利用多基链计算椭圆曲线标量乘的高效算法

郝艳华[1,2]，李 磊[2,3]，王育民[2]

(1. 漳州师范学院计算机科学与工程系 福建 漳州 363000; 2. 西安电子科技大学综合业务网国家重点实验室 西安 710071;
3. 郑州大学信息工程学院 郑州 450052)

【摘要】椭圆曲线标量乘是椭圆曲线密码体制中最耗时的运算，多基链作为双基链的一个推广，具有标量表示长度更短、非零比特数目更少的特点，非常适宜用于椭圆曲线标量乘的快速计算。该文给出了新的五倍点公式，同时以2、3和5作为基底，给出了一个利用多基链计算椭圆曲线标量乘的高效算法。由于多基数表示的高度冗余性，该算法能够抵抗某些边信道攻击，与常用的标准倍点加和非邻接形标量乘算法相比，该算法的运算量更少。

关 键 词 椭圆曲线； 多基链； 公钥密码体制； 标量乘
中图分类号 TN918.2 **文献标识码** A

Efficient Scalar Multiplication Algorithm Using Multibase Chains

HAO Yan-hua[1,2], LI Lei[2, 3], and WANG Yu-min[2]

(1. Department of Computer Science and Engineering, Zhangzhou Normal University Zhangzhou Fujian 363000,
2. National Key Lab, of Integrated Service Networks, Xidian University Xi'an 710071;
3. School of Information Engineering, Zhengzhou University Zhengzhou 450052)

Abstract In the elliptic curve cryptosystem, Scalar multiplication is the most important and computationally costliest operation, thus it becomes one of hot topics. As a generalization of double base chains, multibase chains are very suitable for efficient computation of scalar multiplications of elliptic curves because of shorter representation length and less Hamming weight. In this paper, the formulas for computing the 5-fold of an elliptic curve point P are given. Using 2, 3 and 5 as bases of the multibase chains, an efficient scalar multiplication algorithm of elliptic curve is proposed. This algorithm can offer some protections against some side-channel attacks for the huge redundancy of the multibase representation and cost less compared with stand double-and-add and nonadjacent form for scalar multiplications.

Key words elliptic curve; multibase chain; public key cryptosystem; scalar multiplication

自文献[1-2]分别提出椭圆曲线密码体制(ECC)以来，ECC一直得到众多密码学家及密码学爱好者的青睐。ECC具有传统的其他公钥密码体制无法比拟的优势：(1) ECC的密钥长度更短，160 bit的ECC相当于1 024 bit的RSA；(2) ECC的单比特安全性更高；(3) 由于椭圆曲线复杂的代数结构，它能够提供更多的安全参数。这些特点使得ECC更加适合在资源受限的系统中使用，如灵巧卡。

在ECC的实现中最耗时的运算就是椭圆曲线标量乘法，即kP的计算，其中k为至少160 bit的大整数，P为一个椭圆曲线有理点。为了提高ECC的实现速度，密码学家想出了许多办法来提高椭圆曲线标量乘法的实现效率，其中最直接的方法就是考虑k的表示，如k的二元表示、非邻接形(NAF)及窗口法等。双基数系统最早由文献[3]提出来，不过当时不是用在密码上，而是用在数字信号处理中，之后经过很多学者的努力，将双基链用在了椭圆曲线标量乘中，取得了显著的效果[4-5]。文献[6]将双基链的概念扩展到了多基链的概念，并将其用于计算椭圆曲线标量乘，取得了更好的效果。加快椭圆曲线标量乘法的计算速度的另一个方法就是改进椭圆曲线有理点群的群运算。考虑到椭圆曲线倍点运算比点加运算耗时的特点，文献[7]给出了计算$2P+Q$的有效算法，即先计算$P+Q$，再计算$(P+Q)+P$，而不是先计算$2P$，再计算$2P+Q$，其中P、Q都是椭圆曲线上的有理点，该方法能够减少一个域乘。随后，在此基础上，文

收稿日期：2007－05－22；修回日期：2007－12－02
基金项目：国家自然科学基金(6043027)；福建省青年科技人才创建新基金(2008F3110)；福建省自然科学基金(2006J0045)
作者简介：郝艳华(1976－)，女，博士生，主要从事椭圆曲线及超椭圆曲线密码体制方面的研究.

献[8]进一步加快了椭圆曲线标量乘的运算速度。

本文结合文献[6-8]的思想和结论，给出了素数域上利用多基链计算椭圆曲线标量乘的一个较为高效的方法。

1　数学背景知识介绍

1.1　大素数域上的椭圆曲线

设K为特征不为2、3的有限域，$\overline{K}$为K的代数闭包，当满足$4a^3+27b^2\neq 0$时，K上的Weierstrass方程E：$y^2=x^3+ax+b$ ($a,b\in K$)称为定义在K上的椭圆曲线。满足该曲线方程的所有解(x、y)称为该椭圆曲线上的有理点，所有椭圆曲线上的有理点再加上一个被称为无穷远点的点O构成一个可换加法群。点加运算和倍点运算的运算量是不同的，分别是1[i]+1[s]+2[m]和1[i]+2[s]+2[m]，其中[i]、[s]和[m]分别表示域K上的一次求逆、平方和乘法所需花费。通常选取1[i]= 30[m]，而1[s]=0.8[m]。

1.2　多基链

设$\{b_i\}$、$\{t_i\}$、$\{q_i\}$为3个单调递减序列，$s_i=\pm 1$，则整数$n=\sum_{i=1}^{m}s_i2^{b_i}3^{t_i}5^{q_i}$称为整数$n$的多基链，$m$称为多基链的长度，基集$B$={2, 3, 5}。当$B$={2, 3}时，相应的表示称为双基链。双基链是高冗余的，而且表示长度非常短。与双基链相比，多基链冗余度更高，表示长度也更短。如仅考虑s_i=1情况下，100共有402个双基链表示，而它的多基链表示就有8 425个。当B={2, 3, 5, 7}时，100有43 777种对应表示。b_1、t_1、q_1的大小影响标量乘中二倍点、三倍点和五倍点运算的运算次数，而m–1为标量乘中点加的次数。一个160 bit的大整数如果使用双基链来表示需要大约23项，而如果使用多基链则需要约15项就可以了[6]，因此与使用双基链计算标量乘相比，使用多基链能够大大提高椭圆曲线标量乘法的计算效率。文献[6]给出了求任意整数n的多基数表示的有效算法。

2　五倍点的有效计算

在利用双基链计算椭圆曲线标量乘的过程中，主要用到的运算有P+Q、2P、2P+Q、3P、3P+Q、4P和4P+Q。对于这些运算，文献[8]均给出了高效的算法，表1列出了运算结果。当利用多基链来计算椭圆曲线标量乘时，上面的运算明显是不够的，需要用有效的五倍点公式，文献[6]给出了这样的一个高效算法，它利用了使用m次可除多项式来计算五倍点的想法，该算法尽管非常高效，但是随着m的增大，m次可除多项式的计算会变得非常复杂而且没有规律可寻，结果非常不易验证。已知当基集合中元素个数增加的时候，整数n的基中元素表示的个数会迅速增加，而且表示的长度会迅速减小，因此继续研究推广多基链的概念是提高椭圆曲线标量乘法效率的一个有效途径，如研究B={2, 3, 5, 7, 11}的时候，就需要计算七倍点和十一倍点，它们分别需要计算9次可除多项式和13次可除多项式，而这些计算会变得非常复杂。因此本文考虑另外的方法计算五倍点，该方法简单易记，容易验证，而且非常容易平移到更高倍点的计算中。

表1　素域上各种运算的运算花费

运算	运算花费
P+Q	1[i]+1[s]+2[m]
2P	1[i]+2[s]+2[m]
2P+Q	1[i]+2[s]+9[m]
3P	1[i]+4[s]+7[m]
3P+Q	2[i]+4[s]+9[m]
4P	1[i]+9[s]+9[m]
4P+Q	2[i]+4[s]+11[m]

设$iP=(x_i,y_i)$，计算5P有两种方案：一种是计算2P+3P；另一种是计算2(2P)+P。考虑到倍点运算比点加运算更耗时，因此本文选用第一种方案。

首先计算2P：

$$\lambda_1=\frac{3x_1^2+a}{2y_1},x_2=\lambda_1^2-2x_1,y_2=\lambda_1(x_1-x_2)-y_1$$

再计算3P：

$$\lambda_2=\frac{y_2-y_1}{x_2-x_1},x_3=\lambda_2^2-x_1-x_2,y_3=\lambda_2(x_2-x_3)-y_2$$

最后计算5P：

$$\lambda_3=\frac{y_3-y_2}{x_3-x_2},x_5=\lambda_3^2-x_2-x_3,y_5=\lambda_3(x_2-x_5)-y_2$$

如果直接按照上面的公式计算5P，需要3次求逆，并且需要进行多个容易计算，这就大大增加了冗余计算，因此本文加入一些小技巧来简化运算。

计算2P，利用一次求逆同时求解$(x_2-x_1)^{-1}$和$(x_2-x_3)^{-1}$，并且省去中间变量x_3、y_3的计算。文献[7-8]直接应用了该方法，具体分析如下：

$$\lambda_1=\frac{3x_1^2+a}{2y_1},x_2=\lambda_1^2-2x_1,y_2=\lambda_1(x_1-x_2)-y_1$$

$$\lambda_3=\frac{\lambda_2(x_2-x_3)-y_2-y_2}{x_3-x_2}=\frac{2y_2}{x_2-x_3}-\lambda_2$$

令 $d=(x_1+2x_2)(x_2-x_1)^2-(y_2-y_1)^2$，容易验证 $d=(x_2-x_1)^2(x_2-x_3)$。

令 $D=d(x_2-x_1)$，$I=D^{-1}$，则有：

$$\lambda_2=(y_2-y_1)dI \qquad \lambda_3=2y_2(x_2-x_1)^3I-\lambda_2$$

$$x_5=\lambda_3^2-x_2-\lambda_2^2+x_1+x_2=(\lambda_3+\lambda_2)(\lambda_3-\lambda_2)+x_1$$

$$y_5=\lambda_3(x_2-x_5)-y_2$$

该方案的运算量如表2所示。在Input: $P=(x_1,y_1)\neq O$，Output: $5P=(x_5,y_5)$ 下，总运算量为2[i]+4[s]+11[m]。

表2 素域上求五倍点的第一种方案

顺序	运算	运算量
1	$\lambda_1=\frac{3x_1^2+a}{2y_1}$	1[i]+1[s]+1[m]
2	$x_2=\lambda_1^2-2x_1$	1[s]
3	$y_2=\lambda_1(x_1-x_2)-y_1$	1[m]
4	$d=(x_1+2x_2)(x_2-x_1)^2-(y_2-y_1)^2$	2[s]+1[m]
5	$D=d(x_2-x_1)$	1[m]
6	$I=D^{-1}$	1[i]
7	$\lambda_2=(y_2-y_1)dI$	2[m]
8	$\lambda_3=2y_2(x_2-x_1)^3I-\lambda_2$	3[m]
9	$x_5=(\lambda_3+\lambda_2)(\lambda_3-\lambda_2)+x_1$	1[m]
10	$y_5=\lambda_3(x_2-x_5)-y_2$	1[m]

第二种方案是进行重新组合，利用一次求逆同时求解 $(2y_1)^{-1}$ 和 $(x_1-x_2)^{-1}$，省去中间变量 y_2、y_3 的计算，具体分析如下：

$$\lambda_1=\frac{3x_1^2+a}{2y_1}$$

$$\lambda_2=\frac{\lambda_1(x_1-x_2)-2y_1}{(x_2-x_1)}=\frac{2y_1}{(x_1-x_2)}-\lambda_1$$

令 $d=12x_1y_1^2-(3x_1^2+a)^2$，容易验证 $d=4y_1^2\times(x_1-x_2)$。

令 $D=2y_1d$，$I=D^{-1}$，则有：

$$\lambda_1=(3x_1^2+a)dI \qquad \lambda_2=16y_1^4I-\lambda_1$$

$$x_2=\lambda_1^2-2x_1 \qquad x_3=\lambda_2^2-x_1-x_2$$

$$\lambda_3=\frac{2y_2}{x_2-x_3}-\lambda_2$$

$$x_5=\lambda_3^2-x_2-x_3,\ y_5=\lambda_3(x_2-x_5)-y_2$$

该算法的运算量如表3所示。在Input: $P=(x_1,y_1)\neq O$，Output: $5P=(x_5,y_5)$ 下，总运算量为2[i]+7[s]+7[m]。

由于一般选取1[s]=0.8[m]，因此第二种方案优于第一种方案，在下面计算标量乘的时候用到的五倍点公式均使用第二种方案。五倍点的计算公式也可用于使用B={2,5}的双基链来计算椭圆曲线标量乘的计算中。

表3 素域上求五倍点的第二种方案

顺序	运算	运算量
1	$d=12x_1y_1^2-(3x_1^2+a)^2$	3[s]+1[m]
2	$D=2y_1d$	1[m]
3	$I=D^{-1}$	1[i]
4	$\lambda_1=(3x_1^2+a)dI$	2[m]
5	$\lambda_2=16y_1^4I-\lambda_1$	1[s]+1[m]
6	$x_2=\lambda_1^2-2x_1$	1[s]
7	$x_3=\lambda_2^2-x_1-x_2$	1[s]
8	$\lambda_3=\frac{2y_2}{x_2-x_3}-\lambda_2$	1[i]+1[m]
9	$x_5=\lambda_3^2-x_2-x_3$	1[s]
10	$y_5=\lambda_3(x_2-x_5)-y_2$	1[m]

3 用多基链计算椭圆曲线标量乘算法

前面给出了五倍点的计算公式，结合文献[8]给出的结果，本文给出利用多基链计算椭圆曲线标量乘的算法：

Input：整数 $k=\sum_{i=1}^{m}s_i2^{b_i}3^{t_i}5^{p_i}$，其中 $s_i\in\{-1,1\}$，$b_1\geqslant b_2\geqslant\cdots\geqslant b_m\geqslant 0$，$t_1\geqslant t_2\geqslant\cdots\geqslant t_m\geqslant 0$，$p_1\geqslant p_2\geqslant\cdots\geqslant p_m\geqslant 0$，点 $P\in E(K)$。

Output：点 $kP\in E(K)$。

(1) $Q\leftarrow s_1P$
(2) for $i=1,\cdots,m-1$ do
(3) $u\leftarrow b_i-b_{i+1}$
(4) $v\leftarrow t_i-t_{i+1}$
(5) $w=p_i-p_{i+1}$
(6) If $u=0$ then
(7) If $w=0$ then
(8) $Q\leftarrow 3(3^{v-1}Q)+s_{i+1}P$
(9) else
(10) If $v=0$ then
(11) $Q\leftarrow 5^wQ+s_{i+1}P$
(12) else
(13) $Q\leftarrow 5^wQ$
(14) $Q\leftarrow 3(3^{v-1}Q)+s_{i+1}P$
(15) else
(16) If $w=0$ then
(17) $Q\leftarrow 3^vQ$
(18) $Q\leftarrow 4^{\lfloor (u-1)/2\rfloor}Q$
(19) If $u\equiv 0\ (\mathrm{mod}\ 2)$ then

(20) $Q=4Q+s_{i+1}P$
(21) else
(22) $Q=2Q+s_{i+1}P$
(23) else
(24) $Q=5^w$
(25) $Q\leftarrow 3^vQ$
(26) $Q\leftarrow 4^{\left\lfloor (u-1)/2\right\rfloor}Q$
(27) If $u\equiv 0 \pmod 2$ then
(28) $Q=4Q+s_{i+1}P$
(29) else
(30) $Q=2Q+s_{i+1}P$
(31) Return Q

本文的标量乘算法与文献[6]给出的标量乘算法主要的不同点在于，本文注意到直接计算4P只需要一次求逆，而计算2(2P)则需要两次求逆，在求逆比乘法慢得多的大素数域中，当然直接利用4倍点公式要划算得多。将运算P+Q、2P、2P+Q、3P、3P+Q、4P、4P+Q、5P分别记为A、D、DA、T、TA、Q、QA、F，下面给出该算法的复杂度分析：

上述算法共迭代m–1次，第i轮迭代所需的运算量记为W_i，有：

$$W_i=\delta_{u_i,0}\left[\delta_{w_i,0}((v_i-1)T+TA)+(1-\delta_{w_i,0})\times\right.$$
$$\left.[\delta_{v_i,0}(w_iF+A)+(1-\delta_{v_i,0})(w_iF+(v_i-1)T+TA)]\right]+$$
$$(1-\delta_{u_i,0})\left[\delta_{w_i,0}(v_iT+\left\lfloor\frac{u_i-1}{2}\right\rfloor Q+\delta_{|u_i|_2,0}QA+\delta_{|u_i|_2,1}DA)+\right.$$
$$\left.(1-\delta_{w_i,0})(w_iF+v_iT+\left\lfloor\frac{u_i-1}{2}\right\rfloor Q+\delta_{|u_i|_2,0}QA+\delta_{|u_i|_2,1}DA\right]$$

式中　$\delta_{i,j}=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases}$；$|u|_2$表示模2剩余。因此上述算法的总运算量$W=\sum_{i=1}^{m-1}W_i$。

文献[6]选取了大量的随机整数，给出了整数多基链表示的一些平均数据，结合这些数据，表4给出了本文算法的运算量、标准倍点加及NAF标量乘的运算量的大小比较。

表4　本文算法与其他标量乘算法的运算量比较

算法	运算量
标准倍点加[9]	2 511[m]
NAF[9]	2 214[m]
本文算法	1 652[m]

由表4可以看出本文算法的运算量小于另外两种标量乘算法的运算量。

4　结　论

本文给出了五倍点的运算公式和利用多基链计算椭圆曲线标量乘的高效算法，需要强调的是，利用多基链计算椭圆曲线标量乘不仅能够提高标量乘的运算效率，使得基于椭圆曲线的密码体制实现更加便捷和高效，而且由于双基链表示的高度冗余性，多次计算同一个标量乘，计算过程可以完全不同，因此使用双基链计算椭圆曲线标量乘可以抵抗某些边信道攻击[10]。

参　考　文　献

[1] KOBLITZ N. Elliptic curve cryptosystems[J]. Mathema -tics of Computation, 1987, 48(177): 203-209.
[2] MILLER V S. Uses of elliptic curves in cryptography[C]// CRYPTO’85: Proceedings of Advances in Cryptology. Springer Berlin: Heidelberg Press, 1986, 218: 417-428.
[3] DIMITROV V S, JULLIEN G A. Loading the bases: a new number representation with applications[J]. IEEE Circuits and Systems Magazine, 2003, 3(2): 6-23.
[4] DIMITROV V S, IMBERT L, MISHRA P K. Fast elliptic curve point multiplication using double-base chains [DB/OL]. [2007-04-10]. http://eprint.iacr.org/2005/069.
[5] AVANZI R, DIMITROV V S, DOCHE C et al. Extending scalar multiplication using double bases[C]//ASIA CRYPT’06: Proceedings of Advances in Cryptolo gy-ASIACRYPT 2006. Springer Berlin: Heidelberg Press, 2006, 4284: 130-144.
[6] MISHRA P K, DIMITROV V S. Efficient quintuple formulas for elliptic curves and efficient scalar multiplication using multibase number representation [DB/OL]. [2007-04-10]. http://eprint.iacr.org/2007/040.
[7] EISENTRAGER K, LAUTER K, MONTGOMERY P L. Fast elliptic curve arithmetic and improved Weil pairing evaluation[C]//CT-RSA 2003: Proceedings of Topics in Cryptology. Springer Berlin: Heidelberg Press, 2003, 2612: 343-354.
[8] CIET M, JOYE M, LAUTER K et al. Trading inversions for multiplications in elliptic curve cryptography[J]. Designs, codes and cryptography, 2006, 39: 189-206.
[9] DIMITROV V S, IMBERT L, MISHRA P K. Efficient and secure curve point multiplication using double-base chains[C]//Asiacrypt 2005: Proceedings of Advances in Cryptology-ASIACRYPT 2005. Springer Berlin: Heidelberg Press, 2005, LNCS 3788: 59-78.
[10] KOCHER C, JAFFE J, JUN B. Differential power analysis[C]//Crypto’99: Proceedings of Advances in Cryptology. Springer Berlin: Heidelberg Press, 1999, 1666: 388-397.

编　辑　张　俊