一类超椭圆曲线上的快速除子标量乘

游 林
(杭州电子科技大学通信工程学院, 浙江杭州 310018)

摘 要: 除子标量乘是超椭圆曲线密码体制中的关键运算. 基于单除子标量乘的思想, 将 Duursma 与 Sakurai 给出的关于奇素数域上一类特殊超椭圆曲线上的一个除子标量乘算法推广到奇素数域扩域上更一般的此类超椭圆曲线上, 得到了两个效率更高的公式化的除子标量乘新算法. 这两算法所需的运算量比二元法降低 12% 以上.

关键词: 超椭圆曲线; 超椭圆曲线密码体制; 单除子; 除子标量乘; 算法
中图分类号: TP918.1 文献标识码: A 文章编号: 0372-2112 (2008) 10-2049-06

Fast Divisor Scalar Multiplications on a Class of Hyperelliptic Curves

YOU Lin
(*College of Communication Engineering, Hangzhou Dianzi University, Hangzhou, Zhejiang* 310018, *China*)

Abstract: Divisor scalar multiplication is the key operation in hyperelliptic curve cryptosystem. Based on the idea of simple divisor scalar multiplications, Duursma and Sakurai's algorithm for divisor scalar multiplications on a special class of hyperelliptic curves over prime fields has been improved to a larger class of such hyperelliptic curves over prime-extension fields, and two new formulized algorithms for divisor scalar multiplications are proposed. Compared with binary method, our algorithms are much more efficient and take at least 12% less computation amount.

Key words: hyperelliptic curve; hyperelliptic curve cryptosystems; simple divisor; divisor scalar multiplication; algorithm

1 引言

设 F_q 是有限域, q 是某素数方幂. 定义在 F_q 上的亏格 g 的超椭圆曲线 C 指由下列方程所定义的曲线:

$$v^2+h(u)v=f(u) \tag{1}$$

这里 $h(u), f(u)\in F_q[u]$, $\deg_u(h)\leqslant g$, $\deg_u(f)=2g+1$, 且不存在 $(u,v)\in\overline{F}\times\overline{F}$ 同时满足方程 $v^2+h(u)v=f(u)$ 及偏微分方程 $2v+h(u)=0$ 与 $h'(u)v-f'(u)=0$. 如果 $\mathrm{char}(F_q)$ 是奇数, 则 $h(u)$ 可取为 0.

在超椭圆曲线密码体制中, 最关键的运算是超椭圆曲线 Jacobian 群 $J_C(F_{q^n})$ 上的除子标量乘, 即对除子 $D\in J_C(F_{q^n})$ 及正整数 $m\geqslant 1$, 计算 $mD=\underbrace{D+D+\cdots+D}_{m}$, 常规的算法是二元方法, 其平均运算量大约是 $\log_2(m)/2$ 次除子加与 $\log_2(m)$ 次除子倍加. 实际应用中, m 一般接近于群 $J_C(F_{q^n})$ 的阶或 q^{gn}, 所以计算 mD 平均大约需要 $\frac{1}{2}g^n\log_2 q$ 次除子加与 $g^n\log_2 q$ 次除子倍加.

设 C_q 是定义在 F_q 上的曲线

$$C_q: v^2=u^p+au+b \tag{2}$$

其中 $a\neq 0$, q 是素数 p 的方幂, 则 C_q 是亏格为 $(p-1)/2$ 的超椭圆曲线.

Duursma[1]研究了 q 等于素数 p 及 $a=-1 \bmod p$ 时的超椭圆曲线 C_p, 得到了有关除子标量乘运算的快速算法. 下面我们首先引入单除子的概念, 得出单除子标量乘的公式化算法, 此后利用该算法将 Duursma 的结果推广到更一般的超椭圆曲线 C_q 上, 得到了两个效率更高的可实际应用于超椭圆曲线密码体制的公式化新算法. 最后讨论了可实际应用于密码体制的若干 C_q 型曲线.

2 单除子及单除子标量乘

定义 1 一个约化除子 $D=\langle a(u), b(u)\rangle\in J(F)$ 称作一个素除子, 如果 $a(u)$ 是 $F[u]$ 中的一个不可约多项式. 显然, 对超椭圆曲线 C 上任何一个有限点 $P=(x_P, y_P)$, $\langle x-x_P, y_P\rangle$ 是一个素除子, 称该素除子为单除子.

定理 1 设 C 是由方程(1)所定义的超椭圆曲线. 则每个除子 $\langle\alpha(u), \beta(u)\rangle\in J(F)$ 均可分解成若干素除子之和. 即存在有限个素除子 $\langle a_i(u), b_i(u)\rangle$ 使得

$$\langle\alpha(u), \beta(u)\rangle=\sum_i m_i\langle a_i(u), b_i(u)\rangle$$

这里 m_i 是正整数.

收稿日期: 2006-03-16; 修回日期: 2008-05-16
基金项目: 国家自然科学基金(No. 60763009); 教育部科学技术研究重点项目(No. 207089); 海南省自然科学基金(No. 80528)

证明 设 $\alpha(u)=\prod_i a_i^{m_i}(u)$ 是多项式 $\alpha(u)$ 在 $F[u]$ 中的不可约分解. 取 $b_i(u)=\beta(u) \bmod a_i(u)$. 我们首先证明对每个 i, $\langle a_i(u), b_i(u)\rangle$ 是素除子.

由于 $a_i(u)$ 是不可约的, 所以只需证明 $\langle a_i(u), b_i(u)\rangle$ 是除子, 即依据文献[2]中推论 2, 只要证明

$$a_i(u) \mid (b_i^2(u)+b_i(u)h(u)-f(u))$$

因为 $b_i(u)=\beta(u) \bmod a_i(u)$ 及 $\alpha(u) \mid (\beta^2(u)+\beta(u)h(u)-f(u))$, 所以 $b_i^2(u)+b_i(u)h(u)-f(u) \bmod a_i(u)=\beta^2(u)+\beta(u)h(u)-f(u) \bmod a_i(u)=0$. 下面证明

$$\langle \alpha(u), \beta(u)\rangle=\sum_i m_i\langle a_i(u), b_i(u)\rangle$$

由归纳法, 只需证明上等式对 $\alpha(u)=a_1(u)a_2(u)$ (其中 $\gcd(a_1(u), a_2(u))=1$) 及 $\alpha(u)=a_1^2(u)$ 分别成立即可.

(1) 设 $\alpha(u)=a_1(u)a_2(u)$, $\gcd(a_1(u), a_2(u))=1$. 令

$$\langle \alpha_1(u), \beta_1(u)\rangle=\langle a_1(u), b_1(u)\rangle+\langle a_2(u), b_2(u)\rangle$$

其中 $b_1(u)=\beta(u) \bmod a_1(u)$, $b_2(u)=\beta(u) \bmod a_2(u)$. 由 $\gcd(a_1(u), a_2(u))=1$ 知存在 $s_1(u), s_2(u)\in F[u]$, 使得 $s_1(u)a_1(u)+s_2(u)a_2(u)=1$. 又

$$\begin{aligned}&\gcd(a_1(u), a_2(u), b_1(u)+b_2(u)+h(u))\\&=\gcd(\gcd(a_1(u), a_2(u)), b_1(u)+b_2(u)+h(u))\\&=1\end{aligned}$$

且有 $a_1(u)b_2(u)=a_1(u)\beta(u) \bmod a_1(u)a_2(u)=a_1(u)\beta(u) \bmod \alpha(u)$ 及 $a_2(u)b_1(u)=a_2(u)\beta(u) \bmod a_2(u)a_1(u)=a_2(u)\beta(u) \bmod \alpha(u)$. 所以依据除子合成算法[3]可得

$$\begin{aligned}\alpha_1(u)&=a_1(u)a_2(u)=\alpha(u),\\\beta_1(u)&=s_1(u)a_1(u)b_2(u)+s_2(u)a_2(u)b_1(u) \bmod \alpha(u)\\&=(s_1(u)a_1(u)+s_2(u)a_2(u))\beta(u) \bmod \alpha(u)\\&=\beta(u)\end{aligned}$$

(2) 设 $\alpha(u)=a_1^2(u)$. 那么 $\alpha_1(u)=a_1^2(u)/d^2(u)$, 这里 $d(u)=\gcd(a_1(u), 2b_1(u)+h(u))$, 且存在 $s_1(u), s_2(u)\in F[u]$, 使得

$$d(u)=s_1(u)a_1(u)+s_3(u)(2b_1(u)+h(u)),$$

$$\beta_1(u)=\frac{s_1(u)a_1(u)b_1(u)+s_3(u)(b_1^2(u)+f(u))}{d(u)} \bmod \alpha_1(u)$$

因为 $a_1(u)$ 是不可约的, 所以 $d(u)=1$ 或 $a_1(u)$. 如果 $d(u)=a_1(u)$, 则 $2b_1(u)+h(u)=0 \bmod a_1(u)$, 于是

$$\begin{cases}2b_1(u)+h(u)=0 \bmod a_1(u)\\b_1^2(u)+b_1(u)h(u)-f(u)=0 \bmod a_1(u)\\\beta^2(u)+\beta(u)h(u)-f(u)=0 \bmod a_1^2(u)\end{cases}$$

对上面的第三个方程求导得 $\beta'(u)(2\beta(u)+h(u))+(\beta(u)h'(u)-f'(u))=0 \bmod a_1(u)$. 由于 $\beta(u)=b_1(u) \bmod a_1(u)$, 所以得 $b_1(u)h'(u)-f'(u)=0 \bmod a_1(u)$. 如果 x_P 是 $a_1(u)$ 的一个零点且 $b_1(x_P)=y_P$, 则 $P=(x_P, y_P)$ 曲线 C 上的一个点, 且满足

$$\begin{cases}2y_P+h(x_P)=0\\y_P^2+y_Ph(x_P)-f(x_P)=0\\y_Ph'(x_P)-f'(x_P)=0\end{cases}$$

这说明 $P=(x_P, y_P)$ 是 C 上的一个奇异点, 它与 C 是一条非奇异的超椭圆曲线矛盾. 因此 $d(u)=1$, $\alpha_1(u)=a_1^2(u)=\alpha(u)$, $\beta_1(u)=s_1(u)a_1(u)b_1(u)+s_3(u)(b_1^2(u)+f(u)) \bmod a_1^2(u)$, 这里 $s_1(u)a_1(u)+s_3(u)(2b_1(u)+h(u))=1$. 由 $\beta(u)=b_1(u) \bmod a_1(u)$ 可得 $(\beta(u)-b_1(u))^2=0 \bmod a_1^2(u)$ 及 $\beta(u)a_1(u)=b_1(u)a_1(u) \bmod a_1^2(u)$, 于是

$$\begin{aligned}\beta_1(u)&=s_1(u)a_1(u)\beta(u)+s_3(u)(b_1^2(u)+f(u)) \bmod a_1^2(u)\\&=(\beta(u)-\beta(u)s_3(u)(2b_1(u)+h(u)))\\&\quad+s_3(u)(b_1^2(u)+f(u)) \bmod a_1^2(u)\\&=\beta(u)+s_3(u)((\beta(u)-b_1(u))^2-(\beta^2(u)\\&\quad+\beta(u)h(u)-f(u))) \bmod a_1^2(u)\\&=\beta(u)\end{aligned}$$

这就证明了 $\langle \alpha_1(u), \beta_1(u)\rangle=\langle \alpha(u), \beta(u)\rangle$.

利用定理 1, 很容易证得下面的定理 2.

定理 2 设 $D=\langle u-x, y\rangle=J_C(F_{q^n})$, q 是素数 p 的方幂, m 是正整数, 则有

(1) 如果 $y=0$, 则 $m\langle u-x, y\rangle=\langle u-x, y\rangle$ (m 为奇数) 或 $\langle 1, 0\rangle$ (m 为偶数).

(2) 如果 $y\neq 0$, 则当 $2\leqslant m\leqslant g$ 时有

$$m\langle u-x, y\rangle=\langle (u-x)^m, \sum_{i=0}^{m-1}a_i(u-x)^i\rangle$$

而当 $p\geqslant m>(p-1)/2$ 时有

$$m\langle u-x, y\rangle\sim\langle (u-x)^m, \sum_{i=0}^{m-1}a_i(u-x)^i\rangle$$

这里 $a_0=y$, $a_1=\frac{f'(x)}{2y}$, $a_i=\frac{1}{2y}\left(\frac{f^{(i)}(x)}{i!}-\sum_{\substack{j+k=i\\1\leqslant j,k}}a_ja_k\right)$, $2\leqslant i\leqslant m-1$.

且对于 $p\geqslant m>(p-1)/2$, $\langle (u-x)^m, \sum_{i=0}^{m-1}a_i(u-x)^i\rangle$ 是一个半约化除子.

设 C_5 是 F_{5^n} 上的超椭圆曲线: $v^2=u^5+au+b$, $a\neq 0$. 对任意点 $P=(x, y)\in C_5(F_{5^n})$, $y\neq 0$ 可得

$$2\langle u-x, y\rangle=\langle (u-x)^2, \frac{3a}{y}(u-x)+y\rangle$$

$$4\langle u-x, y\rangle=\langle (u-x)^2+\left(\frac{y^2}{a}-\frac{y^{10}}{a^6}\right)(u-x),$$

$$\left(-\frac{\gamma^{15}}{a^9}-\frac{\gamma^7}{a^4}-\frac{a}{y}\right)(u-x)-y\rangle$$

$$5\langle u-x,y\rangle=\langle(u-x)+\left(\frac{\gamma^2}{a}-\frac{\gamma^{10}}{a^6}\right),-\frac{\gamma^{25}}{a^{15}}\rangle$$

一般地, 对曲线 C_q, 有下面的定理 3.

定理 3　设 $P=(x,y)\in C_q(F_{q^n})$, $y\neq 0$, 则

$$p\langle u-x,y\rangle=\langle u-x_0,y_0\rangle$$

其中 $x_0=a^{-p-1}(x^{p^2}-a^pb+b^p)$, $y_0=-a^{-\frac{p(p+1)}{2}}y^{p^2}$.

证明　令 $G(u,v)=(x^p+au+b)^{\frac{p+1}{2}}-y^pv$ $(\in F_{q^n}[u,v])$. 则 $G(u,v)$的范数[4] 是 $N(G(u,v))$

$$=(x^p+au+b)^{p+1}-y^{2p}(u^p+au+b)$$
$$=(x^p+au+b)^{p+1}-(x^{p^2}+d^px^p+b^p)(u^p+au+b)$$
$$=(x^{p^2}+a^pu^p+b^p)(x^p+au+b)-((x^{p^2}+d^pu^p+b^p)-a^p(u^p-x^p))((x^p+au+b)(u^p-x^p))$$
$$=(u-x)^p(a^{p+1}u-(x^{p^2}-a^pb+b^p))$$

注意到(x,y)与$(x_0,-y_0)$是 $G(u,v)$ 的零点, 而 $(x,-y)$与(x_0,y_0)非 $G(u,v)$的零点. 因此

$$\begin{aligned}\mathrm{div}(G)&=\mathrm{ord}_{(x,y)}(G)+\mathrm{ord}_{x_0,-y_0}(G)\\&=p(x,y)+(x_0,-y_0)-(p+1)\infty\\&=(p(x,y)-p\infty)+((x_0,-y_0)-\infty)\end{aligned}$$

因此得

$$p\langle u-x,y\rangle=-\langle u-x_0,-y_0\rangle=\langle u-x_0,y_0\rangle$$

算法 1

输入: 单除子 $D=\langle u-x,y\rangle\in J_{C_q}(F_{q^n})$, $y\neq 0$, 正整数 r: $2\leqslant r\leqslant p-1$.

输出: $2D, 3D, \cdots, (p-1)D$.

(1) 初始化 $s_1:=s_2:=1$, $x_0:=a^{-p-1}(x^{p^2}-a^pb+b^p)$, $y_0:=-a^{-\frac{p(p+1)}{2}}y^{p^2}$;

(2) 对于 $2\leqslant r\leqslant p-2$, 执行

(a) $s_r:=\sum_{\substack{j+k=r\\1\leqslant j,k}}s_js_k$;

(b) 初始化 $b_r(u):=y$ 对 j 自 1 至 $r-1$ 执行

$$b_r(u):=b_r(u)+(-1)^{j-1}\frac{s_ja^j}{(2y)^{2j-1}}(u-x)^j$$

(3) 对 $2\leqslant r\leqslant(p-1)/2$, 输出$\langle(u-x)^r, b_i(u)\rangle$为 rD;

(4) 对$(p+1)/2\leqslant r\leqslant p-1$, 计算$\langle u-x_0,y_0\rangle+\langle(u-x)^{p-r},-b_{p-r}(u)\rangle$, 并将计算结果输出为 rD.

利用定理 3 及算法 1, 可得到计算单除子标量乘的算法 2.

算法 2　计算曲线 C_q 上的单除子标量乘.

输入: 单除子 $D=\langle u-x,y\rangle\in C_q(F_{q^n})$, $y\neq 0$, 正整数 m;

输出: mD.

(1) 将 m 转换成$p-$ 进制形式: $m=\sum_{i=0}^{l-1}r_ip^i$, 其中 $0\leqslant r_i\leqslant p-1$.

(2) 对 $i=1, \cdots, l-1$, 计算 p^iD 并存储为D_i:

(a) 初始化 $x_0:=x$, $y_0:=y$, $D_0:=D$.

(b) 对 i 自 1 至 $l-1$ 执行:

①$x_i:=a^{-p-1}(x_{i-1}^{p^2}-a^pb+b^p)$,

$y_i:=-a^{-\frac{p(p+1)}{2}}y_{i-1}^{p^2}$;

②$D_i:=\langle u-x_i,y_i\rangle$.

(3) 初始化 $B:=\langle 0,1\rangle$.

(4) 对 i 自 1 至 $l-1$ 执行 $B:=B+r_iD_i$.

(5) 输出 B 为mD.

显然, 对于单除子的标量乘, 算法 2 优于比二元方法.

3　C_q 型曲线上一般除子标量乘的计算

设 $D=\langle\alpha(u),\beta(u)\rangle\in J_{C_q}(F_q)$且 $\alpha(u)=(u-x_1)^{k_1}\cdots(u-x_n)^{k_n}$是$\alpha$ 在其分裂域$F_q\langle\alpha(u)\rangle$中的一次因式分解. 由定理 1 可知在 Jacobian 群 $J_{C_q}(F_q\langle\alpha(u)\rangle)$ 中有

$$mD=m\langle\alpha(u),\beta(u)\rangle=\sum_{i=1}^{l}(mk_i)\langle u-x_i,y_i\rangle$$

其中 $y_i=\beta(u)\bmod(u-x_i)$. 于是可得计算 mD 的算法 3.

算法 3

输入: 正整数 m, 除子 $D=\langle\alpha(u),\beta(u)\rangle\in J_{C_p}(F_q)$;

输出: mD.

(1) 利用多项式分解算法[5]在 $F_q\langle\alpha(u)\rangle$中将 $\alpha(u)$ 分解成一次因式的乘积:

$$\alpha(u)=(u-x_1)^{k_1}\cdots(u-x_n)^{k_n},\ x_1,\ \cdots,\ x_n$$

(2) 对 $i=1, 2, \cdots, n$, 计算 $y_i=\beta(u)\bmod(u-x_i)$;

(3) 利用算法 2 计算单除子标量乘$(mk_i)\langle u-x_i, y_i\rangle$, 并将结果记为 D_i;

(4) 初始化 $B=\langle 1,0\rangle$;

(5) 对 i 自 1 至 n 执行$B:=B+D_i$;

(6) 输出 B.

定理 4　设 q 是素数p 的方幂. 则对任意正整数 t 及 C_q 上的任意除子

$$D=\langle\alpha(u),\beta(u)\rangle=\langle\sum_{0\leqslant j\leqslant g}a_{g-j}u^j,\sum_{0\leqslant k\leqslant g-1}b_{g-k-1}u^k\rangle\in J_{C_q}(F_{q^n})$$

有 $p^tD=\langle\sum_{0\leqslant j\leqslant g}a_{g-j}^{p^{2t}}a^{-(g-j)(p^{2t}-1)/(p-1)}(u-tc)^j$, $(-1)^ta^{-tp(p+1)/2}\sum_{0\leqslant k\leqslant g-1}b_{g-k-1}^{p^{2t}}a^{k(p^{2t}-1)/(p-1)}(u-tc)^k\rangle$成

立, 这里 $c=a^{-1}(-b+(a^{-1}b)^p)$, $g=(p-1)/2$.

证明 只须证明 $\alpha(u)$ 是 F_q 上的一个二次多项式时该定理成立, 然后利用定理 3 及算法 3 的思想, 即可证明当 $\alpha(u)$ 是更高次的多项式时, 该定理仍然成立.

不妨设 $\alpha(u)=u^2+a_1u+a_2=(u-x_1)(u-x_2)$, $\beta(u)=b_0u+b_1$, 其中 $x_1, x_2\in\overline{F}_q$. 则 $y_i=b(u) \bmod (u-x_i)=b_0x_i+b_1$. 于是由定理 3, 当 $x_1\neq x_2$ 时, 可得

$$\begin{aligned}pD&=p\langle u-x_1, y_1\rangle+p\langle u-x_2, y_2\rangle\\&=\langle u-a^{-(p+1)}(x_1^{p^2}-a^pb+b^p), -a^{-p(p+1)/2}y_1^{p^2}\rangle\\&\quad+\langle u-a^{-(p+1)}(x_2^{p^2}-a^pb+b^p), -a^{-p(p+1)/2}y_2^{p^2}\rangle\\&=\langle (u-c)^2+(-a^{-(p+1)})(x_1+x_2)^{p^2}(u-c)\\&\quad+a^{-2(p+1)}(x_1x_2)^{p^2}, -a^{-p(p+1)/2}(x_1-x_2)^{-p^2}(a^{p+1}\\&\quad\cdot(y_1-y_2)^{p^2}(u-c)+(x_1y_2-x_2y_1)^{p^2})\rangle\\&=\langle (u-c)^2+a^{-(p+1)}a_1^{p^2}(u-c)+a^{-2(p+1)}a_2^{p^2},\\&\quad-a^{-p(p+1)/2}(a^{p+1}b_0^{p^2}(u-c)+b_1^{p^2})\rangle\end{aligned}$$

如果 $x_1=x_2$, 则

$$\begin{aligned}pD&=p\langle u-x_1, y_1\rangle+p\langle u-x_1, y_1\rangle\\&=2\langle u-a^{-p(p+1)}(x_1^{p^2}-a^pb+b^p), -a^{-p(p+1)/2}y_1^{p^2}\rangle\\&=\langle (u-a^{-(p+1)}(x_1^{p^2}-a^pb+b^p))^2, \beta_1(u)\rangle\end{aligned}$$

这里

$$\begin{aligned}\beta_1(u)&=(a/2((-a^{-p(p+1)/2}y_1^{p^2})))(u-c-a^{-(p+1)}x_1^{p^2})\\&\quad+(-a^{-p(p+1)/2}y_1^{p^2})\end{aligned}$$

由 $(b_0u+b_1)^2-(u^p+au+b)=0 \bmod (u-x_1)^2$ 及 $y_1^2=x_1^p+ax_1+b$ 可得 $2b_0(b_0x_1+b_1)=a$, 即 $2b_0y_1=a$. 因此

$$\begin{aligned}\beta_1(u)&=-(a/2)a^{p(p+1)/2}b_0^{p^2}(a/2)^{-p^2}(u-c-a^{-(p+1)}x_1^{p^2})\\&\quad+(-a^{-p(p+1)/2}y_1^{p^2})\\&=-a^{-(p-2)(p+1)/2}b_0^{p^2}(u-c-a^{-(p+1)}x_1^{p^2})\\&\quad+(-a^{-p(p+1)/2}y_1^{p^2})\\&=-a^{-p(p+1)/2}(a^{p+1}b_0^{p^2}(u-c)+(-b_0^{p^2}x_1^{p^2}+y_1^{p^2}))\\&=-a^{-p(p+1)/2}(a^{p+1}b_0^{p^2}(u-c)+b_1^{p^2})\end{aligned}$$

依据定理 3 与算法 3, 利用归纳法即可证得本定理成立.

由定理 4, 可得下面的算法 4 与算法 5.

算法 4

输入: 正整数 m, 除子

$$D=\langle\alpha(u), \beta(u)\rangle=\langle\sum_{0\leqslant j\leqslant g}a_{g-j}u^j, \sum_{0\leqslant k\leqslant g-1}b_{g-k-1}u^k\rangle\in J_{C_q}(F_{q^n})$$

输出: mD.

(1) 将 m 转换成 $p-$ 进制形式:

$$m=r_{l-1}p^{l-1}+\cdots+r_1p+r_0$$

其中 $0\leqslant r_i\leqslant p-1$, $r_{p-1}\neq 0$, $l=\lfloor\log_p m\rfloor+1$;

(2) 对于 $t=1, \cdots, l-1$, 利用定理 4 计算 p^tD 并将结果存储为 D_t;

(3) 对 i 自 0 至 $l-1$ 执行

(a) 初始化 $B:=\langle 0, 1\rangle$; (b) $B:=B+r_iD_i$.

(4) 输出 B 为 mD.

算法 5

输入: 正整数 m, 除子

$$D=\langle\alpha(u), \beta(u)\rangle=\langle\sum_{0\leqslant j\leqslant g}a_{g-j}u^j, \sum_{0\leqslant k\leqslant g-1}b_{g-k-1}u^k\rangle\in J_{C_q}(F_{q^n})$$

输出: mD.

(1) 预计算 $2D, 3D, \cdots, (p-1)D$:

(a) $D_0:=\langle 1, 0\rangle$;

(b) 对于 i 自 1 至 $p-1$ 置 $D_i:=D_{i-1}+D$.

(2) 将 m 转换成 $p-$ 进制形式:

$$m=r_{l-1}p^{l-1}+\cdots+r_1p+r_0$$

其中 $0\leqslant r_i\leqslant p-1$, $r_{p-1}\neq 0$, $l=\lfloor\log_p m\rfloor+1$;

(3) 初始化 $B:=D_{r_{l-1}}$;

(4) 对于 i 自 $l-2$ 递减到 0, 执行

(c) 置 $B:=\langle\sum_{0\leqslant j\leqslant g}s_{g-j}u^j, \sum_{0\leqslant k\leqslant g-1}t_{g-k-1}u^k\rangle$;

(d) 置 $\overline{B}:=\langle\sum_{0\leqslant j\leqslant g}s_{g-j}^{p^2}a^{-(g-j)(p+1)}(u-c)^j, -a^{-p(p+1)/2}\sum_{0\leqslant k\leqslant g-1}t_{g-k-1}^{p^2}a^{k(p+1)}(u-c)^k\rangle$;

(e) $B:=\overline{B}+D_{r_i}$.

(5) 输出 B 为 mD.

下面利用文献[6]中定理 14 来分别计算算法 4, 算法 5 及二元法所需的运算量. 这里假设 $q=q^n\gg p$, 于是定理 14 中的小量 $\frac{1}{q}O(g^3)$ 可忽略不计.

对于固定的曲线 C_q 及除子 D, 上面算法中的 c, $a^{-p(p+1)/2}$ 与 $a^{\pm k/(p-1)}(1\leqslant k\leqslant g)$ 可预先计算出, 所以它们的运算量也可忽略. 用 M 与 I 分别表示 F_q 上的 1 次乘运算与 1 次取逆运算.

算法 4 的运算量:

计算 $p^tD(1\leqslant t\leqslant l-1)$ 需要的运算量大约是

$$g(l-1)I+\left(\frac{1}{4}(l-1)(p^2+2p+5+6(p-1))\log_2(p)\right)M$$

该算法中循环计算步骤的运算量, 平均需要

$\frac{(p-1)l}{2p}(\log_2(p-1)+1)-1$ 次除子倍加

及 $\frac{(p-1)l}{p}(\log_2(p-1)-1)$ 次除子加.

算法 5 的运算量:

该算法需要大约 $(p-1)l/p$ 次除子加及乘子为 p 的 $l-1$ 次除子标量乘, 再加上预计算.

乘子为 p 的 $l-1$ 次除子标量乘的运算量是 $\frac{1}{4}(l-1)(p^2+4p-5)M$. 预计算步骤需要 $(p-3)$ 次除子加及一次除子倍加. 因此, 假设 $m\sim\#J(F_{q^n})$, 即 $m\sim q^{gn}=p^{gnk}$, 则对 $p=5,7,11,13,17$, 可得运算量表 1.

表 1　曲线 C_q 上三种算法计算除子标量乘 mD 的运算量

P	二元法	算法 4	算法 5
5	$250nkM+19nkI$	$(167nk-95)M+(14nk-8)I$	$(67nk+213)M+(5nk+18)I$
7	$698nkM+31nkI$	$(523nk-210)M+(25nk-11)I$	$(162nk+835)M+(7nk+40)I$
11	$2292nkM+54nkI$	$(1898nk-540)M+(48nk-16)I$	$(450nk+4059)M+(10nk+99)I$
13	$3440nkM+63nkI$	$(2921nk-752)M+(58nk-18)I$	$(641nk+6985)M+(11nk+132)I$
17	$6681nkM+88nkI$	$(5851nk-1301)M+(83nk-23)I$	$(1143nk+16815)M+(15nk+225)I$

表 1 表明算法 5 所需的运算量比二元法要少得多, 对于 p 取为 5 至 17 的素数时, 要少大约 70% 至 83% 的运算量. 而算法 4 所需的运算量比二元法大约要少大约 33% 至 12%.

下面用两个具体例子来说明上述两算法及二元算法之间的效率性能.

例 **1**　设 C_7 是域 F_7 上的超椭圆曲线: $v^2=u^7-u+1$. 设 $a(u)=u^3+u^2+2$ 及 $b(u)=6u^2+6u$, 则 $D=\langle a(u),b(u)\rangle\in J_{C_7}(F_7)$.

如果用二元法来计算标量乘 $7D$, 则需要 2 次除子倍加及 2 次除子加:

$$\begin{aligned}
2D &= 2^2D+2D+D\\
&= 2(2\langle u^3+u^2+2,6u^2+6u\rangle)\\
&\quad +2\langle u^3+u^2+2,6u^2+6u\rangle\\
&\quad +\langle u^3+u^2+2,6u^2+6u\rangle\\
&= 2\langle (u^2+6u+4)(u+5),5u+5\rangle\\
&\quad +\langle (u^2+6u+4)(u+5),5u+5\rangle\\
&\quad +\langle u^3+u^2+2,6u^2+6u\rangle\\
&= (\langle (u+1)(u+4)(u+6),5u^2+u+2\rangle\\
&\quad +\langle (u^2+6u+4)(u+5),5u+5\rangle)\\
&\quad +\langle u^3+u^2+2,6u^2+6u\rangle\\
&= \langle (u+3)(u^2+3u+6),u^2+5u+5\rangle\\
&\quad +\langle u^3+u^2+2,6u^2+6u\rangle\\
&= \langle (u+3)(u^2+6u+4),u^2+4u+2\rangle
\end{aligned}$$

现在用算法 3 来计算 $7D$:

(1) 在 $F_7\langle a(u)\rangle(=F_{7^2})$ 上分解 $a(u)=u^3+u^2+2$ 得 $a(u)=(u+4\alpha)(u+3\alpha+3)(u+5)$, 这里 α 是 $x^2+x+3\bmod 7$ 的一个零点.

(2) 对 $x_i=3\alpha,4\alpha+4,2$ 分别计算 $y_i=b(u)\bmod(u-x_i)$, 得 $y_1=-\alpha-1$, $y_2=\alpha$, $y_3=1$.

(3) 计算 $7D=7\langle u^3+u^2+2,6u^2+6u\rangle$ 如下:

$$\begin{aligned}
7D &= 7\langle u^3+u^2+2,6u^2+6u\rangle\\
&= 7\langle u-3\alpha,-\alpha-1\rangle+7\langle u-4\alpha-4,\alpha\rangle\\
&\quad +7\langle u-2,1\rangle\\
&= \langle u-(3\alpha)^{49}-2,-(\alpha-1)^{49}\rangle\\
&\quad +\langle u-(4\alpha+4)^{49}-2,-\alpha^{49}\rangle\\
&\quad +\langle u-2^{49}-2,-1\rangle\\
&= \langle u-3\alpha-2,\alpha+1\rangle+\langle u-4\alpha-6,-\alpha\rangle+\langle u-4,-1\rangle\\
&= \langle u^2+6u+4,5u+5\rangle+\langle u-4,-1\rangle\\
&= \langle (u+3)(u^2+6u+4),u^2+4u+2\rangle\\
&= \langle u^3+2u^2+u+5,u^2+4u+2\rangle
\end{aligned}$$

例 1 表明计算标量乘 $7D$ 时, 算法 3 比二元法要有效得多. 确切地说, 设 M 与 I 分别表示上 F_7 的乘与取逆运算, M_1 与 I_1 分别表示 F_{7^2} 上的乘与取逆运算. 那么计算标量乘 $7D$, 二元法需要的运算量为 $585M+17I$, 而算法 3 需要的运算量是 $30M+1I$ 及 $46M_1+1I_1$, * 再加上在 F_{p^2} 上因式分解 u^3+u^2+2 及计算 $y_i=b(u)\bmod(u-x_i)$ 所需的计算量 $16M+26M_1$. 由于 $F_{7^2}\setminus F_7$ 中元的 1 次乘与 1 次取逆运算分别等于 $5M$ 与 $6M+1I$ 的运算量. 算法 3 需要的运算量大约是 $412M+2I$, 这比二元法的运算量要少大约 40.1%.

如果利用定理 4 来直接计算 $7D$, 则需要的运算量仅是 $1I+17M$:

$$\begin{aligned}
&7\langle u^3+u^2+2,6u^2+6u\rangle\\
&= \langle 2^{\vec{7}}(-1)^{(-3^{\bullet}8)}(u-(-1)^{-1}(-1+(-1^{\bullet}1)^7))^0\\
&\quad +1^{\vec{7}}(-1)^{(-(3-2)-8)}(u-(-1)^{-1}(-1+(-1^{\bullet}1)^7)^2\\
&\quad +1^{\vec{7}}(-1)^{(-(3-3)^{\bullet}8)}(u-(-1)^{-1}(-1+(-1^{\bullet}1)^7))^3,\\
&\quad -(-1)^{(-7^{\bullet}4)}((6^{\vec{7}}(-1)^{1^{\bullet}8}(u-(-1)^{-1}(-1+(-1^{\bullet}1)^7))^1\\
&\quad +6^{\vec{7}}(-1)^{2^{\bullet}8}(u-(-1)^{-1}(-1+(-1^{\bullet}1)^7)))^2)\rangle\\
&= \langle u^3+2u^2+u+5,u^2+4u+2\rangle
\end{aligned}$$

例 **2**　设 C_{7^2} 是域 F_{7^2} 上的曲线: $v^2=u^7+\theta u+2$, θ 是 $x^2+x+3=0\bmod 7$ 的一个零点. 则

$$D=\langle u^3+(6\theta+3)u^2+(3\theta+6)u+3\theta,u^2+2u+1\rangle\in J_{C_{7^2}}(F_{7^2})$$

* 注意到对每个元 $\beta\in F_{7^2}$, 计算 β^{49} 一般需要的运算量是 $7M_1$, 这是因为 $\beta^{49}=(\beta^{2^4})^2\beta^{2^4}\beta$. 而在例 1 中, 因为 $\beta^{49}=\beta$, 所以 β^{49} 的计算量可忽略不记.

且 $u^3+(6\theta+3)u^2+(3\theta+6)u+3\theta$ 在 F_{7^2} 上是不可约的.

下面分别用二元法, 算法 4 及算法 5 来计算 $67D$.

用二元法计算 $67D$ 需要 $24I+1052M$:

$$\begin{aligned}67D &= 2^6D+2D+D\\ &= \langle (u+2\theta+1)(u^2+(6\theta+3)u+5\theta+3),\\ &\quad (4\theta+3)u^2+(3\theta+2)u+\theta+3\rangle\\ &\quad +\langle u^3+(5\theta+1)u^2+(2\theta+2)u+2\theta+2,\\ &\quad (\theta+2)u^2+(3+6\theta)u+2\theta+2\rangle\\ &\quad +\langle u^3+(6\theta+3)u^2+(3\theta+6)u+3\theta,\\ &\quad u^2+2u+1\rangle\\ &= \langle u^3+(4\theta+3)u^2+(\theta+5)u+2\theta+5,\\ &\quad u^2+(\theta+5)u+3\theta+4\rangle.\end{aligned}$$

用算法 4 计算 $67D$ 需要 $17I+746M$:

$$\begin{aligned}67D &= 7^2D+2(7D)+2(2D)\\ &= \langle 3^{7^4}\theta^{-3(7^4-1)6^{-1}}+(3\theta+6)^{7^4}\theta^{-2(7^4-1)6^{-1}}(u-2\theta^{-1}(-2+(\theta^{-1}2)^7))\\ &\quad +(6\theta+3)^{7^4}\theta^{-(7^4-1)6^{-1}}(u-2\theta^{-1}(-2+(\theta^{-1}2)^7))^2+(u-2\theta^{-1}(-2+(\theta^{-1}2)^7))^3,\\ &\quad 1+2^{7^4}\theta^{(7^4-1)6^{-1}}(u-2\theta^{-1}(-2+(\theta^{-1}2)^7))\\ &\quad +\theta^{2(7^4-1)6^{-1}}(u-2\theta^{-1}(-2+(\theta^{-1}2)^7))^2\rangle\\ &\quad +2\langle 3^{7^2}\theta^{-3(7^2-1)6^{-1}}+(3\theta+6)^{7^2}\theta^{-2(7^2-1)6^{-1}}\\ &\quad \cdot(u-\theta^{-1}(-2+(\theta^{-1}2)^7))\\ &\quad +(6\theta+3)^{7^2}\theta^{-(7^2-1)6^{-1}}(u-\theta^{-1}(-2+(\theta^{-1}2)^7))^2+(u-\theta^{-1}(-2+(\theta^{-1}2)^7))^3,\\ &\quad 1+2^{7^2}\theta^{(7^2-1)6^{-1}}(u-\theta^{-1}(-2+(\theta^{-1}2)^7))\\ &\quad +\theta^{2(7^2-1)6^{-1}}(u-\theta^{-1}(-2+(\theta^{-1}2)^7))^2\rangle\\ &\quad +\langle u^3+(\theta+2)u^2+2\theta u+2\theta+3,\\ &\quad (3+3\theta)u^2+(1+2\theta)u+\theta\rangle\\ &= \langle u^3+(2+5\theta)u^2+(2\theta+3)u+(2+\theta),\\ &\quad 6u^2+(\theta+1)u+(4+5\theta)\rangle\\ &\quad +\langle u^3+(2\theta+1)u^2+5u+3\theta+2,\\ &\quad (5\theta+5)u^2+3\theta u+6\theta+3\rangle\\ &\quad +\langle u^3+(\theta+2)u^2+2\theta u+2\theta+6,\\ &\quad (3+3\theta)u^2+(1+2\theta)u+\theta\rangle\\ &= \langle u^3+(4\theta+3)u^2+(5+\theta)u+2\theta+5,\\ &\quad u^2+(5+\theta)u+3\theta+4\rangle.\end{aligned}$$

用算法 5 计算 $67D$ 需要 $15I+649M$:

$$\begin{aligned}67D &= 7((7D)+2D)+4D\\ &= 7\langle u^3+4u^2+(2\theta+5)u+3,\\ &\quad (3\theta+6)u^2+(2\theta+5)u+2\theta+1\rangle\\ &\quad +\langle u^3+(\theta+2)u^2+2\theta u+2\theta+3,\\ &\quad (3+3\theta)u^2+(1+2\theta)u+\theta\rangle\\ &= \langle u^3+(4\theta+3)u^2+(5+\theta)u+2\theta+5,\\ &\quad u^2+(5+\theta)u+3\theta+4\rangle.\end{aligned}$$

从该例子看, 算法 5 是最有效的, 而二元法是最差的. 算法 4 需要的计算量比二元法要少大约 29.1%, 算法 5 则比二元法要少 38.2% 的计算量.

4 结论

通过引入单除子标量乘思想, 我们将 Duursma[1] 计算 C_p 型超椭圆曲线上除子标量乘的算法推广到奇素数域扩域上更一般的 C_q 型超椭圆曲线上, 得到了两个效率更高的计算除子标量乘的公式化新算法. 当 q 为 5 至 17 等素数的方幂时, 这两算法所需的运算量比二元法要降低 12% 以上. 但是对于一些特殊的 C_q 型曲线, 如 F_5 上的曲线 $v^2=u^5+au\ (a=2,3)$, F_{5^2} 上的曲线 $v^2=u^5+vu+1$, 由于其特征多项式分别为 t^4+25 与 t^4+625, 利用文献[7]中的算法 2 则效率会更高.

参考文献:

[1] I Duursma, K Sakurai. Efficient algorithms for the Jacobian variety of hyperelliptic curves $y^2=x^p-x+1$ over a finite field of odd characteristic p [A]. Proceedings of International Conference on Coding Theory, Cryptography and Related Areas [C]. Guanajuato: Springer Verlag, 2000. 73-89.

[2] You Lin, FAN Yun. Jacobian groups of hyperelliptic curves in hyperelliptic cryptosystems[J]. Chinese Journal of Electronics, 2003, 12(4): 642-647.

[3] D Cantor. Computing in the jacobian of a hyperelliptic curve [J]. Mathematics of Computation, 1987, 48(177): 95-101.

[4] N Koblitz. Algebraic Aspects of Cryptography [M]. Berlin: Springer Verlag Press, 1998. 159-168.

[5] Joachim von zur Gathen, Jü rgen Gerhard. Modern Computer Algebra[M]. Cambridge: Cambridge University Press, 1999. 353-386.

[6] A Enge. The extended euclidean algorithm on polynomials, and the computational efficiency of hyperelliptic cryptosystems[J]. Des. Codes Cryptography, 2001, 23(1): 53-74.

[7] You Lin, et al. Speeding up scalar multipications on hyperelliptic curves by making use of frobenius endomorphism[J]. Chinese Journal of Electronics, 2006, 15(1): 123-128.

作者简介:

游 林 男, 1966 年生于江西临川, 博士, 杭州电子科技大学通信工程学院教授. 主要研究兴趣为密码学、代数学及其应用. E-mail: mryoulin@gmail.com