Euclidean addition chains scalar multiplication on curves with efficient endomorphism

Fangan-Yssouf Dosso, Fabien Herbaut, Nicolas Méloni, Pascal Véron

Université de Toulon
Laboratoire IMath

Journées codage & cryptographie 2017
La Bresse, 25 avril 2017

Introduction

- Brique principale des cryptosystèmes basés sur les courbes elliptiques : le calcul de kP où k est un entier et P un point d'une courbe elliptique.
- **Objectif :** Proposer une méthode sûre et efficace qui, à partir des chaînes d'additions euclidiennes (EAC), effectue ce calcul.
- Exemple d'utilisation : ECDH (Elliptic Curve Diffie–Hellman)

Plan

Endomorphisme sur une courbe

Endomorphisme sur une courbe

Soit E une courbe elliptique définie sur un corps fini K.
Un endomorphisme ϕ sur $E(K)$ est une application telle que :

- $\phi : E \longrightarrow E$ avec $\phi(\infty) = \infty$
- $\phi(P_1 + P_2) = \phi(P_1) + \phi(P_2)$ pour tout $P_1, P_2 \in E$.

Remarques :

- Soit N un nombre premier tel que : N divise $\#(E)$ mais N^2 ne divise pas $\#(E)$. Alors, il existe un unique $\lambda \in [0, N-1]$ tel que $\phi(P) = \lambda.P$, $\forall P \in E$ d'ordre N.
- λ s'obtient avec les paramètres de E, indépendamment de P. C'est une racine (mod N) du polynôme caractéristique de ϕ.

Endomorphisme facilement calculable

- Facilement calculable signifie : s'obtient avec seulement quelques multiplications (et additions éventuellement) sur K.
- Exemple :
  Soit $E : y^2 = x^3 + b$ est une courbe définie sur $\mathbb{F}_p$ avec $p \equiv 1 (mod\ 3)$.
  Si $P = (x, y)$ alors $\phi(P) = \lambda P = (\beta x, y)$,
  où β est un élément de $\mathbb{F}_p$ d'ordre 3.
  Le polynôme caractéristique de ϕ est $X^2 + X + 1$.

La méthode GLV

- Galant, Lambert et Vanstone (GLV), en 2001.
- Principe :
  - Soient k un entier de taille n bits, E une courbe avec un endomorphisme facilement calculable et $P \in E$.
  - On décompose k en deux entiers positifs k_1 et k_2 tels que : $k = k_1 + k_2\lambda$ avec k_1 et k_2 de taille environ $(n/2)$ bits.
  - Ainsi, $kP = k_1.P + k_2(\lambda P) = k_1.P + k_2.Q$ (avec $Q = \phi(P) = \lambda P$)
  - On utilise ensuite un algorithme de multiplication simultanée de deux points par deux scalaires pour obtenir kP.
- Coût moyen : (n/2) doublements + (n/4) additions
- Remarque : pas SPA-secure.

GLV-SAC

- Faz-Hernandez, Longa et H. Sanchez, en 2015.
- Version SPA-secure de la méthode GLV.
- Identique au GLV sauf que k_1 et k_2 sont représentés de manière à effectuer à chaque tour de boucle un doublement et une addition.
- Coût : (n/2) doublements + (n/2) additions.

Chaîne d'additions euclidiennes (EAC)

EAC

Une chaîne d'additions euclidiennes C est une séquence binaire qui permet de calculer un (unique) entier k à partir d'un couple (a, b).

Soit $C = (c_1, c_2, ..., c_n) \in \{0, 1\}^n$ une EAC de longueur n.

Si C calcule k à partir de (a, b),

alors on notera : $k = \chi_{a,b}(C)$.

Exemple d'EAC

Exemple 1 : calcul de $\chi_{a,b}(C)$

Soit $C = (10110)$ une EAC de longueur 5.
Pour $(a, b) = (1, 2)$, on calcule l'entier correspondant comme suit :

$$(1,\ 2) \xrightarrow{1} (1,\ 3) \xrightarrow{0} (3,\ 4) \xrightarrow{1} (3,\ 7) \xrightarrow{1} (3,\ 10) \xrightarrow{0} (10,\ 13)$$

Ainsi, $k = \chi_{1,2}(C) = 10 + 13 = 23$.

On prend $(u_0, v_0) = (a, b)$ et $\forall i \geq 1$,

$(u_i, v_i) = (u_{i-1}, u_{i-1} + v_{i-1})$ si $c_i = 1$ (petit pas), ou

$(u_i, v_i) = (v_{i-1}, u_{i-1} + v_{i-1})$ si $c_i = 0$ (grand pas).

EAC et calcul de kP

Exemple 2 : calcul de $\chi_{a,b}(C).P$

On reprend l'EAC $C = (10110)$ et $(a, b) = (1, 2)$, on calcule $\chi_{1,2}(C).P$ comme suit :

$$(1.P,\ 2.P) \xrightarrow{1} (1.P,\ 3.P) \xrightarrow{0} (3.P,\ 4.P) \xrightarrow{1} (3.P,\ 7.P) \xrightarrow{1} (3.P,\ 10.P) \xrightarrow{0} (10.P,\ 13.P)$$

Ainsi, $\chi_{1,2}(C).P = 10.P + 13.P = 23.P$.

On prend $(u_0, v_0) = (a.P, b.P)$ et $\forall i \geq 1$,

$(u_i, v_i) = (u_{i-1}, u_{i-1} + v_{i-1})$ si $c_i = 1$ (petit pas), ou

$(u_i, v_i) = (v_{i-1}, u_{i-1} + v_{i-1})$ si $c_i = 0$ (grand pas).

La méthode ZADDU

- Proposée en 2007 par Méloni.
- Permet de calculer $P + Q$ dans le système de coordonnées Jacobien.
- $ZADDU(P, Q) \rightarrow (P', P + Q)$
  où P' est équivalent à P dans le système de coordonnées Jacobien et $Z_{P'} = Z_{P+Q}$.
- **Nécessite** : $Z_P = Z_Q$
- Coût : $5M + 2S$.
- $P + Q$ coûte de manière générale $8M + 4S$.
- $2.P$ coûte de manière générale $2M + 5S$.

avec M : la multiplication de deux entiers et S : l'élévation au carré.

L'algorithme : EAC-mult

EAC-mult

- Entée : aP, bP tels que $Z_{aP} = Z_{bP}$ et $C \in \{0,1\}^n$ une EAC.
- Sortie : $\chi_{a,b}(C).P$
- Début :
  - $(U_1, U_2) \leftarrow (aP, bP)$
  - pour $i = 1 \ldots n$ faire :
    - si $c_i = 1$ alors $(U_1, U_2) \leftarrow ZADDU(U_1, U_2)$
    - sinon $(U_1, U_2) \leftarrow ZADDU(U_2, U_1)$
  - $(U_1, U_2) \leftarrow ZADDU(U_1, U_2)$
  - retourner U_2
- Fin

Remarque : Cet algorithme est SPA-secure.

Calcul de C à partir de k

Pour construire une EAC pour un entier k, il suffit de prendre un entier g premier avec k et d'appliquer sur ces derniers la version soustractive de l'algorithme d'Euclide.

Exemple : Soient $k = 23$ et $g = 13$, on a :
$23–13 = 10$
$13–10 = 3$ (grand pas)
$10–3 = 7$ (petit pas)
$7–3 = 4$ (petit pas)
$4–3 = 1$ (grand pas)
$3–1 = 2$ (petit pas)
$2–1 = 1$
En lisant du bas vers le haut, on obtient $C = (10110)$ [et $(a, b) = (1, 2)$].

Calcul de C à partir de k : problèmes

Remarque

Il faut que l'EAC C qui calcule k soit (assez) courte pour que l'EAC-mult soit une alternative intéressante aux méthodes existantes.

Or

Problèmes

Pour un entier k donné, on ne sait pas dans un temps raisonnable :

- trouver une EAC courte qui calcule k.
- trouver une EAC de taille donnée qui calcule k.

Note : Il existe une implémentation matérielle FPGA efficace pour des chaînes de petites tailles.

Solutions

- Principe :
  - Choisir le couple (a, b).
  - Générer une EAC C (au lieu d'un entier k) de taille n appropriée.
  - Calculer $\chi_{a,b}(C)P$ avec l'EAC-mult.
- Donc, on évite les problèmes difficiles mentionnés plus haut.
- Mais, on a un nouveau problème : la distribution des entiers obtenus à partir de l'ensemble $\mathcal{M}_n$ des EAC de longueur n.

Solutions

- Principe :
  - Choisir le couple (a, b).
  - Générer une EAC C (au lieu d'un entier k) de taille n appropriée.
  - Calculer $\chi_{a,b}(C)P$ avec l'EAC-mult.
- Donc, on évite les problèmes difficiles mentionnés plus haut.
- Mais, on a un nouveau problème : la distribution des entiers obtenus à partir de l'ensemble $\mathcal{M}_n$ des EAC de longueur n.

Non injectivité de $\chi_{1,2}$

Soit $n > 0$ et $C = (c_1, ..., c_n) \in \mathcal{M}_n$, alors :
$\chi_{1,2}(c_1, ..., c_n) = \chi_{1,2}(c_n, ..., c_1)$,
i.e : C et son image miroir calculent le même entier.

Solution 1

- Herbaut, Liardet, Méloni, Teglia et Véron (en 2010).
- Soit $\mathcal{M}_n^0 = \{C \in \{0,1\}^{2n} \text{ avec } (c_1, ..., c_n) = 0^n\}$.

Proposition 1 : Injectivité de $\chi_{1,2}$ sur $\mathcal{M}_n^0$

La restriction de $\chi_{1,2}$ à $\mathcal{M}_n^0$ est injective,

et $\forall C \in \mathcal{M}_n^0,\ \ \chi_{1,2}(C) \leq F_{2n+4}$.

Note :

- F_n est le n-ième nombre de Fibonacci.
- $0^n = (0, ..., 0)$ [n fois].

Solution 1

- Soit $\psi_{a,b}$ l'application telle que $\psi_{a,b}(C) = (v_n, u_n)$, le dernier couple obtenu lors du calcul de $\chi_{a,b}(C)$ avec $C \in \mathcal{M}_n$.
- On a : $\psi_{1,2}(0^n) = (F_{n+2}, F_{n+3})$,

  donc, si P est **fixé**, on peut pré-calculer $F_{n+2}P$ et $F_{n+3}P$, pour ensuite appliquer l'EAC-mult avec $aP = F_{n+2}P$, $bP = F_{n+3}P$ et $C \in \mathcal{M}_n$.
- Remarque : si P n'est **pas fixé**, cette solution n'est pas efficace.

Solution 2

Objectif : Étendre l'EAC-mult au cas où P n'est pas fixé.

Proposition 2

Soient n, a et b des entiers tels que :
- $pgcd(a, b) = 1$
- $a > F_{n+2}$ ou $b > F_{n+2}$

alors la restriction de $\chi_{a,b}$ à $\mathcal{M}_n$ est injective.

i.e : les 2^n EAC de $\mathcal{M}_n$ calculent 2^n entiers differents avec $\chi_{a,b}$.

Note : pour $a = F_{n+2}$ et $b = F_{n+3}$, on retrouve la proposition 1.

Solution 2

Corollaire 1

Soient E une courbe elliptique et $P \in E$ d'ordre N.
Soient n, a et b des entiers tels que :

- $pgcd(a, b) = 1$
- $a > F_{n+2}$ ou $b > F_{n+2}$
- $aF_{n+1} + bF_{n+2} < N$ et $aF_{n+2} + bF_{n+1} < N$

alors l'EAC-mult permet de calculer 2^n points différents avec aP, bP et $\{C \in \mathcal{M}_n\}$.

Remarque : pour $a = F_{n+2}$ et $b = F_{n+3}$, on retrouve la solution 1.

Solution 2

- D'après le corollaire 1, si $a = 1$, il suffit que $b > F_{n+2}$ et $N > F_{n+1} + bF_{n+2}$.

  Ainsi, si P varie il faut qu'on puisse calculer facilement (rapidement) bP.

- D'où le choix de courbes avec des endomorphismes facilement calculables.

EAC-mult et endomorphismes

- Idée : prendre $\phi(P) = bP$ et donc $b = \lambda$.
- Problème : on n'a pas nécessairement $\lambda > F_{n+2}$.
- Mais, on montre que :

Proposition 3

Soient $X^2 + rX + s$ le polynôme caractéristique de ϕ et N l'ordre de P. Si :

- $\#(E)/N \leq 4$
- $N > F_{n+2}^2 (1 + |r| + s)$

alors l'EAC-mult permet de calculer 2^n points différents avec P, $\phi(P)$ et $\{C \in \mathcal{M}_n\}$.

EAC-mult et endomorphismes

Remarques :

- Si $K = GF(q)$, la proposition 3 (et le théorème de Hasse) imposent que $\lceil log_2(q) \rceil >= 1.4 \times n$.
- Donc, un corps un peu plus grand ($\times 1.4$) que pour d'autres méthodes (comme le GLV) pour un même niveau de sécurité.

Comparaison : EAC-mult, GLV et GLV-SAC

Coûts dans le système de coordonnées Jacobien

Soit $n/2$, le niveau de sécurité souhaité, alors :

- coût(GLV) $\approx n/2 \times (5.5M_n + 7S_n)$
- coût(GLV-SAC) $\approx n/2 \times (9M_n + 9S_n)$
- coût(EAC-mult) $\approx (n+1) \times (5M_t + 2S_t)$ [avec $t = 1.4 \times n$]

où M_n est la multiplication de deux entiers de taille n et S_n l'élévation au carré.

Comparaison : EAC-mult, GLV et GLV-SAC

- Donc, l'EAC-mult toujours plus lente que le GLV (qui n'est pas SPA-secure).
- Si le coût d'une multiplication est égal au coût d'une élévation au carré alors :
  - coût(GLV-SAC) $\approx n.(9M_n)$
  - coût(EAC-mult) $\approx n.(7M_t)$

  Dans ce cas, l'EAC-mult devient plus rapide que le GLV-SAC si $T(7M_t) < T(9M_n)$,
  i.e : $T(M_t)/T(M_n) < 1.29$ où T est le temps d'exécution.

Ratios multiplications modulaires

Exemples de ratios pour n = 256 et t = 358.

Ratios	multiplication	reduc. Modul.	mult-mod
C (GMP)	1.072	1.206	**1.147**
Android (BigInteger)	1.029	1.084	**1.067**
Java (BigInteger)	1.854	1.798	1.810

Table: Ratios de temps d'executions de la multiplication modulaire

Comparaison : EAC-mult, GLV et GLV-SAC

Exemples de ratios pour n = 256 et t = 358.

Ratios	Java	Android	C (GNU GMP)	Openssl
EAC-mult/GLV	1,75	**0.98**	1.32	1.09
EAC-mult/GLV-SAC	1,30	**0.73**	**0.96**	**0.97**

Application : EAC-mult et l'ECDH

Soit E une courbe elliptique avec un endomorphisme ϕ facilement calculable.
Soit $P \in E$ d'ordre $N > F_{n+2}^2 (1 + |r| + s)$ et $\#(E)/N \leq 4$.

Application : EAC-mult et l'ECDH

Soit E une courbe elliptique avec un endomorphisme ϕ facilement calculable.
Soit $P \in E$ d'ordre $N > F_{n+2}^2 (1 + |r| + s)$ et $\#(E)/N \leq 4$.

ECDH entre Alice et Bob

- Alice génère $C_A \in M_n$, calcule $A = \chi_{1,\lambda}(C_A)P$ et l'envoie à Bob.
- Bob génère $C_B \in M_n$, calcule $B = \chi_{1,\lambda}(C_B)P$ et l'envoie à Alice.
- Alice et Bob calculent respectivement $\chi_{1,\lambda}(C_A)B$ et $\chi_{1,\lambda}(C_B)A$.
- Le secret partagé est : $K = \chi_{1,\lambda}(C_A)B = \chi_{1,\lambda}(C_B)A$.

Conclusion

- L'EAC-mult est une méthode efficace et SPA-secure.
- L'EAC-mult peut être une alternative intéressante au GLV-SAC.

Conclusion

- L'EAC-mult est une méthode efficace et SPA-secure.
- L'EAC-mult peut être une alternative intéressante au GLV-SAC.

- Mais, pour l'instant pas de resultat sur la repartition des entiers calculés.
  Donc, déconseillée pour les protocoles de signatures (comme l'ECDSA).

Merci de votre attention.

Questions ?