Multipoint evaluation の高速実装

情報セキュリティ大学院大学
石黒 司
小崎 俊二
松尾　和人

2009/3/7

目次

背景

- multipoint evaluation：
    - 多項式へ複数の値を代入するアルゴリズム
    - 数論アルゴリズムやセキュリティ技術の基盤アルゴリズム
- multipoint evaluation の高速化
  $\longrightarrow$ 多項式の因子分解や超楕円曲線の位数計算の高速化

$\longrightarrow$ multipoint evaluation の実装評価を行った

Multipoint evaluation アルゴリズム

INPUT:

$$f \in \mathbb{F}_p[X], \deg f < n = 2^k$$

$$\mu_0, \cdots, \mu_{n-1} \in \mathbb{F}_p$$

OUTPUT:

$$f(\mu_0), \cdots, f(\mu_{n-1})$$

$n-1$ 次の多項式に元を一つ代入する計算量：$O(n)$ (Hornar 法)

$n-1$ 次の多項式に元を n 個代入する計算量：$O(n^2)$

$\Rightarrow O(M(n) \log n)$

$M(n)$: 次数 n の有限体上の多項式の乗算に必要な計算量

Multipoint evaluation アルゴリズム

- Moenck, Borodin,
  "Fast modular transform via division," 1972.
- Bostan, Lecerf and Schost,
  "Tellegenr's principle into practice," 2003.
- Montgomery,
  "An FFT Extension od the Elliptic Curve Method of Factorization," 1992.

Moenck アルゴリズム - Building Up Subproduct Tree

$$M_{3,0} = m_0 m_1 \cdots m_7$$

$$M_{2,0} = m_0 m_1 m_2 m_3 \qquad M_{2,1}$$

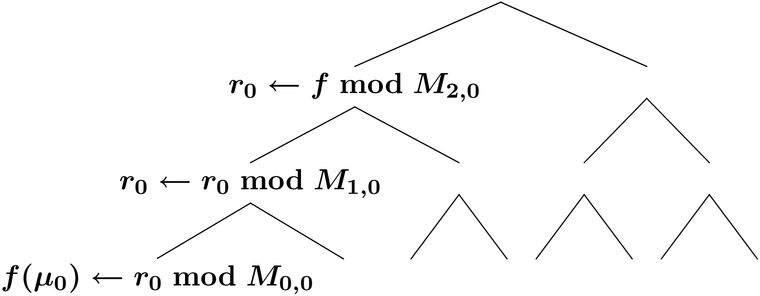
$$M_{1,0} = m_0 m_1 \qquad M_{1,1} = m_2 m_3 \qquad M_{1,2} \qquad M_{1,3}$$

$$m_0 = X - \mu_0 \quad m_1 \quad m_2 \quad m_3 \quad m_4 \quad m_5 \quad m_6 \quad m_7$$

$$O(M(n) \log n)$$

Moenck アルゴリズム - Going Down Subproducts Tree

$$f \in \mathbb{F}_p[X], \mu_0, \cdots, \mu_7 \in \mathbb{F}_p\ ,\ \deg f < 8 = 2^3$$



除算に必要な計算量 $O(M(n))$

Moenck **アルゴリズムの計算量は** $O(M(n)\log n)$

Montgomery アルゴリズム

Going Down Subproducts Tree **中の除算の計算量を削減する手法**

除算アルゴリズム

INPUT

$a, b \in \mathbb{F}_p[X]$, where $b \neq 0$ is monic

OUTPUT

$r \in \mathbb{F}_p[X]$ where $a = qb + r, \deg r < \deg b$

$m \leftarrow \deg a - \deg b$

$c \leftarrow Newt(\mathrm{rev}(b), m+1)$ s.t. $\mathrm{rev}(b)c \equiv 1 \pmod{X^{m+1}}$

$q \leftarrow \mathrm{rev}(a)c \bmod X^{m+1}$

$q \leftarrow \mathrm{rev}_m(q)$

$r \leftarrow a - bq$

$$a = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_0 \in \mathbb{F}_p[X]$$
$$\mathrm{rev}(a) = a_0 X^n + a_1 X^{n-1} + \cdots + a_n$$

Modified polynomial remainder

INPUT

$a, b, c \in \mathbb{F}_p[X]$, where $b \neq 0$ is monic,

$c \equiv \mathrm{rev}(b)^{-1} \pmod{X^{\deg a - \deg b + 1}}$

OUTPUT

$r \in \mathbb{F}_p[X]$ where $a = qb + r, \deg r < \deg b$

$m \leftarrow \deg a - \deg b$

$//c \leftarrow Newt(\mathrm{rev}(b), m+1)$ s.t.$\mathrm{rev}(b)c \equiv 1 \pmod{X^{m+1}}$

$q \leftarrow \mathrm{rev}(a)c \bmod X^{m+1}$

$q \leftarrow \mathrm{rev}_m(q)$

$r \leftarrow a - bq$

Going Down Subproducts Tree

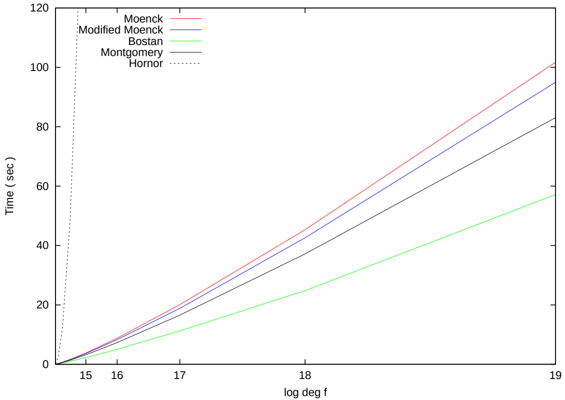
$$(M_{3,0})^{-1} \bmod X^8$$

$$(M_{2,0})^{-1} \bmod X^4 \qquad (M_{2,1})^{-1} = {M_{3,0}}^{-1} M_{2,0} \bmod X^4$$

$$(M_{1,0})^{-1} \bmod X^2$$

Multipoint Evaluation 結果

$\mathbb{F}_p$: 120bit



実験環境： CPU:Opteron(tm) Processor 2.7GHz, memory:16GB
NTL5.4.2, GCC4.3.2, on SUSE Linux

まとめ

- Montogomery アルゴリズムの実装を行い評価した。
- Moenck アルゴリズムよりも高速であるが、
  Bostan アルゴリズムよりも低速であった。