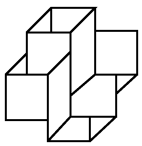
Implementação para Multiplicação por Escalar em Curvas Elípticas sobre $\mathbb{Z}_p$

Pedro Carlos da Silva Lara

Fábio Borges de Oliveira

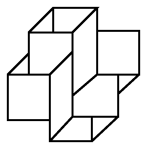
{pcslara, borges}@lncc.br

LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA – LNCC



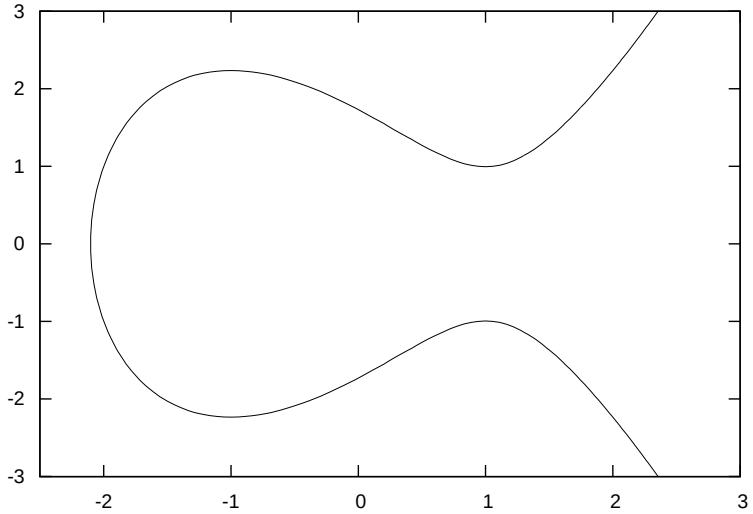
Introdução

Em 1975, Diffie e Hellman propuseram uma solução para que dois usuários estabelecessem uma chave secreta compartilhada em canal de comunicação inseguro. A segurança deste método estava baseada no Problema do Logaritmo Discreto (PLD) em grupos multiplicativos da forma $\mathbb{Z}_p^*$. Esta é considerada a primeira prática de criptografia assimétrica.



Introdução

Em 1985, Neal Koblitz e Victor Miller propuseram, independentemente, a aplicação de curvas elípticas em criptografia assimétrica. Ao invés de usar o grupo multiplicativo $\mathbb{Z}_p^*$ usa-se o grupo formado pelos pontos de uma curva elíptica $\Omega(\mathbb{Z}_p)$.

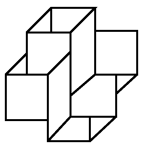


Vantagens

A tabela abaixo compara o tamanho das chaves criptográficas, mantendo em todas segurança equivalente.

Tabela 1: Tamanho das chaves em *bits*.

Modelo de Criptografia			
Simétrico	ECC	RSA	Razão ECC:RSA
80	163	1024	1:6
128	256	3072	1:12
192	384	7680	1:20
256	512	15360	1:30

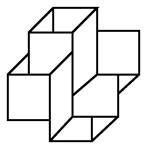


O que são curvas elípticas?

Definição. Uma curva elíptica sobre um corpo $\mathbb{F}$ (assumiremos sempre que $\mathbb{F}$ é um corpo de característica maior que 3) é o lugar geométrico dos pontos $(x, y) \in \mathbb{F} \times \mathbb{F}$ que satisfazem a equação

$$y^2 + axy + by = x^3 + cx^2 + dx + e \qquad (1)$$

mais um ponto, chamado de ponto no infinito, que será denotado por ∞.



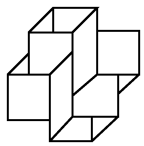
O que são curvas elípticas?

Podemos simplificar a equação (1) deixando na forma

$$y^2 = x^3 + ax + b \tag{2}$$

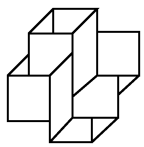
com $a, b \in \mathbb{F}$. Esta curva deve ser uma curva não-singular, ou seja, não possui raízes múltiplas, para tanto precisamos ter:

$$\Delta = 4a^3 + 27b^2 \neq 0$$



O PLD Sobre Curvas Elípticas

Considere dois pontos $P, Q \in \Omega(\mathbb{F}_q)$. O PLD sobre curvas elípticas consiste em encontrar um $k \in \mathbb{Z}$ tal que $Q = kP$. Quando estas variáveis são adequadas não existe um algoritmo satisfatório que consiga resolver este problema.

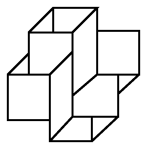


O PLD Sobre Curvas Elípticas (Exemplo)

Seja

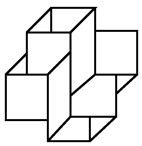
$$\Omega : y^2 = x^3 + 3x + 2$$

uma curva elíptica sobre $\mathbb{Z}_{11}$. Considere dois pontos desta curva, digamos $(4, 1), (6, 7) \in \Omega$. O PLD sobre curvas elípticas consiste em calcular k tal que $k \cdot (4, 1) = (6, 7)$, neste caso iremos apresentar todos os múltiplos do ponto $(4, 1)$.



O PLD Sobre Curvas Elípticas (Exemplo)

$i \cdot (4,1)$	$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$	$\lambda^2 - x_1 - x_2 = x_3$	$\lambda(x_1 - x_3) - y_1 = y_3$	(x_3, y_3)
$2 \cdot (4,1)$	9	7	5	$(7,5)$
$3 \cdot (4,1)$	$\frac{(5-1)}{(7-4)} = 9$	$(9^2 - 4 - 7) = 3$	$(9 \cdot (4-3) - 1) = 4$	$(3,4)$
$4 \cdot (4,1)$	$\frac{(4-1)}{(3-4)} = 5$	$(5^2 - 4 - 3) = 2$	$(5 \cdot (4-2) - 1) = 4$	$(2,4)$
$5 \cdot (4,1)$	$\frac{(4-1)}{(2-4)} = 8$	$(8^2 - 4 - 2) = 10$	$(8 \cdot (4-10) - 1) = 8$	$(10,8)$
$6 \cdot (4,1)$	$\frac{(8-1)}{(10-4)} = 4$	$(4^2 - 4 - 10) = 6$	$(4 \cdot (4-6) - 1) = 4$	$(6,4)$
$\mathbf{7 \cdot (4,1)}$	$\mathbf{\frac{(4-1)}{(6-4)} = 3}$	$\mathbf{(3^2 - 4 - 6) = 6}$	$\mathbf{(3 \cdot (4-6) - 1) = 7}$	$\mathbf{(6,7)}$
$8 \cdot (4,1)$	$\frac{(7-1)}{(6-4)} = 7$	$(7^2 - 4 - 6) = 10$	$(7 \cdot (4-10) - 1) = 3$	$(10,3)$
$9 \cdot (4,1)$	$\frac{(3-1)}{(10-4)} = 3$	$(3^2 - 4 - 10) = 2$	$(3 \cdot (4-2) - 1) = 7$	$(2,7)$
$10 \cdot (4,1)$	$\frac{(7-1)}{(2-4)} = 4$	$(4^2 - 4 - 2) = 3$	$(4 \cdot (4-3) - 1) = 7$	$(3,7)$
$11 \cdot (4,1)$	$\frac{(7-1)}{(3-4)} = 8$	$(8^2 - 4 - 3) = 7$	$(8 \cdot (4-7) - 1) = 6$	$(7,6)$
$12 \cdot (4,1)$	$\frac{(6-1)}{(7-4)} = 5$	$(5^2 - 4 - 7) = 4$	$(5 \cdot (4-4) - 1) = 10$	$(4,10)$



Algoritmos Utilizados (Multiplicação por escalar: Método Binário)

Entrada: Um inteiro $k = \sum_{i=0}^{j} 2^i k_i$ onde $k_i \in \{0, 1\}$ (representação em base binária), um ponto $P \in \Omega$ e a curva Ω

Saída: O ponto $k \cdot P \in \Omega$

início

$Q \leftarrow \infty$;

para $i = j$ **até** 0 **faça**

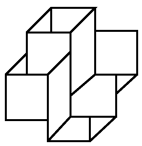
$Q \leftarrow 2 \cdot Q$;

se $k_i = 1$ **então**

$Q \leftarrow Q + P$;

retorna $Q \in \Omega$;

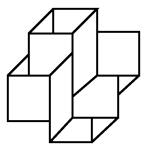
fim



Algoritmos Utilizados (Multiplicação por escalar: Método Binário)

A densidade média da representação binária de k é $\frac{1}{2}$, como $j = \log_2 k$ logo o tempo de execução esperado do algoritmo anterior é de aproximadamente $(j+1)/2$ adições de pontos (A) mais $j+1$ duplicações de pontos (D) denotado por:

$$\left(\frac{j+1}{2}\right) A + (j+1)D$$

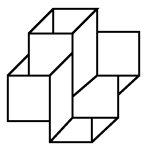


Algoritmos Utilizados (NAF)

Para algum K inteiro podemos representá-lo da seguinte maneira:

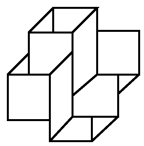
$$K = \sum_{i=0}^{j} k_i 2^i$$

onde $k_i \in \{0, \pm 1\}$. A forma não-adjacente NAF, é a representação de K, onde $k_i k_{i+1} = 0$ para todo $i \geq 0$.



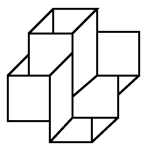
Algoritmos Utilizados (NAF)

A grande vantagem de se usar a representação NAF de um inteiro é que, de maneira geral, terá uma proporção maior de coeficientes iguais a zero. Assim fica reduzido o número de adições na multiplicação. O número esperado de coeficientes não nulos em uma representação NAF para um inteiro K com j bits é $j/3$. Na representação binária espera-se $j/2$ bits não nulos



Algoritmos Utilizados (NAF)

Fazemos $K = 52419574521$.
$(K)_2 = 1100001101000111001100111010111111001$
NAF$(K) = 10\bar{1}0001010\bar{1}0100100\bar{1}010\bar{1}0100\bar{1}0\bar{1}0000\bar{1}001$
Onde $\bar{1} = -1$. Observe que a representação NAF apresenta uma maior quantidade de zeros e nunca dois coeficientes diferentes de zero seguidos.



Algoritmos Utilizados (NAF)

Cálculo da representação NAF de um inteiro.

Entrada: Um inteiro K

Saída: NAF(K)

início

$i \leftarrow 0$;

enquanto $K \geq 1$ **faça**

se K *é ímpar* **então**

$k_i \leftarrow K \pmod{4}$;

$K \leftarrow K - k_i$;

senão

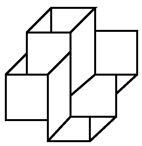
$k_i \leftarrow 0$;

$K \leftarrow K \gg 1$;

$i \leftarrow i + 1$;

retorna $(k_j, k_{j-1}, \ldots, k_0)$;

fim



Algoritmos Utilizados (Multiplicação por escalar: NAF)

Multiplicação por escalar usando NAF binário.

Entrada: Um inteiro K e um ponto $P \in \Omega$

Saída: $KP \in \Omega$

início

Calcule a representação NAF de K, $\sum_{i=0}^{j} k_i 2^i = \text{NAF}(K)$ (algoritmo anterior);

$Q \leftarrow \infty$;

para $i = j$ **até** 0 **faça**

$Q \leftarrow 2Q$;

se $k_i = 1$ **então**

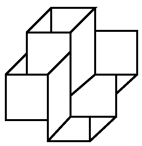
$Q \leftarrow Q + P$;

se $k_i = -1$ **então**

$Q \leftarrow Q - P$;

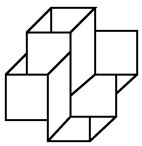
retorna Q;

fim



Algoritmos Utilizados (NAF com dimensão w)

O valor de $\text{NAF}_w(K)$ para $w \geq 2$ é o uma representação única da forma $K = \sum_{i=0}^{j} k_i 2^i$ onde cada $|k_i| < 2^{w-1}$. Fazendo uma analogia com o NAF mostrado anteriormente temos que $\text{NAF}_2(K) = \text{NAF}(K)$.



Algoritmos Utilizados (NAF com dimensão w)

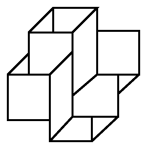
Exemplo: Considere $K = 52419574521$.

$\text{NAF}_2(K) = 10\bar{1}0001\bar{0}1\bar{0}1001\bar{0}0\bar{1}0\bar{1}0\bar{1}0\bar{0}0\bar{1}0\bar{0}0\bar{0}\bar{1}001$

$\text{NAF}_3(K) = 3000100\bar{3}00100\bar{0}0\bar{3}00\bar{3}00\bar{1}00\bar{3}0000\bar{1}001$

$\text{NAF}_4(K) = 300000700\bar{7}0000\bar{3}00\bar{3}00000\bar{5}00000007$

A densidade de coeficientes não nulos na representação $\text{NAF}_w(K)$ é $\frac{j}{w+1}$, onde $j = \log_2 K$.



Algoritmos Utilizados (NAF com dimensão w)

Cálculo de NAF$_w(K)$

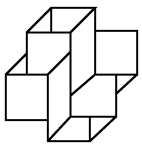
Entrada: Um inteiro K

Saída: NAF$_w(K)$

início

- $i \leftarrow 0$;
- **enquanto** $K \geq 1$ **faça**
  - **se** K *é ímpar* **então**
    - $k_i \leftarrow K \pmod{2^w}$;
    - $K \leftarrow K - k_i$;
  - **senão**
    - $k_i \leftarrow 0$;
  - $K \leftarrow K \gg 1$;
  - $i \leftarrow i + 1$;
- **retorna** $(k_j, k_{j-1}, \ldots, k_0)$;

fim



Algoritmos Utilizados (Multiplicação por escalar: NAF com dimensão w)

Entrada: Um inteiro K e um ponto $P \in \Omega$

Saída: $KP \in \Omega$

início

Calcule a representação NAF de K, $\sum_{i=0}^{j} k_i 2^i = \text{NAF}_w(K)$;

Precomputação: Calcule $P_i = iP$ para $i \in \{1, 3, 5, 7, \ldots, 2^{w-1} - 1\}$;

$Q \leftarrow \infty$;

para $i = j$ **até** 0 **faça**

$Q \leftarrow 2Q$;

se $k_i \neq 0$ **então**

se $k_i > 0$ **então**

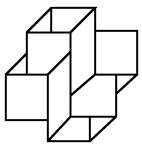
$Q \leftarrow Q + P_{k_i}$;

senão

$Q \leftarrow Q - P_{-k_i}$;

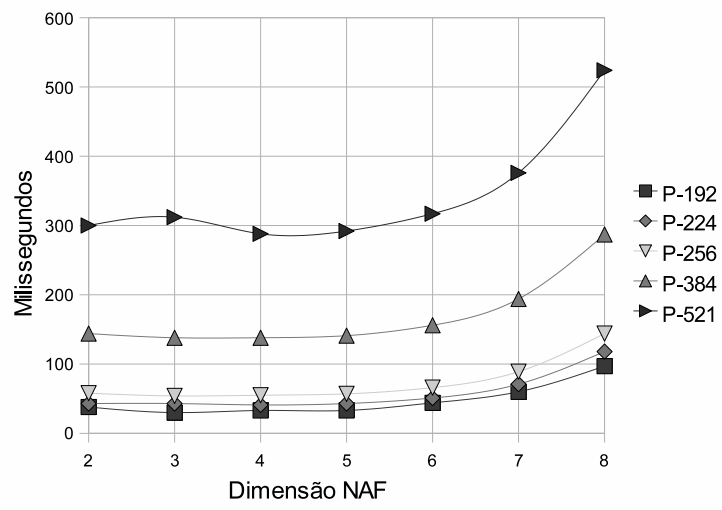
retorna Q;

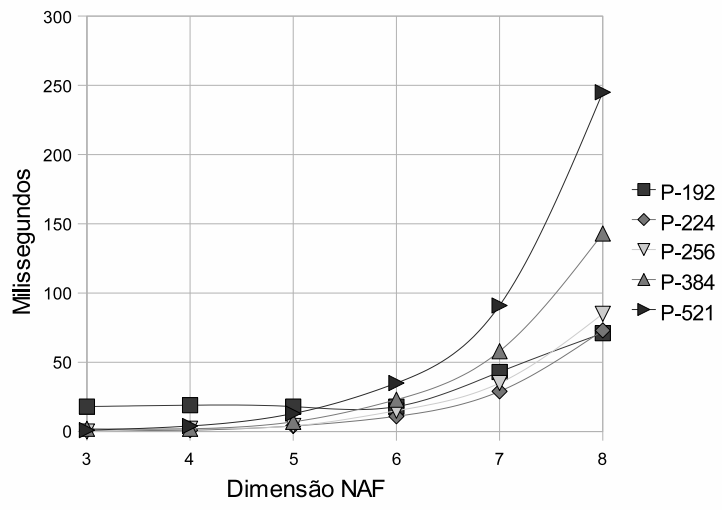
fim



Resultados e Comparações

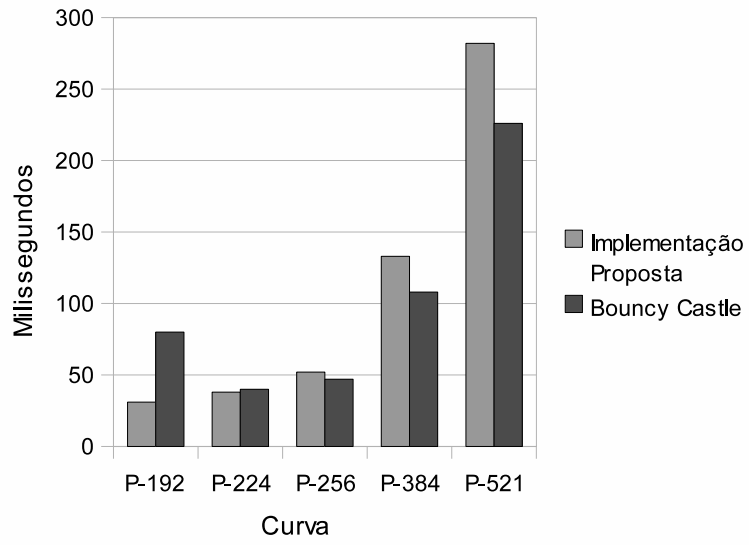
Tempo de execução para a implementação usando a representação NAF.

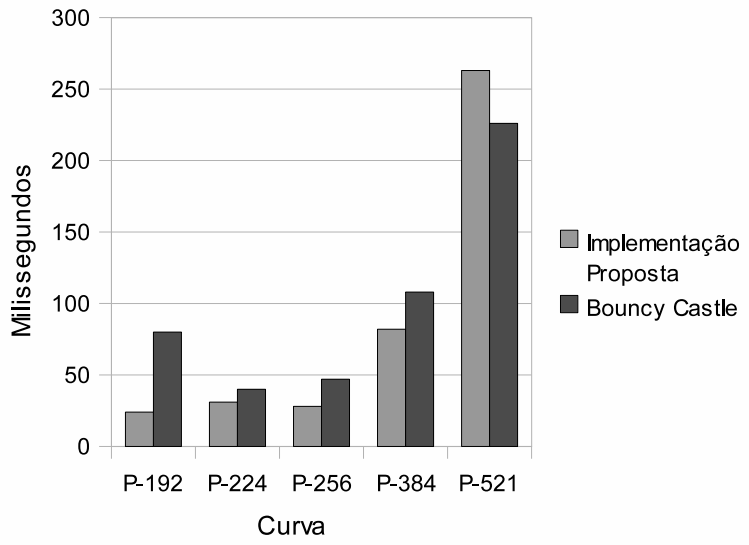




Resultados e Comparações

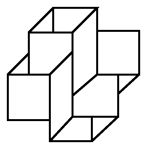
Comparamos a nossa implementação com a biblioteca Java código aberto *Bouncy Castle Crypto Package* versão 1.41.





Conclusões

- O uso da representação NAF reduz o tempo de execução da multiplicação por escalar.
- As dimensões da representação NAF que apresentaram melhores resultados são $3 \leq w \leq 5$.
- Sem contabilizar a pré-computação a dimensão da representação NAF que rendeu desempenhos melhores é $w = 9$.



Último Slide

- Obrigado!!!
- Quaisquer sugestões serão muito bem–vindas!