ホーム 情報セキュリティ 暗号技術

楕円曲線演算の高速化

楕円曲線演算の計算量

▶ 標数 $p > 3$ の体

楕円加算と2倍算の計算量を下表に示す． 表において， I は逆元を M は乗算を示す．例えば，アフィン座標系での加算は，逆元1回と乗算3回で実行できることを示す．

楕円演算の演算量(標数 $p > 3$ の体)

演算	アフィン座標	射影座標
加算	$1I + 3M$	$16M$
2倍算 ($a \neq -3$)	$1I + 4M$	$10M$
2倍算 ($a = -3$)	$1I + 4M$	$8M$

一般的に，逆元時間 > 5〜10 × 乗算時間ならば射影系の座標が高速になる．

▶ 標数 2 の体

楕円加算と2倍算の計算量を下表に示す． 表において， I は逆元， M は乗算， S は平方算を示す． 標数2の体の場合，平方演算のコストは通常の乗算に比べてはるかに低いため，乗算と平方算を区別している． 表から分かるように，標数2の体の場合，射影座標において点を2倍するコストは一般の点の加法に比べて3倍近く高速になる．

楕円演算の演算量(標数 2 の体)

演算	アフィン座標	射影座標
加算 ($a \neq 0$)	$1I + 2M + 1S$	$15M + 5S$
加算 ($a = 0$)	$1I + 2M + 1S$	$14M + 4S$
2倍算	$1I + 2M + 1S$	$5M + 5S$

楕円点の k 倍算

楕円曲線における点の k 倍算 (スカラー倍算) は，アーベル群における一般のべき乗計算問題の特別な場合である．そのため，一般の整数における演算の高速化手法が適用可能である．

二進展開法 (Binary method)

入力: 点 P， n ビット整数 $k = \Sigma_{\{j=0\sim n-1\}} k_j 2^j$, $k_j \in \{0, 1\}$.

出力: 点 $Q = [k]\ P$.

1. $Q \leftarrow O$
2. For $j = n - 1$ to 0 by -1 do
3. $\quad Q \leftarrow [2]\ Q$
4. $\quad$ If $k_j = 1$ then $Q \leftarrow Q + P$
5. Return Q

二進展開法では， n を k の二進展開の長さ， W を k における1の個数とするとき， $n - 1$ 回の2倍算と $W - 1$ 回の加算を必要とする．平均的に $W = n/2$ とすれば，これらの回数はそれぞれ， $n - 1$， $n/2 - 1$ となる．

m進展開法

m 進展開法は， k の m 進展開を利用する． ある $r \geq 1$ に対し， $m = 2^r$ とする． $r = 1$ の場合は二進展開法になる．

入力： 点 P, 整数 $k = \Sigma_{\{j=0\sim d-1\}} k_j\, m^j$, $k_j \in \{0, 1, \ldots, m-1\}$.

出力： 点 $Q = [k]P$.

事前計算：

1. $P_1 \leftarrow P$
2. For $i = 2$ to $m - 1$ do $P_i \leftarrow P_{i-1} + P$ ($P_i = [i]P$ を計算)
3. $Q \leftarrow O$

主計算：

1. For $j = d - 1$ to 0 by -1 do
2. $Q \leftarrow [m]Q$ (2倍算を r ($m = 2^r$) 回実行)
3. $Q \leftarrow Q + P_{k_j}$
4. Return Q

移動窓法(Sliding window method)

移動窓法では，乗数 k のビットは，長さ r のブロック(窓)の中で処理される．$r > 1$ を仮定する．

入力： 点 P, 整数 $k = \Sigma_{\{j=0\sim n-1\}} k_j 2^j$, $k_j \in \{0, 1\}$.

出力： 点 $Q = [k]P$.

事前計算：

1. $P_1 \leftarrow P$, $P_2 \leftarrow [2]P$
2. For $i = 1$ to $2^{r-1} - 1$ do $P_{2i+1} \leftarrow P_{2i-1} + P_2$
3. $j \leftarrow n - 1$, $Q \leftarrow O$

主計算：

1. While $j \geqq 0$ do
2. If $k_j = 0$ then $Q \leftarrow [2]Q$, $j \leftarrow j - 1$
3. Else do
4. t を $j - t + 1 \leqq r$ と $k_t = 1$ である最小の整数とする
5. $h_j \leftarrow (k_j\, k_{j-1} \cdots k_t)_2$
6. $Q \leftarrow [2^{j-t+1}]Q + P_{h_j}$
7. $j \leftarrow t - 1$
8. Return Q

符号付きm進展開窓法

- 符号付き数表現
  符号付き桁(SD: Signed Digit)表示とは，

  $$k = \Sigma_{\{i=0\sim m s_i\}} 2^i, \quad s_i \in \{-1, 0, 1\}$$

  の形式の数表現である．この表現は，二進数表示を含み，またすべての整数 k ($0 \leqq k \leqq 2^{m+1} - 1$) も含まれている．しかし，この表現は 3^{m+1} 個の組合せがあるので冗長である．
  SD表現が空疎(sparse)であるとは，隣接した0でないビットが存在しないこと，すなわち任意の $i \geqq 0$ について $s_i\, s_{i+1} = 0$ となることである．この表現は非隣接形式(NAF:non-adjacent form)と呼ばれる． 任意の整数 k はNAFを一意に持ち，以下のアルゴリズムで求められる．

- NAFへの変換

  入力： 整数 $k = \Sigma_{\{i=0\sim m-1\}} k_i\, 2^i$, $k_i \in \{0, 1\}$

  出力： NAF $k = \Sigma_{\{i=0\sim m\}}\, s_i\, 2^i$, $k_i \in \{-1, 0, 1\}$

  1. $c_0 \leftarrow 0$
  2. For $i = 0$ to m do
  3. $c_{i+1} \leftarrow$ **floor** $[(k_i + k_{i+1} + c_i)/2]$ ($i \geqq m$ に対して$k_i = 0$)
  4. $s_i \leftarrow k_i + c_i - 2c_{i+1}$
  5. Return $(s_m\, s_{m-1} \cdots s_0)$

- 符号付きm進展開への変換
  入力: 整数 $k = \Sigma_{\{j=0\sim l-1\}} k_j\, 2^j$, $k_j \in \{0, 1\}$, $k_l = 0$.
  出力: $\{(b_i, e_i)\}_{\{i=0\sim d-1\}}$ の数列.

  1. $d \leftarrow 0$, $j \leftarrow 0$

2. While $j \leqq l$ do
3. If $k_j = 0$ then $j \leftarrow j + 1$
4. Else do
5. $t \leftarrow$ **min** $\{l, j + r - 1\}$, $h_d \leftarrow (k_t\, k_{t-1} \cdots k_j)_2$
6. If $h_d > 2^r$ then do
7. $b_d \leftarrow h_d - 2^r$
8. 数 $(k_l\, k_{l-1} \cdots k_{t+1})_2$ を1増やす
9. Else $b_d \leftarrow h_d$
10. $c_d \leftarrow j$, $d \leftarrow d + 1$, $j \leftarrow t + 1$
11. Retuen 数列 $(b_0, e_0), (b_1, e_1), \ldots, (b_{d-1}, e_{d-1})$

▶ 符号付きm進展開窓法

入力： 点 P, 整数 $k = \Sigma_{\{i=0\sim d-1\}}\, b_i\, 2^{e_i}$ を満たす $\{(b_i, e_i)\}_{\{i=0\sim d-1\}}$.

出力： 点 $Q = [k]P$.

事前計算：

1. $P_1 \leftarrow P$, $P_2 \leftarrow [2]P$
2. For $i = 1$ to $2^{r-2} - 1$ do $P_{2i+1} \leftarrow P_{2i-1} + P_2$
3. $Q \leftarrow P_{b_{d-1}}$

主計算：

1. For $i = d - 2$ to 0 by -1 do
2. $Q \leftarrow [2^{e_{i+1}} - e_i]\, Q$
3. If $b_i > 0$ then $Q \leftarrow Q + P_{b_i}$
4. Else $Q \leftarrow Q - P_{-b_i}$
5. $Q \leftarrow [2^{e_0}]Q$
6. Return Q

演算法の比較

楕円点の k 倍算 $P = [k]Q$ を計算する場合，k の値は次のように分割されて処理される．

k = 1001010110110100010100100010

- 2進展開法
  1001010110110100010100100010
- m 進展開法 ($m = 4$)
  1001 0101 1011 0100 0101 0010 0010
- 移動窓法 ($r = 4$)
  1001 0 1011 0 1101 000 101 00 1 000 1 0

2進展開法では全てのビットが順番に処理されるが，m 進展開法では4ビットの窓7個に対して処理が行われる．また，移動窓法では6個の窓に対して処理が行われる．

Prev Page Top Next