentation BS

Perspectives

- Multiplieur de Montgomery en grande base dans lequel les opérandes sont représentés dans un système redondant (d'Avizienis)
  - Moins de cycles d'horloge
  - Mais...temps de cycle et surface augmentée
- Utiliser une méthode d'inversion modulaire en BS
  - PGCD
  - Inversion de Montgomery
- Utiliser des extensions arithmétiques du FPGA
  - blocs DSP
- Concevoir une unité arithmétique autonome
  - Implanter un ensemble de registres (blocs RAM du FPGA)
  - Conversion BS $\rightarrow$ Numération simple de position
- Implantation sur circuit dédié ASIC

Sommaire

Sommaire

Comparaisons

- Nombreuses contre-mesures pour les attaques simples par observation dans la littérature
- 2 unités arithmétiques de la littérature embarquent des protections :
  - [Byrne, 2007] : chaînes d'addition proposées par [Méloni, 2007]
  - [Ghosh, 2008] : opérations arithmétiques peuvent s'effectuer en parallèle
- Contre-mesure vis-à-vis des attaques simples par observation la plus rapide (à 1 processeur) de la littérature combine :
  - $\mathrm{NAF}_2(k)$
  - atomicité [Chavallier-Mames, 2004]

$\longrightarrow$ En implantant cette contre-mesure, notre unité arithmétique pour l'ECC est protégée contre les attaques simples par observation avec le plus petit temps de calcul de $[k]P$

Sommaire

Protéger un cryptosystème ECC grâce à l'implantation de la préservation de la parité

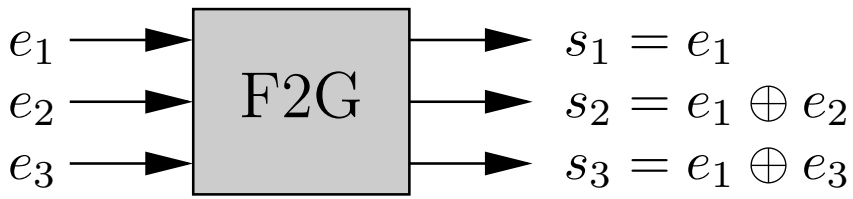
- [Parhami, 2006] propose des portes logiques pour laquelle la parité des entrées = parité des sorties (*Parity-Preserving Logic Gates*, PPLGs)
- Exemple : une PPLG ayant deux bits en entrée (e_1, e_2) et deux bits en sortie (s_1, s_2) doit obéir à la relation :

$$e_1 \oplus e_2 = s_1 \oplus s_2$$

- Proposées initialement pour augmenter la fiabilité des circuits
- Objectifs de cette étude :
  - Montrer que les PPLGs peuvent également être utilisées comme contre-mesure vis-à-vis des attaques par perturbation
  - Donner des résultats pratiques (surcoût, taux de détection de fautes)

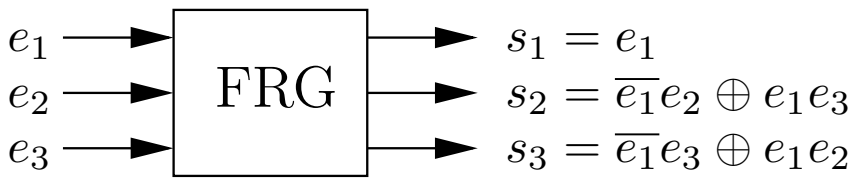
PPLGs

- Portes pour lesquelles la parité des entrées = parité sorties
- Porte de Feynman (F2G)



$\longrightarrow s_1 \oplus s_2 \oplus s_3 = e_1 \oplus (e_1 \oplus e_2) \oplus (e_1 \oplus e_3) = e_1 \oplus e_2 \oplus e_3$

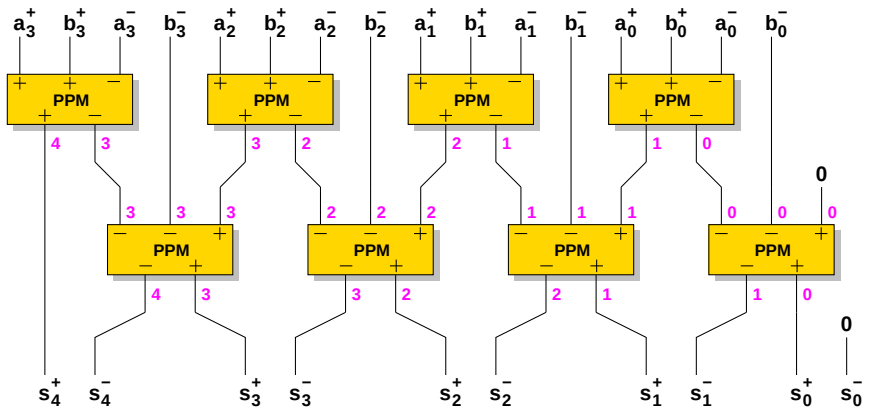
- Porte de Fredkin (FRG)



$\longrightarrow s_1 \oplus s_2 \oplus s_3 = e_1 \oplus (\overline{e_1}e_2 \oplus e_1e_3) \oplus (\overline{e_1}e_3 \oplus e_1e_2) =$
$e_1 \oplus e_2(e_1 \oplus \overline{e_1}) \oplus e_3(e_1 \oplus \overline{e_1}) = e_1 \oplus e_2 \oplus e_3$

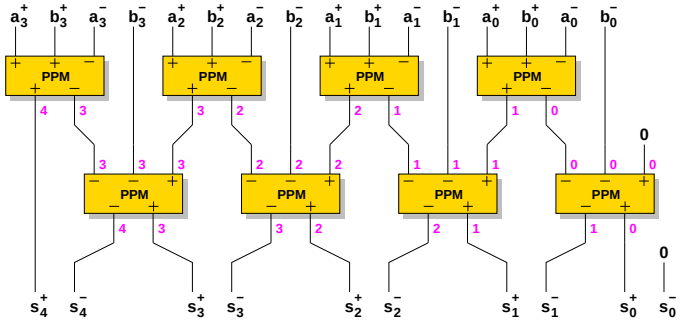
Conception d'un circuit préservant la parité (1/3)

- Exemple d'application : BSA



- 1. Choisir la partie du circuit à protéger
  - Diviser le circuit initial en n sous-circuits identiques (notés $\mathcal{C}$)
  - Chaque $\mathcal{C}$ sera ainsi protégé de la même façon avec des PPLGs
  - Exemple : $\mathcal{C}$ a comme bits d'entrées a_i^+, b_i^+, a_i^-, b_i^- et c_i^+ et comme bits de sortie s_{i+1}^- et s_i^+

Conception d'un circuit préservant la parité (2/3)



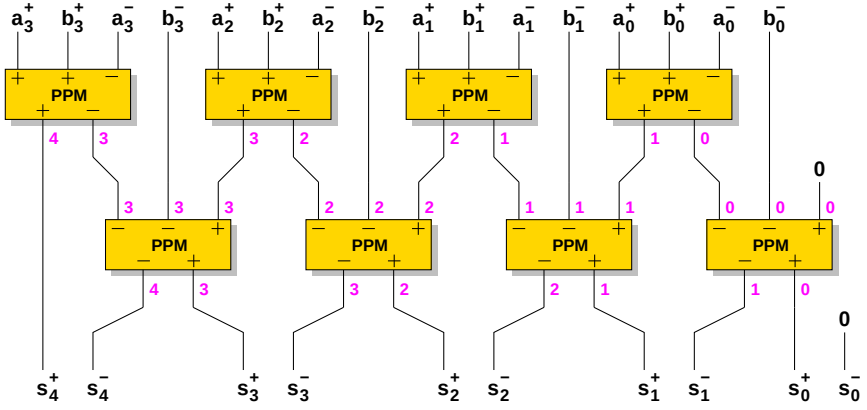
- 2. Obtenir les équations logiques correspondantes
- $\mathcal{C}$ relativement simple afin de récupérer les équations facilement
- Exemple : 1ère ligne de PPMs

$$\begin{cases} c_i^- = a_i^+ \oplus b_i^+ \oplus a_i^- \\ c_i^+ = a_{i-1}^+ . b_{i-1}^+ + a_{i-1}^+ . \overline{a_{i-1}^-} + b_{i-1}^+ . \overline{a_{i-1}^-} \end{cases}$$

2nde ligne de PPMs

$$\begin{cases} s_{i+1}^- = c_i^- . b_i^- + c_i^- . \overline{c_i^+} + b_i^- . \overline{c_i^+} \\ s_i^+ = c_i^- \oplus b_i^- \oplus c_i^+ \end{cases}$$

Conception d'un circuit préservant la parité (3/3)



- 3. Exprimer ces équations dans un corps de Galois
- Ni F2G, ni FRG ne peuvent calculer la fonction « OR »

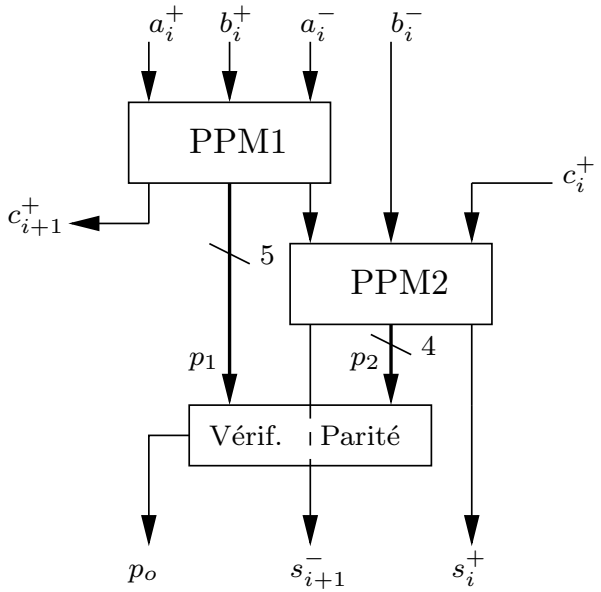
$\longrightarrow$ Transformer les équations logiques grâce à $x + y = x \oplus y \oplus x.y$

- Exemple :

$$\begin{cases} c_i^+ = a_{i-1}^+ . b_{i-1}^+ \oplus a_{i-1}^+ . \overline{a_{i-1}^-} \oplus b_{i-1}^+ . \overline{a_{i-1}^-} \\ s_{i+1}^- = c_i^- . b_i^- \oplus c_i^- . c_i^+ \oplus b_i^- . c_i^+ \end{cases}$$

- 4. Implanter ces équations grâce aux PPLGs

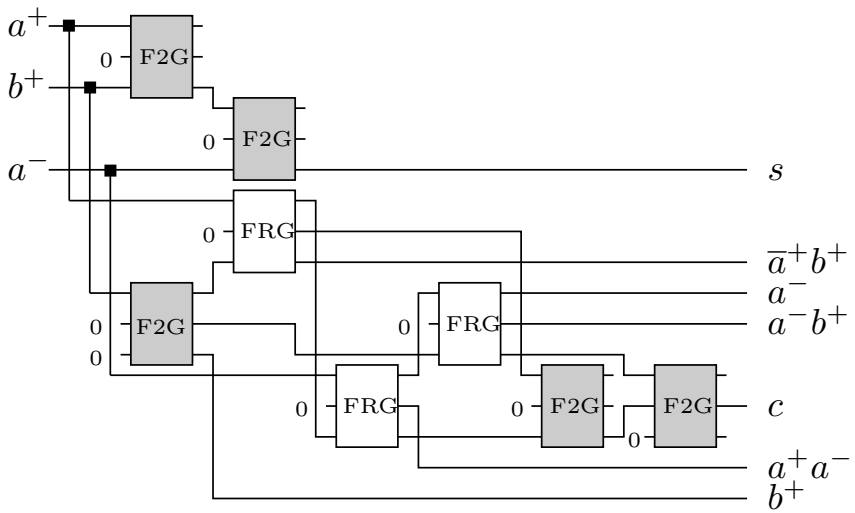
$\mathcal{C}$ protégé contre les fautes



$$\begin{cases} par_1 = a_i^+ \oplus b_i^+ \oplus a_i^- \oplus c_i^- \\ par_2 = a_i^+ \oplus b_i^+ \oplus a_i^- \oplus p_1 \oplus c_{i+1}^+ \\ par_3 = c_i^+ \oplus b_i^- \oplus c_i^- \oplus s_{i+1}^- \oplus p_2 \oplus s_i^+ \end{cases}$$

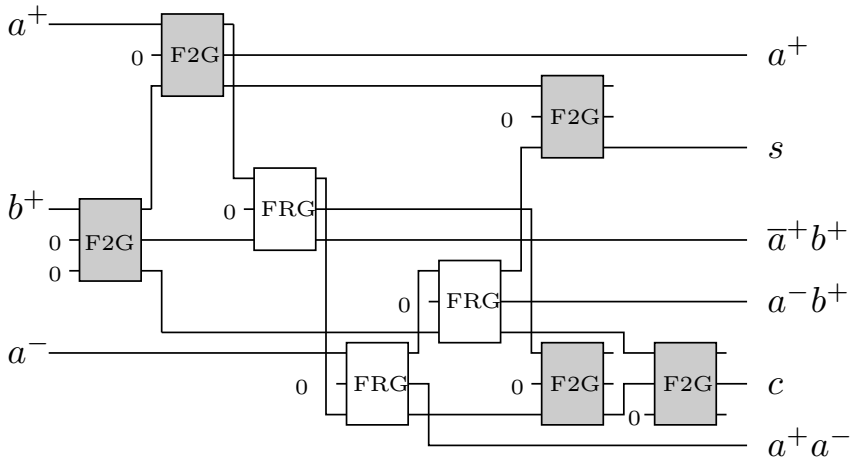
Si $p_o = par_1 + par_2 + par_3 = 1$, faute(s) détectée(s)

Cellule PPM1



- Duplication : F2G($e_1 = x$, $e_2 = 0$, $e_3 = 0$, $s_1 = x$, $s_2 = x$, $s_3 = x$)
- Bits de parité potentiellement simplifiables

Cellule PPM2



- Duplication : F2G($e_1 = x$, $e_2 = 0$, $e_3 = 0$, $s_1 = x$, $s_2 = x$, $s_3 = x$)
- Bits de parité potentiellement simplifiables

Résultats d'implantation

- Surcoût (avec $n = 160$)

Architecture	**Surface** (μm^2)	**Chemin critique** (ns)
BSA sans PPLGs	134440	1,39
BSA avec PPLGs	698157	5,69
Surcoût	$\times$5,2	$\times$4,1

- Capacité de détection de fautes

Nombre de bits fautés	**1 bit**	**2 bits**
Fautes détectées	80%	86,8%
Fautes sans conséquence	14,3%	2,1%
Fautes non-détectées	5,7%	11,1%

Conclusion et perspectives

- Utilisation du principe de la préservation de la parité pour protéger les cryptosystèmes contre les attaques par perturbation
  - Méthode de conception
  - Premiers résultats encourageants
- Perspectives :
  - Diminuer le taux de fautes non-détectées
  - Diminuer le surcoût sur la surface
    - Trouver de nouvelles PPLGs plus complexes
  - Étude plus poussée sur le mécanisme de détection
    - Autres modèles de fautes (collage, etc.), > 2 fautes, fautes contigues
  - Suite logicielle pour automatisation
  - Utiliser cette méthode pour protéger le reste de l'unité arithmétique (multiplexeurs, etc.)

Sommaire

Bilan des travaux effectués et perspectives (1/2)

- Conception d'une unité arithmétique hautes performances pour l'ECC
- Perspectives :
  - Multiplieur de Montgomery en grande base dans lequel les opérandes sont représentés dans un système redondant (d'Avizienis)
  - Inversion modulaire en BS
  - Extensions arithmétiques du FPGA
  - Unité arithmétique à l'autonomie accrue
  - Implantation sur circuit dédié ASIC
- Protection de cette unité arithmétique contre les attaques simples par observation amenant un temps de calcul de $[k]P$ plus petit que l'état de l'art

Bilan des travaux effectués et perspectives (2/2)

- Protection de cette unité arithmétique contre les attaques par perturbation à l'aide de la préservation de la parité
- Perspectives :
  - Diminuer le taux de fautes non-détectées
  - Diminuer le surcoût sur la surface
  - Étude plus poussée sur le mécanisme de détection
  - Suite logicielle pour automatisation
  - Utiliser cette méthode pour protéger le reste de l'unité arithmétique
- Plus largement, cette étude se veut être une aide utile pour tout concepteur de circuits pour l'ECC devant concilier des impératifs de performance et de sécurité

Production scientifique (selon norme AERES)

- Communications avec actes dans un congrès international
  - FDTC 2008, NordSec 2007, ReCoSoC 2006
- Communications avec actes dans un congrès national
  - JP-CNFM 2008
- Communications orales sans actes dans un congrès international ou national
  - YACC 2008, JNRDM 2008
- Communications par affiche dans un congrès international ou national
  - JNRDM 2007, ARCHI'07
- Distinction
  - Meilleure présentation orale aux JNRDM 2008
- Productions scientifiques futures
  - Article dans revue internationale avec comité de lecture pour l'unité arithmétique proposée au cours de cette étude

Cette présentation est terminée.
Je vous remercie de votre attention.
Je tâcherai de répondre à vos questions le plus clairement possible.
(Cette présentation a été réalisée avec la classe Beamer « Montpellier »)