Effiziente Implementierung von elliptischen und hyperelliptischen Kurven

Peter Schwabe

Lehrstuhl für Theoretische Informationstechnik, RWTH Aachen

31.05.2006

Überblick

1. Kurze Wiederholung – elliptische und hyperelliptische Kurven
2. Einleitung und Motivation
3. Algorithmen
4. Zusammenfassung und Ausblick

Überblick

1. Kurze Wiederholung – elliptische und hyperelliptische Kurven
2. Einleitung und Motivation
3. Algorithmen
4. Zusammenfassung und Ausblick

Diplomarbeit

- Implementation von elliptischen und hyperelliptischen Kurven in C++

Überblick

1. Kurze Wiederholung – elliptische und hyperelliptische Kurven
2. Einleitung und Motivation
3. Algorithmen
4. Zusammenfassung und Ausblick

Diplomarbeit

- Implementation von elliptischen und hyperelliptischen Kurven in C++
- Verwendung der GnuMP-Bibliothek

Überblick

1. Kurze Wiederholung – elliptische und hyperelliptische Kurven
2. Einleitung und Motivation
3. Algorithmen
4. Zusammenfassung und Ausblick

Diplomarbeit

- Implementation von elliptischen und hyperelliptischen Kurven in C++
- Verwendung der GnuMP-Bibliothek
- Optimierung in Hinblick auf kryptographische Anwendungen

Gruppen elliptischer Kurven

- **char$(\mathbf{K}) \neq \mathbf{2}, \mathbf{3}$:** $y^2 = x^3 + ax + b$

Gruppen elliptischer Kurven

- **char(K) $\neq$ 2, 3:** $y^2 = x^3 + ax + b$
- **char(K) $=$ 2, nicht supersingulär:** $y^2 + xy = x^3 + ax^2 + b$

Gruppen elliptischer Kurven

- **char**$(\mathbf{K}) \neq \mathbf{2}, \mathbf{3}$**:** $y^2 = x^3 + ax + b$
- **char**$(\mathbf{K}) = \mathbf{2}$**, nicht supersingulär:** $y^2 + xy = x^3 + ax^2 + b$
- Punkte bilden eine abelsche Gruppe
  (mit Punkt im Unendlichen $\mathcal{O}$ als neutralem Element)

Gruppen hyperelliptischer Kurven

- $y^2 + h(x)y = f(x), \qquad \deg(h) \le g, \quad \deg(f) \le 2g + 1$

Gruppen hyperelliptischer Kurven

- $y^2 + h(x)y = f(x), \qquad \deg(h) \leq g, \quad \deg(f) \leq 2g + 1$
- Divisoren sind formale Summen von Punkten auf hyperelliptischen Kurven

Gruppen hyperelliptischer Kurven

- $y^2 + h(x)y = f(x), \qquad \deg(h) \leq g, \quad \deg(f) \leq 2g + 1$
- Divisoren sind formale Summen von Punkten auf hyperelliptischen Kurven
- Endliche Gruppe besteht aus Divisorklassen

Gruppen hyperelliptischer Kurven

- $y^2 + h(x)y = f(x), \qquad \deg(h) \leq g, \quad \deg(f) \leq 2g + 1$
- Divisoren sind formale Summen von Punkten auf hyperelliptischen Kurven
- Endliche Gruppe besteht aus Divisorklassen
- Elemente dieser Gruppe lassen sich darstellen als 2-Tupel von Polynomen von Grad $\leq g$ bzw. $< g$ (Mumford-Darstellung)

Gruppen hyperelliptischer Kurven

- $y^2 + h(x)y = f(x), \qquad \deg(h) \leq g, \quad \deg(f) \leq 2g + 1$
- Divisoren sind formale Summen von Punkten auf hyperelliptischen Kurven
- Endliche Gruppe besteht aus Divisorklassen
- Elemente dieser Gruppe lassen sich darstellen als 2-Tupel von Polynomen von Grad $\leq g$ bzw. $< g$ (Mumford-Darstellung)
- Cantors Algorithmus verwendet nur diese beiden Polynome zum Addieren zweier Gruppenelemente

DLP und Punktmultiplikation

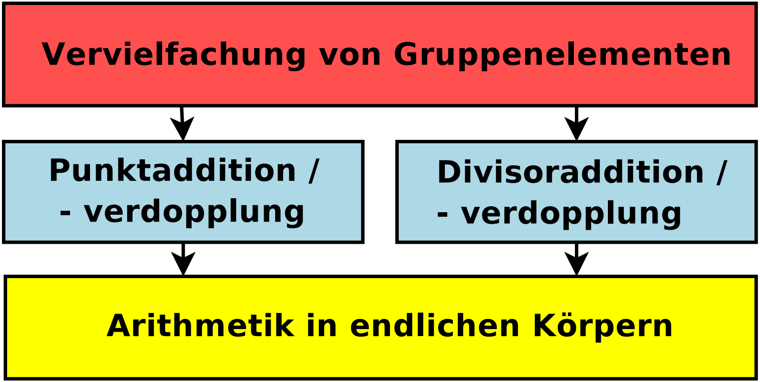
- DLP: Gegeben $k \cdot P$, berechne k

DLP und Punktmultiplikation

- DLP: Gegeben $k \cdot P$, berechne k
- Wie berechnet man effizient $k \cdot P$

DLP und Punktmultiplikation

- DLP: Gegeben $k \cdot P$, berechne k
- Wie berechnet man effizient $k \cdot P$



Körper primer Ordnung

- “Einfachste” Form endlicher Körper

Endliche Körper

Körper primer Ordnung

- “Einfachste” Form endlicher Körper
- Elemente sind ganze Zahlen (*integer*)

Endliche Körper

Körper primer Ordnung

- “Einfachste” Form endlicher Körper
- Elemente sind ganze Zahlen (*integer*)
- Multiplikation und Addition in $\mathbb{Z}$ mit anschließender modularer Reduktion

Endliche Körper

Körper primer Ordnung

- “Einfachste” Form endlicher Körper
- Elemente sind ganze Zahlen (*integer*)
- Multiplikation und Addition in $\mathbb{Z}$ mit anschließender modularer Reduktion

Körper nicht-primer Ordnung

- Elemente sind Polynome über $\mathbb{F}_p$

Endliche Körper

Körper primer Ordnung

- “Einfachste” Form endlicher Körper
- Elemente sind ganze Zahlen (*integer*)
- Multiplikation und Addition in $\mathbb{Z}$ mit anschließender modularer Reduktion

Körper nicht-primer Ordnung

- Elemente sind Polynome über $\mathbb{F}_p$
- Darstellung in *Polynombasis* (Koeffizientenvektor), Normalbasis oder als Potenz eines Erzeugers von $\mathbb{F}_q^*$

Endliche Körper

Körper primer Ordnung

- "Einfachste" Form endlicher Körper
- Elemente sind ganze Zahlen (*integer*)
- Multiplikation und Addition in $\mathbb{Z}$ mit anschließender modularer Reduktion

Körper nicht-primer Ordnung

- Elemente sind Polynome über $\mathbb{F}_p$
- Darstellung in *Polynombasis* (Koeffizientenvektor), Normalbasis oder als Potenz eines Erzeugers von $\mathbb{F}_q^*$
- Bei Charakteristik 2: Darstellung als Bitfolge

Multiplikation in endlichen Körpern

- Schulbuchmultiplikation zweier n-Bit Zahlen benötigt $O(n^2)$

Multiplikation in endlichen Körpern

- Schulbuchmultiplikation zweier n-Bit Zahlen benötigt $O(n^2)$
- Zahlreiche schnelle Methoden: *Karatsuba-Ofman*, Schönhage-Strassen (FFT)

Multiplikation in endlichen Körpern

- Schulbuchmultiplikation zweier n-Bit Zahlen benötigt $O(n^2)$
- Zahlreiche schnelle Methoden: *Karatsuba-Ofman*, Schönhage-Strassen (FFT)

Karatsuba-Ofman-Multiplikation

Multiplikation zweier $n = 2l$ Bit Zahlen $a = a_1 \cdot 2^l + a_0$ und $b = b_1 \cdot 2^l + b_0$

$$
\begin{aligned}
a \cdot b &= (a_1 \cdot 2^l + x_0) \cdot (b_1 \cdot 2^l + b_0) \\
&= {\color{red}a_1 b_1} 2^{2l} + [{\color{blue}(a_0 + a_1) \cdot (b_0 + b_1)} - {\color{red}a_1 b_1} - {\color{green}a_0 b_0}] 2^l + {\color{green}a_0 b_0}
\end{aligned}
$$

Multiplikation in endlichen Körpern

- Schulbuchmultiplikation zweier n-Bit Zahlen benötigt $O(n^2)$
- Zahlreiche schnelle Methoden: *Karatsuba-Ofman*, Schönhage-Strassen (FFT)

Karatsuba-Ofman-Multiplikation

Multiplikation zweier $n = 2l$ Bit Zahlen $a = a_1 \cdot 2^l + a_0$ und $b = b_1 \cdot 2^l + b_0$

$$
\begin{aligned}
a \cdot b &= (a_1 \cdot 2^l + x_0) \cdot (b_1 \cdot 2^l + b_0) \\
&= a_1 b_1 2^{2l} + [(a_0 + a_1) \cdot (b_0 + b_1) - a_1 b_1 - a_0 b_0] 2^l + a_0 b_0
\end{aligned}
$$

- Komplexität ist $O(n^{\log_2 3}) \sim O(n^{1,58})$

Quadrieren in endlichen Körpern

Charakteristik 2

- Quadrieren durch “Einsetzen von Nullen”

Quadrieren in endlichen Körpern

Charakteristik 2

- Quadrieren durch “Einsetzen von Nullen”
- Beispiel: $(x^4 + x^2 + x + 1)^2 = x^8 + x^4 + x^2 + 1$

Quadrieren in endlichen Körpern

Charakteristik 2

- Quadrieren durch “Einsetzen von Nullen”
- Beispiel: $(x^4 + x^2 + x + 1)^2 = x^8 + x^4 + x^2 + 1$
- $10111 \rightarrow 100010101$

Quadrieren in endlichen Körpern

Charakteristik 2

- Quadrieren durch “Einsetzen von Nullen”
- Beispiel: $(x^4 + x^2 + x + 1)^2 = x^8 + x^4 + x^2 + 1$
- $10111 \rightarrow 100010101$
- Anschließende modulare Reduktion

Quadrieren in endlichen Körpern

Charakteristik 2

- Quadrieren durch “Einsetzen von Nullen”
- Beispiel: $(x^4 + x^2 + x + 1)^2 = x^8 + x^4 + x^2 + 1$
- $10111 \rightarrow 100010101$
- Anschließende modulare Reduktion

Charakteristik > 2

- Es gibt Optimierungen, mit $S \sim 0.85M$

Invertierung in endlichen Körpern

- Invertierung mit Hilfe des erweiterten Euklidischen Algorithmus in $O(n^2)$

Invertierung in endlichen Körpern

- Invertierung mit Hilfe des erweiterten Euklidischen Algorithmus in $O(n^2)$
- Weitere Algorithmen sind bekannt

Invertierung in endlichen Körpern

- Invertierung mit Hilfe des erweiterten Euklidischen Algorithmus in $O(n^2)$
- Weitere Algorithmen sind bekannt
- Verhältnis des Aufwandes einer Inversion zu dem einer Multiplikation

Invertierung in endlichen Körpern

- Invertierung mit Hilfe des erweiterten Euklidischen Algorithmus in $O(n^2)$
- Weitere Algorithmen sind bekannt
- Verhältnis des Aufwandes einer Inversion zu dem einer Multiplikation
  - Für B-163: $I = 5M$ bzw. $I = 8M$ (Handbook of Elliptic Curve Cryptography)

Invertierung in endlichen Körpern

- Invertierung mit Hilfe des erweiterten Euklidischen Algorithmus in $O(n^2)$
- Weitere Algorithmen sind bekannt
- Verhältnis des Aufwandes einer Inversion zu dem einer Multiplikation
    - Für B-163: $I = 5M$ bzw. $I = 8M$ (Handbook of Elliptic Curve Cryptography)
    - Für P-192: $I = 80M$ (Handbook of Elliptic Curve Cryptography)

Addition und Verdopplung von Punkten

Addition

- $P = (x_P, y_P), Q = (x_Q, y_Q) \rightarrow P + Q = R = (x_R, y_R)$ mit
- $x_R = \left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q$
- $y_R = \left(\frac{y_Q - y_P}{x_Q - x_P}\right)(x_P - x_R) - y_P$

Addition und Verdopplung von Punkten

Addition

- $P = (x_P, y_P), Q = (x_Q, y_Q) \rightarrow P + Q = R = (x_R, y_R)$ mit
- $x_R = \left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q$
- $y_R = \left(\frac{y_Q - y_P}{x_Q - x_P}\right)(x_P - x_R) - y_P$

Verdopplung

- $P = (x_P, y_P), 2P = (x_R, y_R)$ mit
- $x_R = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P$
- $y_R = \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P$

Darstellung von Punkten in Projektiven Koordinaten

- Idee: Zusätzliche Information zur Vermeidung von Inversionen nutzen

Darstellung von Punkten in Projektiven Koordinaten

- Idee: Zusätzliche Information zur Vermeidung von Inversionen nutzen
- Betrachte Äquivalenzklassen
  $(X : Y : Z) = \{\gamma^c X, \gamma^d Y, \gamma Z \mid \gamma \in K^*\}$

Darstellung von Punkten in Projektiven Koordinaten

- Idee: Zusätzliche Information zur Vermeidung von Inversionen nutzen
- Betrachte Äquivalenzklassen $(X : Y : Z) = \{\gamma^c X, \gamma^d Y, \gamma Z \mid \gamma \in K^*\}$
- $(X : Y : Z)$ heißt projektiver Punkt und (X, Y, Z) ist ein Repräsentant von $(X : Y : Z)$

Darstellung von Punkten in Projektiven Koordinaten

- Idee: Zusätzliche Information zur Vermeidung von Inversionen nutzen
- Betrachte Äquivalenzklassen $(X : Y : Z) = \{\gamma^c X, \gamma^d Y, \gamma Z \mid \gamma \in K^*\}$
- $(X : Y : Z)$ heißt projektiver Punkt und (X, Y, Z) ist ein Repräsentant von $(X : Y : Z)$
- Bijektive Abbildung zwischen der Menge affiner und der projektiver Punkte mit $Z \neq 0$:
  - $(x, y) \mapsto (x, y, 1)$
  - $(X, Y, Z) \mapsto (X/Z^c, Y/Z^d)$

Darstellung von Punkten in Projektiven Koordinaten

- Idee: Zusätzliche Information zur Vermeidung von Inversionen nutzen
- Betrachte Äquivalenzklassen $(X : Y : Z) = \{\gamma^c X, \gamma^d Y, \gamma Z \mid \gamma \in K^*\}$
- $(X : Y : Z)$ heißt projektiver Punkt und (X, Y, Z) ist ein Repräsentant von $(X : Y : Z)$
- Bijektive Abbildung zwischen der Menge affiner und der projektiver Punkte mit $Z \neq 0$:
  - $(x, y) \mapsto (x, y, 1)$
  - $(X, Y, Z) \mapsto (X/Z^c, Y/Z^d)$
- $\mathcal{O}$ kann dargestellt werden durch $(X_\mathcal{O} : Y_\mathcal{O} : 0)$

Jacobi-Koordinaten

- Werden verwendet für Kurven über Körpern der Charakteristik > 3

Jacobi-Koordinaten

- Werden verwendet für Kurven über Körpern der Charakteristik > 3
- $c = 2$ und $d = 3$

Jacobi-Koordinaten

- Werden verwendet für Kurven über Körpern der Charakteristik > 3
- $c = 2$ und $d = 3$
- Projektive Kurvengleichung (Ersetze x durch X/Z^2 und y durch Y/Z^3)
$$Y^2 = X^3 + aXZ^4 + bZ^6$$

Jacobi-Koordinaten

- Werden verwendet für Kurven über Körpern der Charakteristik > 3
- $c = 2$ und $d = 3$
- Projektive Kurvengleichung (Ersetze x durch X/Z^2 und y durch Y/Z^3)
$$Y^2 = X^3 + aXZ^4 + bZ^6$$
- $\mathcal{O}$ wird dargestellt durch $(1 : 1 : 0)$
- Ist $P = (X : Y : Z)$, so ist $-P = (X : -Y : Z)$

Jacobi-Koordinaten

- Werden verwendet für Kurven über Körpern der Charakteristik > 3
- $c = 2$ und $d = 3$
- Projektive Kurvengleichung (Ersetze x durch X/Z^2 und y durch Y/Z^3)
$$Y^2 = X^3 + aXZ^4 + bZ^6$$
- $\mathcal{O}$ wird dargestellt durch $(1 : 1 : 0)$
- Ist $P = (X : Y : Z)$, so ist $-P = (X : -Y : Z)$
- Addition (projektiv - affin) benötigt 3 Quadrierungen und 8 Multiplikationen
- Verdopplung (projektiv) benötigt 6 Quadrierungen und 4 Multiplikationen

López-Dahab Koordinaten

- Werden verwendet für nicht supersinguläre Kurven über Körpern der Charakteristik 2

López-Dahab Koordinaten

- Werden verwendet für nicht supersinguläre Kurven über Körpern der Charakteristik 2
- $c = 1$ und $d = 2$

López-Dahab Koordinaten

- Werden verwendet für nicht supersinguläre Kurven über Körpern der Charakteristik 2
- $c = 1$ und $d = 2$
- Projektive Kurvengleichung:

$$Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4$$

López-Dahab Koordinaten

- Werden verwendet für nicht supersinguläre Kurven über Körpern der Charakteristik 2
- $c = 1$ und $d = 2$
- Projektive Kurvengleichung:

$$Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4$$

- $\mathcal{O}$ wird dargestellt durch $(1 : 0 : 0)$
- Ist $P = (X : Y : Z)$, so ist $-P = (X : X + Y : Z)$

López-Dahab Koordinaten

- Werden verwendet für nicht supersinguläre Kurven über Körpern der Charakteristik 2
- $c = 1$ und $d = 2$
- Projektive Kurvengleichung:

$$Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4$$

- $\mathcal{O}$ wird dargestellt durch $(1 : 0 : 0)$
- Ist $P = (X : Y : Z)$, so ist $-P = (X : X + Y : Z)$
- Addition (projektiv - affin) benötigt 8 Multiplikationen
- Verdopplung (projektiv) benötigt 4 Multiplikationen

Cantors Algorithmus

Eingabe Zwei Divisorklassen $D_1 = [a_1, b_1]$ und $D_2 = [a_2, b_2]$ auf der hyperelliptischen Kurve $C : y^2 + h(x)y = f(x)$

Ausgabe Den reduzierten Divisor D, so dass $D = D_1 + D_2$

$d_1 \leftarrow \mathrm{gcd}(a_1, a_2) \qquad [d_1 = e_1 a_1 + e_2 a_2]$

$d \leftarrow \mathrm{gcd}(d_1, b_1 + b_2 + h) \qquad [d = c_1 d_1 + c_2(b_1 + b_2 + h)]$

$s_1 \leftarrow c_1 e_1, \quad s_2 \leftarrow c_1 e_2, \quad s_3 \leftarrow c_2$

$a \leftarrow \frac{a_1 a_2}{d^2}, \quad b \leftarrow \frac{s_2 a_1 b_2 + s_2 a_2 b_1 + s_3(b_1 b_2 + f)}{d} \mod a$

repeat

$\quad a' \leftarrow \frac{f - bh - b^2}{a}, \quad b' \leftarrow (-h - b) \mod a'$

$\quad a \leftarrow a', \quad b \leftarrow b'$

until $\deg a \leq g$

Normiere a

return $[a, b]$

“Projektive” und “Neue” Koordinaten für Geschlecht 2 Kurven

- In affiner Darstellung werden 4 Elemente des zugrundeliegenden Körpers gespeichert: $[u_1, u_0, v_1, v_0]$ representiert $\mathrm{div}(x^2 + u_1 x + u_0, v_1 x + v_0)$

“Projektive” und “Neue” Koordinaten für Geschlecht 2 Kurven

- In affiner Darstellung werden 4 Elemente des zugrundeliegenden Körpers gespeichert: $[u_1, u_0, v_1, v_0]$ representiert $\mathrm{div}(x^2 + u_1 x + u_0, v_1 x + v_0)$
- Voraussetzung: Charakteristik des zugrunde liegenden Körpers ist ungerade und $\neq 5$

“Projektive” und “Neue” Koordinaten für Geschlecht 2 Kurven

- In affiner Darstellung werden 4 Elemente des zugrundeliegenden Körpers gespeichert: $[u_1, u_0, v_1, v_0]$ representiert $\mathrm{div}(x^2 + u_1 x + u_0, v_1 x + v_0)$
- Voraussetzung: Charakteristik des zugrunde liegenden Körpers ist ungerade und $\neq 5$
- “Projektive Koordinaten”: $[U_1, U_0, V_1, V_0, Z]$ steht für $\mathrm{div}(x^2 + U_1/Zx + U_0/Z, V_1/Zx + V_0/Z)$

“Projektive” und “Neue” Koordinaten für Geschlecht 2 Kurven

- In affiner Darstellung werden 4 Elemente des zugrundeliegenden Körpers gespeichert: $[u_1, u_0, v_1, v_0]$ representiert $\mathrm{div}(x^2 + u_1 x + u_0, v_1 x + v_0)$
- Voraussetzung: Charakteristik des zugrunde liegenden Körpers ist ungerade und $\neq 5$
- “Projektive Koordinaten”: $[U_1, U_0, V_1, V_0, Z]$ steht für $\mathrm{div}(x^2 + U_1/Zx + U_0/Z, V_1/Zx + V_0/Z)$
- “Neue Koordinaten”: $[U_1, U_0, V_1, V_0, Z_1, Z_2]$ steht für $\mathrm{div}(x^2 + U_1/Z_1^2 x + U_0/Z_1^2, V_1/(Z_1^3 Z_2)x + V_0/(Z_1^3 Z_2)))$

“Projektive” und “Neue” Koordinaten für Geschlecht 2 Kurven

- In affiner Darstellung werden 4 Elemente des zugrundeliegenden Körpers gespeichert: $[u_1, u_0, v_1, v_0]$ representiert $\mathrm{div}(x^2 + u_1 x + u_0, v_1 x + v_0)$
- Voraussetzung: Charakteristik des zugrunde liegenden Körpers ist ungerade und $\neq 5$
- “Projektive Koordinaten”: $[U_1, U_0, V_1, V_0, Z]$ steht für $\mathrm{div}(x^2 + U_1/Zx + U_0/Z, V_1/Zx + V_0/Z)$
- “Neue Koordinaten”: $[U_1, U_0, V_1, V_0, Z_1, Z_2]$ steht für $\mathrm{div}(x^2 + U_1/Z_1^2 x + U_0/Z_1^2, V_1/(Z_1^3 Z_2)x + V_0/(Z_1^3 Z_2)))$
- Für Charakteristik 2 gibt es ähnliche Darstellungen

Double and Add

Right-to-Left

Eingabe $k = (k_{t-1}, \ldots, k_0)$, P Erzeuger einer zyklischen additiven Gruppe

Ausgabe $k \cdot P$

$Q \leftarrow \mathcal{O}$
for $(i = 0, i \leq t-1, i++)$ **do**
  if $(k_i = 1)$ **then**
    $Q \leftarrow Q + P$
  end if
  $P \leftarrow 2P$
end for
return Q

Double and Add

Left-to-Right

Eingabe $k = (k_{t-1}, \ldots, k_0)$, P Erzeuger einer zyklischen additiven Gruppe

Ausgabe $k \cdot P$

$Q \leftarrow \mathcal{O}$
for $(i = t - 1, i \geq 0, i - -)$ **do**
    $Q \leftarrow 2Q$
    if $(k_i = 1)$ **then**
        $Q \leftarrow Q + P$
    end if
end for
return Q

Die Non-adjacent Form (NAF)

Definition

Die NAF einer natürlichen Zahl k ist eine Darstellung als

$$k = \sum_{i=0}^{l-1} k_i 2^i$$

mit $k_i \in \{0, \pm 1\},\ k_{l-1} \neq 0$ und der zusätzlichen Anforderung, dass keine zwei aufeinanderfolgenden k_i ungleich 0 sind.

Die Non-adjacent Form (NAF)

Definition

Die NAF einer natürlichen Zahl k ist eine Darstellung als

$$k = \sum_{i=0}^{l-1} k_i 2^i$$

mit $k_i \in \{0, \pm 1\},\ k_{l-1} \neq 0$ und der zusätzlichen Anforderung, dass keine zwei aufeinanderfolgenden k_i ungleich 0 sind.

Eigenschaften der NAF

- Die NAF einer Ganzzahl k ist eindeutig und wird mit NAF(k) bezeichnet

Die Non-adjacent Form (NAF)

Definition

Die NAF einer natürlichen Zahl k ist eine Darstellung als

$$k = \sum_{i=0}^{l-1} k_i 2^i$$

mit $k_i \in \{0, \pm 1\},\ k_{l-1} \neq 0$ und der zusätzlichen Anforderung, dass keine zwei aufeinanderfolgenden k_i ungleich 0 sind.

Eigenschaften der NAF

- Die NAF einer Ganzzahl k ist eindeutig und wird mit NAF(k) bezeichnet
- NAF(k) lässt sich in Linearzeit berechnen

Die Non-adjacent Form (NAF)

Definition

Die NAF einer natürlichen Zahl k ist eine Darstellung als

$$k = \sum_{i=0}^{l-1} k_i 2^i$$

mit $k_i \in \{0, \pm 1\},\ k_{l-1} \neq 0$ und der zusätzlichen Anforderung, dass keine zwei aufeinanderfolgenden k_i ungleich 0 sind.

Eigenschaften der NAF

- Die NAF einer Ganzzahl k ist eindeutig und wird mit NAF(k) bezeichnet
- NAF(k) lässt sich in Linearzeit berechnen
- NAF(k) hat die wenigsten Stellen $\neq 0$ aller Darstellungen, die vorzeichenbehaftete Bits verwenden

Die Non-adjacent Form (NAF)

Definition

Die NAF einer natürlichen Zahl k ist eine Darstellung als

$$k = \sum_{i=0}^{l-1} k_i 2^i$$

mit $k_i \in \{0, \pm 1\},\ k_{l-1} \neq 0$ und der zusätzlichen Anforderung, dass keine zwei aufeinanderfolgenden k_i ungleich 0 sind.

Eigenschaften der NAF

- Die NAF einer Ganzzahl k ist eindeutig und wird mit NAF(k) bezeichnet
- NAF(k) lässt sich in Linearzeit berechnen
- NAF(k) hat die wenigsten Stellen $\neq 0$ aller Darstellungen, die vorzeichenbehaftete Bits verwenden
- Der durchschnittliche Anteil an Stellen $\neq 0$ beträgt $\frac{1}{3}$

Punktmultiplikation mit der NAF

Eingabe $k \in \mathbb{N}$, P Erzeuger einer zyklischen additiven Gruppe
Ausgabe $k \cdot P$

```
Berechne NAF(k) = \sum_{i=0}^{l-1} k_i 2^i
Q ← O
for (i = l − 1, i ≥ 0, i − −) do
    Q ← 2Q
    if (k_i = 1) then
        Q ← Q + P
    end if
    if (k_i = −1) then
        Q ← Q − P
    end if
end for
return Q
```

Die Windowed-NAF (wNAF)

Definition

Sei $w \geq 2 \in \mathbb{N}$. Eine w-NAF einer positiven Ganzzahl k ist eine Darstellung der Form

$$\sum_{i=0}^{l-1} k_i 2^i$$

in der jedes k_i entweder 0 oder ungerade ist und in der für jedes k_i gilt, dass $|k_i| < 2^{w-1}$. Weiterhin ist $k_{l-1} \neq 0$ und höchstens eines von w aufeinanderfolgenden k_i ist ungleich 0.

Die Windowed-NAF (wNAF)

Definition

Sei $w \geq 2 \in \mathbb{N}$. Eine w-NAF einer positiven Ganzzahl k ist eine Darstellung der Form

$$\sum_{i=0}^{l-1} k_i 2^i$$

in der jedes k_i entweder 0 oder ungerade ist und in der für jedes k_i gilt, dass $|k_i| < 2^{w-1}$. Weiterhin ist $k_{l-1} \neq 0$ und höchstens eines von w aufeinanderfolgenden k_i ist ungleich 0.

Eigenschaften

- Jedes $k \in \mathbb{N}$ hat eine eindeutige w-NAF Darstellung $\mathrm{NAF}_w(k)$

Die Windowed-NAF (wNAF)

Definition

Sei $w \geq 2 \in \mathbb{N}$. Eine w-NAF einer positiven Ganzzahl k ist eine Darstellung der Form

$$\sum_{i=0}^{l-1} k_i 2^i$$

in der jedes k_i entweder 0 oder ungerade ist und in der für jedes k_i gilt, dass $|k_i| < 2^{w-1}$. Weiterhin ist $k_{l-1} \neq 0$ und höchstens eines von w aufeinanderfolgenden k_i ist ungleich 0.

Eigenschaften

- Jedes $k \in \mathbb{N}$ hat eine eindeutige w-NAF Darstellung $\mathrm{NAF}_w(k)$
- $\mathrm{NAF}_w(k)$ lässt sich in Linearzeit berechnen

Die Windowed-NAF (wNAF)

Definition

Sei $w \geq 2 \in \mathbb{N}$. Eine w-NAF einer positiven Ganzzahl k ist eine Darstellung der Form

$$\sum_{i=0}^{l-1} k_i 2^i$$

in der jedes k_i entweder 0 oder ungerade ist und in der für jedes k_i gilt, dass $|k_i| < 2^{w-1}$. Weiterhin ist $k_{l-1} \neq 0$ und höchstens eines von w aufeinanderfolgenden k_i ist ungleich 0.

Eigenschaften

- Jedes $k \in \mathbb{N}$ hat eine eindeutige w-NAF Darstellung $\mathrm{NAF}_w(k)$
- $\mathrm{NAF}_w(k)$ lässt sich in Linearzeit berechnen
- Es gilt $\mathrm{NAF}_2(k) = \mathrm{NAF}(k)$

Die Windowed-NAF (wNAF)

Definition

Sei $w \geq 2 \in \mathbb{N}$. Eine w-NAF einer positiven Ganzzahl k ist eine Darstellung der Form

$$\sum_{i=0}^{l-1} k_i 2^i$$

in der jedes k_i entweder 0 oder ungerade ist und in der für jedes k_i gilt, dass $|k_i| < 2^{w-1}$. Weiterhin ist $k_{l-1} \neq 0$ und höchstens eines von w aufeinanderfolgenden k_i ist ungleich 0.

Eigenschaften

- Jedes $k \in \mathbb{N}$ hat eine eindeutige w-NAF Darstellung $\mathrm{NAF}_w(k)$
- $\mathrm{NAF}_w(k)$ lässt sich in Linearzeit berechnen
- Es gilt $\mathrm{NAF}_2(k) = \mathrm{NAF}(k)$
- Der durchschnittliche Anteil an Stellen $\neq 0$ beträgt $1/(w+1)$

Punktmultiplikation mit der wNAF

Eingabe $k \in \mathbb{N}$, P Erzeuger einer zyklischen additiven Gruppe und $w \in \mathbb{N}$

Ausgabe $k \cdot P$

Berechne $\mathrm{NAF}_w(k) = \sum_{i=0}^{l-1} k_i 2^i$
Berechne $P_i = i \cdot P \; \forall \; i \in \{1, 3, 5, \ldots 2^{w-1} - 1\}$
$Q \leftarrow \mathcal{O}$
for $(i = l - 1, i \geq 0, i - -)$ **do**
  $Q \leftarrow 2Q$
  if $(k_i \neq 0)$ **then**
    if $(k_i > 0)$ **then**
      $Q \leftarrow Q + P_{k_i}$
    end if
    if $(k_i < 0)$ **then**
      $Q \leftarrow Q - P_{-k_i}$
    end if
  end if
end for
return Q

Zusammenfassung

Körperoperationen für P-192

- Double and Add
  - Affine Koordinaten: 23857
  - Jacobi-Koordinaten: 2500
- NAF-Methode
  - Affine Koordinaten: 21206
  - Jacobi-Koordinaten: 2162
- wNAF-Methode
  - Jacobi-Koordinaten: 2160 (w=4)

Zusammenfassung

Körperoperationen für B-163

- Double and Add
  - Affine Koordinaten: 2430
  - LD-Koordinaten: 1306
- NAF-Methode
  - Affine Koordinaten: 2160
  - LD-Koordinaten: 1090
- wNAF-Methode
  - Affine Koordinaten: 1980 (w=4)
  - LD-Koordinaten: 954 (w=4)

Ausblick

- Vergleiche meiner Implementation mit anderen Implementationen (Benchmarking)
- Betrachtung weiterer interessanter Spezialfälle (OEFs, Koblitz-Kurven, BN-Kurven)
- Hardwarenahe Implementation (Smartcards)
- Arithmetik in endlichen Körpern auf Grafikkarten?