超椭圆曲线上的 Jacobian 加法算法的优化

陈智雄[1,2]，黄振杰[1]，肖国镇[1]

(1. 西安电子科技大学 综合业务网理论与关键技术国家重点实验室,陕西 西安 710071；
2. 莆田学院 数学系,福建 莆田 351100)

摘要：计算超椭圆曲线上的 Jacobian 群中两个元素 D 和 E 的运算 $2D+E$ 是标量乘法的重要过程,该运算通常分为一个倍点和一个加法进行两次计算. 通过优化 T. Lange 的计算公式,直接计算 $2D+E$,减少了中间结果的计算量,使得计算效率提高 6% ~8%. 该方案也可用于超椭圆曲线密码体制的数字签名验证和 Weil/Tate 对的计算.

关键词：超椭圆曲线密码；Jacobian 加法算法；Harley 算法

中图分类号：TN911； **文献标识码**：A **文章编号**：1001-2400(2005)06-0922-05

An optimized algorithm for Jacobians of hyperelliptic curves

CHEN Zhi-xiong[1,2], *HUANG Zhen-jie*[1], *XIAO Guo-zhen*[1]

(1. State Key Lab. of Integrated Service Networks, Xidian Univ., Xi'an 710071, China;
2. Dept. of Mathematics, Putian Univ., Putian 351100, China)

Abstract: In hyperelliptic curve cryptosystems, it is one of the main steps in computing scalar multiplication to evaluate $2D+E$ from the given elements D and E in the Jacobian of a hyperelliptic curve. In this paper, an optimized method for computing $2D+E$ is proposed, which will raise the computing efficiency by about 6% ~8%.

Key Words: hyperelliptic curve cyrptography; Jacobian; addition algorithm; Harley algorithm

基于椭曲曲线的密码体制(ECC)已经得到广泛的应用,作为椭圆曲线的一个推广,N. Koblitz[1] 在 1989 年提出的用超椭圆曲线(HCC)来建立公钥密码体制,其安全性基于有限域上超椭圆曲线的 Jacobian 群的离散对数问题. 研究表明,超椭圆曲线密码要比椭圆曲线密码优越,在同等安全等级下,前者所用的基域要小,所供选择的曲线越多. 但越椭圆曲线密码体制目前还处于理论研究阶段,理论尚未成熟,还有很多问题尚待解决. 一个重要的问题是超椭圆曲线的 Jacobian 群运算(两个除子的和)相当复杂,另一个问题是随机生成适合密码使用的安全曲线相当困难. 但是,超椭圆曲线密码体制比其他密码体制有许多优点,因此它已受到高度的关注. 从已有 HCC 的实现看,其实现速度比 ECC 慢,有关 ECC 的实现可参见文[2]等,因此如何减少超椭圆曲线的 Jacobian 群的加法和标量乘的计算量,有效提高超椭圆曲线密码的实现速度对超椭圆曲线密码走向实用具有极其重要的意义.

超椭圆曲线的 Jacobian 群运算算法主要包括 Cantor[3] 算法和 Harley[4] 算法. Cantor 算法分复合(composition)与归约(reduction)两个过程. Harley 充分利用中国剩余定理和牛顿迭代法提出了一个更为简单的算法,对亏格为 2 的超椭圆曲线计算加法(addition)和倍点(doubling)分别需要 $2I+27M$ 和 $2I+30M$ 的计算量,这里 I 和 M 及文中的 S 分别表示基域元素的逆运算、乘法运算和平方运算. 在文[5~7]中 Jacobian 群运算也得到进一步的研究,其中突出的是 T. Lange[8] 优化的 Harley 算法,计算加法和倍点分别只需要 $I+3S+22M$ 和 $I+5S+22M$ 的计算量.

收稿日期：2004-12-15

基金项目：973 项目(G1999035804)；福建省自然科学基金资助项目(A0540011)；福建省教育厅科学基金资助项目(JA04264)；莆田市科技计划资助项目(05022)

作者简介：陈智雄(1972-),男,西安电子科技大学博士研究生.

计算标量乘法(scalar multiplication)是公钥密码体制中最重要的计算过程,而计算 $2D+E$ 是该过程的一个步骤之一,笔者对亏格为 2 的超椭圆曲线上的 Jacobian 群运算算法[8]进行优化,使得计算 $2D+E$ 的效率提高 6% ~8%,从而有效提高标量乘法、Weil/Tate 对的计算效率,对基于超椭圆曲线的公钥密码体制的加密与数字签名有积极的意义.

1　超椭圆曲线及其 Jacobian 群运算

设 $F=\mathrm{GF}(q)$ 是一个阶为 q(q 是一个素数的幂) 的有限域,$\bar{F}$ 是 F 的代数闭包.

众所周知,F 上的亏格为 g 的超椭圆曲线 H(参见文[1,9])可由如下 Weierstrass 方程给出:

$$H: y^2+h(x)\,y=f(x)\quad , \tag{1}$$

其中 $h,f\in F[x]$,h 的次数 $\deg(h)$ 至多为 g,f 是一个首一多项式使得 $\deg(f)=2g+1$,且 H 上没有(有限)奇异点:一个奇异点 $(x,y)\in\bar{F}\times\bar{F}$ 同时满足方程(1) 和方程(1) 的两个偏导数方程 $2y+h(x)=0$,$h'(x)y-f'(x)=0$. H 有一个惟一的无限点 ∞. 特别地,当 $g=1$ 时,H 即为椭圆曲线.

一个除子 D 是 H 上的点的有限形式和:$D=\sum_{P\in H} m_P P$,$m_P\in Z$,其中只有有限个整数 m_P 不为零. 除子 D 的次数定义为 $\deg D=\sum_{P\in H} m_P$,D 的支撑集定义为集合 $\mathrm{Supp}(D)=\{P\in H\mid m_P\neq 0\}$. 设 $P=(x,y)\in H$,P 的反点为 $\tilde{P}=(x,-y-h(x))$,如果 $P=\infty$,则 $\tilde{P}=\infty$. 函数 $r\in\bar{F}[x,y]/<H>$ 的除子定义为 $\mathrm{div}(r)=\sum m_P P-(\sum m_P)\infty$,称之为主除子,其中 $P\in H$ 满足 $r(P)=0$ 或 $r(P)=\infty$(前者称为零点 zero,后者称为极点 pole),m_P 为 r 在 P 点的阶.

两个除子的加法运算定义为 $\sum m_P P+\sum n_P P=\sum (m_P+n_P)P$. 于是所有零次除子构成一个加法交换群,记为 $\overline{\overline{D}}^0$,所有主除子构成 $\overline{\overline{D}}^0$ 的一个子群 $\overline{\overline{P}}$,那么商群 $\overline{\overline{D}}^0/\overline{\overline{P}}$ 称为 Jacobian 群. 超椭圆曲线的离散对数问题(HCDLP)是指:给定 Jacobian 中的两个除子 D 和 E,求整数 m 使得 $E=mD$(如果 m 存在).

一个半既约除子形如 $D=\sum m_i P_i-(\sum m_i)\infty$,其中每个 $m_i\geqslant 0$ 且对于 $P_i\in H\setminus\{\infty\}$,若 $P_i\in\mathrm{Supp}(D)$,则 $\tilde{P}_i\notin\mathrm{Supp}(D)$,除非 $P_i=\tilde{P}_i$,此时 $m_i=1$. 称除子 $D=\sum m_i P_i-(\sum m_i)\infty$ 是既约除子,如果 D 是半既约的且 $\sum m_i\leqslant g$(H 的亏格). Jacobian 群中的每一个元素可用一个既约除子惟一表示.

一个半既约除子可用一对多项式来表示.

命题　一个半既约除子 $D=\sum m_i P_i-(\sum m_i)\infty$ 可用惟一的一对多项式 $a(x)$,$b(x)\in F[x]$ 表示[10]:对于每一个 $P_i=(x_i,y_i)\in H$,$a(x)=\prod (x-x_i)^{m_i}$;$b(x_i)=y_i$;$\deg(b)<\deg(a)=\deg D$;$a\mid b^2+bh-f$. 记为 $D=[a(x),b(x)]$ 或 $D=[a,b]$,称为 Mumford 表示式. Jacobian 群运算就是利用 Mumford 表示式进行计算.

有关超椭圆曲线的详细内容可参见文[1,9].

2　优化的算法

在基于离散对数的公钥密码体制中,比如椭圆曲线密码体制,标量乘法 mP 是加密和签名中最核心的计算环节. 如果把 m 表示为二进制(non-adjacent form)形式,那么有三分之一的概率计算 $2Q\pm P$,由一次倍点(中间结果 $2Q$) 和一次加法得到. 文[11,12] 通过减少中间结果的计算量,直接计算 $2Q\pm P$ 来达到提高标量乘法效率的目的. 受到这种思想的启发,可推广到超椭圆曲线上. 注意到在 T. Lange[8] 提供的计算加法(addition)公式中,计算 $[u_1,v_1]+[u_2,v_2]$ 时,v_1 只用到一次,那么在计算 $2D+E=(E+D)+D$ 时,中间结果 $E+D=[u',v']$ 中的 v' 不计算,它用其他已计算的量表示,代入下一轮的计算过程,从而减少计算量,$2D+E$ 和 $3D$ 分别节省 $2S+3M$ 和 $4M$. 如果一个逆运算 I 按 $5M$ 换算,那么计算 $2D\pm E$ 和 $3D$ 大约提高 6% ~

8%. 注意到$2D - E = 2D + \tilde{E}$,这里只考虑加法运算.

设亏格为 2 的超椭圆曲线 $H: y^2 + h(x)y = f(x)$,其中 $h(x) = h_2x^2 + h_1x + h_0$,$f(x) = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$. 设 D 和 E 是次数都为 2 的既约除子,且没有公共的支撑集. 为完整起见,在以下的计算中,提供了文[8]的主要步骤. 与[8]一致,假设 $h_2, h_1, f_4 \in F_2$. 表中 I,M 和 S 分别表示基域元素的逆、乘法和平方运算,且只考虑 D 和 E 的支撑集中没有互反的元素,即在通常情况下的运算过程.

2.1 计算 $2D+E$

设 $E = [u_1, v_1]$,$D = [u_2, v_2]$,其中 $u_i = x^2 + u_{i1}x + u_{i0}$,$v_i = v_{i1}x + v_{i0}$.

由 $2D + E = (E + D) + D$,分两个过程来完成.

过程一:

表 1 计算 $E+D=[u', v']$

步骤	过程	计算量
1	计算 u_1 和 u_2 的结式 r:	
	$z_1 = u_{11} - u_{21}$, $z_2 = u_{20} - u_{10}$, $z_3 = u_{11}z_1 + z_2$, $r = z_2z_3 + z_1^2u_{10}$	$1S, 3M$
2	计算 $\mathrm{inv} = r/u_2 \bmod u_1$: $\mathrm{inv}_1 = z_1$, $\mathrm{inv}_2 = z_3$.	
3	计算 $s' = rs = (v_1 - v_2)\ \mathrm{inv} \bmod u_1$:	
	$w_0 = v_{10} - v_{20}$, $w_1 = v_{11} - v_{21}$, $w_2 = \mathrm{inv}_0 w_0$, $w_3 = \mathrm{inv}_1 w_1$,	
	$s_1' = (\mathrm{inv}_0 + \mathrm{inv}_1)(w_0 + w_1) - w_2 - w_3(1 + u_{11})$, $s_0' = w_2 - u_{10}w_3$;	
	If $s_1' = 0$ call Harley algorithm.	$5M$
4	计算 $s'' = x + s_0/s_1 = x + s_0'/s_1'$:	
	$w_1 = (r s_1')^{-1} (= 1/r^2 s_1)$, $w_2 = r w_1 (= 1/s_1')$, $w_3 = s_1'^2 w_1 (= s_1)$,	
	$w_4 = r w_2 (= 1/s_1)$, $w_5 = w_4^2$,	
	$s_0'' = s_0' w_2$; 并令 $\alpha = s''$, $\beta = w_3$(第二过程用).	$I, 2S, 5M$
5	计算 $l' = s'' u_2 = x^3 + l_2' x^2 + l_1' x + l_0'$:只计算 $l_1' = u_{21}s_0'' + u_{20}$	$1M$
6	计算 $u' = x^2 + u_1' x + u_0'$:	
	$u_0' = (s_0'' - u_{22})(s_0'' - z_1 + h_2 w_4) - u_{10} + l_1' + (h_1 + 2v_{21})w_4 + (2u_{21} + z_1 - f_4)w_5$;	
	$u_1' = 2s_0'' - z_1 + h_2 w_4 - w_5$;	$3M$
	计算量合计	$I + 3S + 17M$

于是得到 $E + D$ 的中间结果 u',虽然 v' 不计算,但 $v' = -h - (\alpha \cdot \beta \cdot u_2 + v_2) \bmod u'$.

过程二:

表 2 计算 $[u', \Delta] + D(\Delta$ 即为上式的 $v')$

步骤	过程	计算量
7,8	同表 1 步骤 1,2(注 此时 $u_2 \cdot \mathrm{inv} \equiv r \bmod u'$)	$1S, 3M$
9	计算 $s' = r\,s = (\Delta - v_2)\ \mathrm{inv} \bmod u'$:	
	$w_0 = h_2 u_1' - h_1 - 2v_{21}$, $w_1 = h_2 u_0' - h_0 - 2v_{20}$, $w_2 = \mathrm{inv}_1 w_0$, $w_3 = \mathrm{inv}_0 w_1$,	
	$w_4 = (\mathrm{inv}_0 + \mathrm{inv}_1)(w_0 + w_1) - w_2 - w_3$, $w_5 = r\beta$,	
	$s_1' = w_4 - w_2 u_1' - w_5$, $s_0' = w_3 - w_2 u_0' - w_5 s_0''$($s_0''$ 在表 1 Step 4)	
	If $s_1' = 0$ call Harley algorithm.	$7M$
10	同表 1 步骤 4	$I, 2S, 5M$
11	计算 $l' = s'' u_2 = x^3 + l_2' x^2 + l_1' x + l_0'$:	
	$l_2' = u_{21} + s_0''$, $l_1' = u_{21}s_0'' + u_{20}$, $l_0' = u_{20}s_0''$;	$2M$
12	同表 1 步骤 6	$3M$
13	计算 $v' \equiv -h - (w_3 l' + v_2) \bmod u' = v_1' x + v_0'$:	
	$w_1 = l_2' - u_1'$, $w_2 = u_1' w_1 - l_1' + u_0'$, $v_1' = w_2 w_3 - v_{21} - h_1 + h_2 u_1'$;	
	$w_2 = u_0' w_1 - l_0'$, $v_0' = w_2 w_3 - v_{20} - h_0 + h_2 u_0'$	$4M$
	计算量合计	$I + 3S + 24M$

步骤 12 与步骤 13 中的 u' 与 v' 即为 $2D+E$ 的最终结果$[u',v']$.

注 步骤 9 成立是因为

$s'=(\Delta-v_2)\ \mathrm{inv}\bmod u'=(-h-\alpha\beta u_2-2v_2)\ \mathrm{inv}\bmod u'=[(-h-2v_2)\ \mathrm{inv}-\alpha\beta r]\bmod u'$.

因此,计算 s' 只需计算上面的最后一个式子,其中$(-h-2v_2)\ \mathrm{inv}\bmod u'$ 的计算量为 $5M$,$-\alpha\beta r\bmod u'$ 的计算量为 $2M$,共需 $7M$ 那么计算 $2D+E$ 总共需要 $2I+6S+41M$,如果按一个倍点和一个加法进行计算,总共需要 $2I+8S+44M$ 计算量,从而节省 2 个平方运算和 3 个乘法运算. 如果 $1I=5M$,$1S=0.8M$,则计算 $2D+E$ 的速度将提高 7.6% 左右.

2.2　计算 $3D$

类似于上节的方法,分两个过程来计算 $3D$.

设 $D=[u,v]$:$u=x^2+u_1x+u_0$,$v=v_1x+v_0$.

过程一:

表 3　计算 $2D=[u',v']$

步骤	过　　程	计算量
1	计算 $\bar v=(h+2v)\bmod u=\bar v_1x+\bar v_0$: $\bar v_1=h_1+2v_1-h_2u_1$　,　$\bar v_0=h_0+2v_0-h_2u_0$	
2	计算结式 $r=\mathrm{res}(\bar v,u)$: $w_0=v_1^2$　,　$w_1=u_1^2$　,　$w_2=\bar v_1^2$　,　$w_3=u_1\bar v_1$ $r=u_0w_2+\bar v_0(\bar v_0-w_3)$	$2S,3M$
3	计算 $\mathrm{inv}'=\mathrm{inv}_1'+\mathrm{inv}_0'$:$\mathrm{inv}_1'=-\bar v_1$　,　$\mathrm{inv}_0'=\bar v_0-w_3$	
4	计算 $k'=(f-hv-v^2)/u\bmod u=k_1'x+k_0'$: $w_3=f_3+w_1$　,　$w_4=2u_0$　;　$k_1'=2(w_1-f_4u_1)+w_3-w_4-v_1h_2$　; $k_0'=u_1(2w_4-w_3+f_4u_1+v_1h_2)+f_2-w_0-2f_4u_0-v_1h_1-v_0h_2$	$1M$
5	计算 $s'=k'\cdot\mathrm{inv}'\bmod u$: $w_0=k_0'\mathrm{inv}_0'$　,　$w_1=k_1'\mathrm{inv}_1'$　; $s_1'=(\mathrm{inv}_1'+\mathrm{inv}_0')(k_1'+k_0')-w_0-w_1(1+u_1)$　;　$s_0'=w_0-u_0w_1$　; If $s_1'=0$ call Harley algorithm.	$5M$
6	计算 $s''=x+s_0''=x+s_0/s_1$: $w_1=(rs_1')^{-1}(=1/r^2s_1)$　,　$w_2=rw_1(=1/s_1')$　,　$w_3=s_1'^2w^1(=s_1)$　; $w_4=rw_2(=1/s_1)$　,　$w_5=w_4^2$　; $s_0''=s_0'w_2$　;　并令 $\alpha=s''$　,　$\beta=w_3$(第二过程用)	$I,2S,5M$
7	计算 $u'=x^2+u_1'x+u_0'$: $u_1'=2s_0''+h_2w_4=w_5$　;　$u_0'=s_0''^2-w_4(h_2(s_0''-u_1)+h_1+2v_1)+(2u_1-f_4)w_5$;　$S,2M$	
	计算量合计	$I+5S+16M$

得到 $2D$ 的中间结果 u',同样 v' 不计算,$v'=-h-(\alpha\cdot\beta\cdot u_2+v_2)\bmod u'$.

过程二:计算 $[u',\Delta]+D$(Δ 即为上式的 v')

该过程与表 2 相同,需要 $I+3S+24M$. 那么计算 $3D$ 总共需要 $2I+8S+40M$. 如果按一倍点和一加法进行计算,总共需要 $2I+8S+44M$ 计算量,从而节省 4 个乘法运算. 如果 $1I=5M$,计算 $3D$ 的速度将提高 6.6% 左右.

以上方法可与其他方法相结合,来提高标量乘法的计算效率. 同时,这种方法也可有效地提高基于超椭圆曲线的 Weil/Tate 对的计算效率.

注:T. Lange[8] 提供的计算公式中,总假设 h_1 为 0 或 1,其实没有必要. 从公式中发现,h_1 的取值没有影响加法的计算量,而倍点也只增加一个乘法运算. 又根据文[13,14],当基域特征为 2 或不等于 5 时,H 总可转化为更简单的等价形式,使得 $h_2=0$ 或 1 及 $f_4=0$.

3 结 束 语

提出了超椭圆曲线的 Jacobian 群运算的一个优化方案,它是超椭圆曲线密码体制的重点研究课题之一. 超椭圆曲线密码体制自提出以来,就成为研究的热点之一. 虽然它是椭圆曲线密码体制的自然推广,但决不是简单的推广,尚有许多具体问题[15]需要解决和完善.

参考文献:

[1] Koblitz N. Hyperelliptic Cryptolgraphy[J]. J of Crypto, 1989, 1(3):139-150.

[2] Wang Hui, Zhang Fangguo, Wang Yumin. The Software Implementation of the Elliptic Curve Cryptosystem Over Large Prime Fields[J]. Journal of Xidian University, 2002, 29(3):426-428.

[3] Cantor D G. Computing in the Jacobian of a Hyperelliptic Curve[J]. Math Comp, 1987, 48(1):95-101.

[4] Harley R. Addition Text, Doubling C[DB/OL]. http://cristal.inria.fr/~harley/hyper, 2000-11-12.

[5] Nagao K. Improving Group Law Algorithms for Jacobinas of Hyperelliptic Curves[A]. ANTS-IV, LNCS 1838[C]. Berlin: Springer-Verlag, 2000. 439-448.

[6] Matsuo K, Chao J, Tsujii S. Fast Genus Two Hyperelliptic Curve Cryptosystems[DB/OL]. Technical Report of IEICE, ISEC2001-31, http://lab.iisec.ac.jp/~matsuo_lab/pub/MCT01, 2001-10-15.

[7] Kuroki J, Gonda M, Matsuo K, et al. Fast Genus Three Hyperelliptic Curve Cryptosystems[DB/OL]. Proc of SCIS2002, Symposium on Cryptography and Information Security, IEICE, Japan. http://lab.iisec.ac.jp/~matsuo_lab/pub/KGM+02, 2002-08-17

[8] Lange T. Efficient Arithmetic on Genus 2 Hyperelliptic Curves Over Finite Fields Via Explicit Formulae[DB/OL]. Cryptology ePrint Archive, Report 2002/121, http://eprint.iacr.org, 2002-09-10.

[9] Menezes A J, Wu Y H, Zuccherato R Z. An Elementary Introduction to Hyperelliptic Curves[R]. [s.l]: Technical Report CORR 96-19, Depart of C&O, Univ of Waterloo, 1996.

[10] Mumford D. Tata Lectures on Theta Ⅱ[M]. Boston: Birkhauser-Verlag, 1984.

[11] Eisentrager K, Lauter K, Montgomery P L. Fast Elliptic Curve Arithmetic and Improved Weil Pairing Evaluation[A]. M Joye, Editor, Topic in Cryptology-CT-RSA 2003, LNCS 2612[C]. Berlin: Springer-Verlag, 2003. 343-354.

[12] Ciet M, Joye M, Lauter K. Trading Inversions for Multiplications in Elliptic Curve Cryptography[DB/OL]. Cryptology ePrint Archive, Report 2003/257, http://eprint.iacr.org, 2003-12-05.

[13] Encinas L H, Menezes A J, Masque J M. Isomorphism Classes of Genus-2 Hyperelliptic Curves Over Finite Fields[J]. Applicable Algebra in Engineering Communication and Computing, 2002, 13(1):57-65.

[14] Choie Y, Jeong E. Isomorphism Classes of Hyperelliptic Curves of Genus 2 Over F_2^n [DB/OL]. Cryptology ePrint Archieve, Report 2003/213, http://erpint.iacr.org, 2003-10-08.

[15] 张方国,王育民. 超椭圆曲线密码体制的研究与进展[J]. 电子学报,2002, 30(1):126-131.

(编辑:李维东)