软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(4):782−788 [doi: 10.3724/SP.J.1001.2011.03730]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

椭圆曲线密码中一种多标量乘算法*

陈厚友+, 马传贵

(信息工程大学 信息工程学院,河南 郑州 450002)

A Multiple Scalar Multiplications Algorithm in the Elliptic Curve Cryptosystem

CHEN Hou-You+, MA Chuan-Gui

(Institute of Information Engineering, Information Engineering University, Zhengzhou 450002, China)

+ Corresponding author: E-mail: chenhouyou1979@gmail.com

Chen HY, Ma CG. A multiple scalar multiplications algorithm in the elliptic curve cryptosystem. *Journal of Software*, 2011,22(4):782−788. http://www.jos.org.cn/1000-9825/3730.htm

Abstract: The main operations of elliptic curve cryptosystems (ECCs) are scalar multiplications and multi-scalar multiplications, which heavily determined the overall implementation of the efficiency of ECC. This algorithm extends the fixed-base window method by using the signed integer factorial expansions of scalar. The main characteristic of this method is that only a point addition computation is required, and it greatly improves the computational performance of a multi-scalar. Furthermore, the correctness proof and complexity analysis of the new algorithm are presented. At last, experimental results show that the computational efficiency increases about 47.8% to 56.5% when compared with other existing methods in the case m=2.

Key words: point multiplication; multi-scalar multiplication; factorial expansion; T-multi-scalar multiplication; fixed-base window method

摘 要: 标量乘和多标量乘是实现椭圆曲线密码体制的核心运算,其运算速度从整体上决定了椭圆曲线密码体制的实现效率.提出了一种多标量乘算法,该算法的基本思想是,将标量用带符号的整数阶乘展开式表示,并结合固定基窗口标量乘算法,使得实现多标量乘算法只需做点加运算即可.这不仅突破了传统求多标量乘算法的模式,而且提高了多标量乘的计算速度.同时,还对算法正确性和复杂度进行了分析.由实验结果可知,在 m=2 的情况下,该算法在计算效率上比已有的多标量乘算法提高了约 47.8%~56.5%.

关键词: 点乘;多标量乘;阶乘展开式;T-形多标量乘;固定基窗口算法

中图法分类号: TP309 **文献标识码**: A

自 1985 年 Miller 和 Koblitz 各自独立地提出椭圆曲线公钥密码以来,椭圆曲线公钥密码体制(elliptic curve cryptosystem,简称 ECC)就以其独特的优势,如计算速度快、存储空间小、带宽要求低、计算参数少等,引起了密码学工作者的高度关注.发展至今,ECC 不仅被广泛地应用到信息安全领域,而且还形成了一些国际组织认可并作为公钥密码加密标准的国际标准(IEEE PI1363,ANSI X9,ISO/IEC 和 NIST 等).但如何高效而快速地实现椭

* 基金项目: 国家自然科学基金(90704003); 国家高技术研究发展计划(863)(863-317-01-04-99, 2007AA01Z431); 河南省重大科技攻关项目(092101210502)

收稿时间: 2008-11-20; 定稿时间: 2009-08-28

© 中国科学院软件研究所 http://www.jos.org.cn

圆曲线的加、解密问题,仍然是人们关注和研究的重点.

标量乘及多标量乘算法是实现椭圆曲线密码系统的核心运算,其运算速度从整体上决定了 ECC 的实现效率.标量乘算法即计算 kP,其中,k 是随机选取的一个大正整数,P 是椭圆曲线上的一个基点.快速标量乘计算的关键问题是如何高效地分解大正整数 k,在这方面,人们已提出了诸多快速算法,如二进制标量乘、带符号的二进制标量乘、窗口标量乘、固定基 Comb 标量乘[1]、Frobenius 标量乘[2]、Montgomery 标量乘[3]以及点折半标量乘[2]等等.而 $k_1P_1+\ldots+k_{m-1}P_{m-1}+k_mP_m$ 的计算被称为多标量乘,其中,P_i 为椭圆曲线上的点,且正整数 $k_i(i=1,\ldots,m)$ 的长度要求近似相等.多标量乘运算是构成许多密码协议最核心的部分,如 ECDH[4],EC-NR[4]和多重数字签名[5]等.在椭圆曲线数字签名体制的验证过程中,常常需要计算 $kP+JQ$.此外,标量乘也可转化成多标量乘来计算,如 Comb 算法、Lim 和 Lee 算法[6]以及随机化标量乘算法[7]等.因此,椭圆曲线密码体制的快速实现不仅要求标量乘 kP 的快速实现,还要求多标量乘的快速实现.目前,多标量乘算法主要有直接算法、Shamir 窍门方法[8]、联合稀疏形方法(JSF)[8]和 T-形多标量乘[9]等,其主要思想都是把 k_i 展开成 0,1 比特串,并且使得零列尽可能地多.但上述算法仅仅在点加的计算效率有所提高,而在倍点的计算上却没有任何改进.目前,国内外较为流行的整数二进制展开式方法是 NAF 方法[10],但由于其汉明密度是 1/3,即对于多标量乘的列而言,为 0 的概率是 $(2/3)^m$,由此可知,NAF 方法的点加效率会随着 m 的增大而有所下降.T-形多标量乘[9]是目前较新的一种多标量乘算法,该算法的主要特点是,利用 T-形式带符号二进制串的生成方法,使得零列出现的概率为 $P(S_0)$[9].Shamir 窍门方法[1]本质上也是一种固定窗口方法,其计算效率比 T-形多标量乘和联合稀疏形方法(JSF)都要高(见后文的表 2).但是,由于 Shamir 窍门方法的预置点数呈指数级增长,因此,在存储空间较小的情况下,该方法具有很大的局限性.特别地,当 $m,\omega(\omega$是窗口宽度)较小时,其计算效率并不高(详见后文的表 3).

本文在研究椭圆曲线标量乘和多标量乘算法的基础上,根据文献[11]的定理 2,提出了一种新的快速算法——带符号的整数阶乘展开式多标量乘算法.该算法的基本思想是,把私钥用带符号的整数阶乘展开式表示,即用类基(1!,2!,…,S!)来表示私钥.其主要优势表现在:(1) 标量乘 kP 的计算就简化成为 k_iP_i 的计算,其中,k_i 为私钥的带符号整数阶乘展开式中的系数,而 k_i 远远小于私钥.例如,200 比特的一个大数,其系数最大仅为 24;(2) 传统上通常采用多标量乘的方法来计算标量乘,而本文结合固定基窗口标量乘方法[3]的思想,采用标量乘的方法来实现多标量乘的运算.这样一来,实现多标量乘便不再需要计算倍点.由于椭圆曲线上的点乘运算 M 和倍点运算 D 有 $D\approx0.8M$ 的近似关系[9],因此,本文所提出的算法不仅能够简化计算,同时还极大地提高了计算效率.最后,本文通过比较发现,当 $m=2$ 时,与现有的多标量乘算法相比,所提算法在计算效率上提高了约 47.8%~56.5%.

本文第 1 节主要介绍现有的椭圆曲线上点乘的快速算法.第 2 节介绍带符号的整数阶乘展开式多标量乘算法.第 3 节为结束语.

1 椭圆曲线上点乘的快速算法

由于 kP 是普通 Abel 群的模指数运算的特殊情况,因此,指数运算的各种方法和技巧同样适用于椭圆曲线点的标量乘法.为了后面叙述方便,不妨假定运算量的运行时间符号分别为:M 表示点乘,D 表示倍点.在这里,我们仅介绍本文使用到的算法.

1.1 固定基窗口算法

固定基窗口算法[1]的主要特点是,先计算预置表再进行主程序的运算,以使得主程序不需要运算倍点.在忽略计算预置表运行时间的情况下,该算法的期望运行时间约为 $(2^{\omega}+d-3)M$.其中,ω表示窗口宽度,d 表示以 2^{ω}为基的 k 的长度.

1.2 带符号的整数阶乘展开式的算法

引理 1[11]. 对于任一正整数 $k(0\leqslant k\leqslant S!)$,可表示为

$$k=k_S S!+\ldots+k_2 2!+k_1,\ 0\leqslant|k_i|\leqslant\left\lfloor\frac{i+1}{2}\right\rfloor,\ i=1,2,\ldots,S \tag{1}$$

© 中国科学院软件研究所 http://www.jos.org.cn

称公式(1)为带符号的整数阶乘展开式.其中,(1!,2!,…,S!)被称为整数 k 的类基.

2 带符号的整数阶乘展开式多标量乘算法

注意到,第 1.1 节固定基窗口算法的计算速度主要由 d 和ω所决定,但是,由于 $d=\lceil t/\omega \rceil$,即 $d-1$ 将随着ω的增加而减小,因此,若仅依靠通过改变窗口宽度来提高固定基窗口算法的计算效率是不可行的.整数阶乘展开式的主要思想是,把整数 k 用类基(1!,2!,…,S!)的线性组合来表示,展开式中相应的系数为 $(k'_S,...,k'_1)$.而本文在以下第 2.1 节所提出的带符号的整数阶乘展开式多标量乘算法则是上述两种算法的有机结合.整数 k 既可以采用以 2^ω 为基的表达式 $(k_{d-1},...,k_1,k_0)_{2^\omega}$ 线性表示,也可以采用类基(1!,2!,…,S!)线性表示,但当k较大时,用后者表示比用前者表示在多标量乘算法上具有如下优势:首先,后者需要的存储点数更少(比较见表 2);其次,由于 $k'_i \leqslant$,而 $k_i \leqslant \lfloor (S+1)/2 \rfloor 2^\omega-1$,显然,以 2^ω为基的表达式中的系数 k_i 以指数级增长,而采用类基(1!,2!,…,S!)的表达式中的系数 k'_i 仅以整数 1 递增(比较见表 1).因此,整数阶乘展开式算法使得标量以较小的系数展开,从而减少了算法中的点加次数;而固定基窗口算法则保证了多标量乘的实现仅需要点加运算.

Table 1 Precomputer number of points

表 1 预置点的个数

k value (bit)	160~164	165~169	170~175	176~180	181~186	187~191	192~197	198~202
Precomputer number of points	40	41	42	43	44	45	46	47

Table 2.1 Shamir's method

表 2.1 Shamir 窍门算法

m	2	3	4	5
$\omega=3t=163$	63	511	4 095	32 767
$\omega=4t=192$	255	4 095	65 535	1 048 575

Table 2.2 New method

表 2.2 新算法

m	2	3	4	5
$t=163$	80	120	160	200
$t=192$	92	138	184	230

新算法的具体实现方法如下:第 1 步,求 k_i 的带符号的阶乘展开式;第 2 步,计算预置表,并存储 2!P_i,3!P_i,…,S!P_i;第 3 步,按照固定基窗口标量乘算法计算 $\sum_{i=1}^{m} k_i P$.

2.1 带符号的整数阶乘展开式多标量乘算法

算法 1. 预置表程序.

输入:P;

输出:预置表.

1. $Q_i \leftarrow P$
2. i from 2 to $n-1$

   $Q_i \leftarrow iQ_1$ 把每个 Q_i 存储到预置表中
3. 返回预置表 Q_i

算法 2. 新的多标量乘算法.

输入:$k_i,P_i,P,Q \in E(F_q),i=1,2,\ldots,m$;

输出:$k_1P_1+\ldots+k_{m-1}P_{m-1}+k_mP_m$.

(1) 计算 $k_i,i=1,2,\ldots,m$ 的带符号的阶乘展开式

(2) 计算 $P_i,i=1,2,\ldots,m$ 的阶乘展开式的预置表

(3) 记 $k_i=(k_{i_1},k_{i_2},...,k_{i_s})$, $i=1,2,\ldots,m$

(4) $R \leftarrow \infty, A \leftarrow \infty$

(5) for j from $\left\lfloor \frac{s}{2} \right\rfloor$ to 1 do

© 中国科学院软件研究所 http://www.jos.org.cn

① 对每个满足 $k_{i_j}=j$ 的 i,执行 $R\leftarrow R+P_{i_j}$

② 对每个满足 $k_{i_j}=-j$ 的 i,执行 $R\leftarrow R-P_{i_j}$

③ $A\leftarrow R+A$

(6) 返回 A

2.2 新算法的正确性分析

2.2.1 算法的终止

算法 2 的步骤(1),由引理 1 可知,对于每一个 k_i 都可以用带符号的阶乘展开式表示.算法 2 的步骤(2)需要计算预置表,即计算 $j!P_i(i=1,2,\ldots,m,j=1,\ldots,S)$.由于多标量乘中 P_i 的个数 m 是有限的,并由引理 1 可知,S 由 k 所决定,也即由步骤(2)中的 k_i 所决定,由于 k_i 有界,所以 S 必有界,从而 mS 亦有限.因此,预存点的个数是有限的.同理,可以考虑算法 2 的步骤(5).综上所述,算法 2 一定终止.

2.2.2 算法的正确性

算法的正确性证明:即需要证明 $A=\sum_{i=1}^{m}k_iP_i$. 由文献[1]给出的一个推论可知,

$$kP=\sum_{i=0}^{d-1}k_i2^{\omega i}P=\sum_{j=0}^{2^\omega-1}\left(j\sum_{i:k_i=j}2^{\omega i}P\right)=\sum_{j=0}^{2^\omega-1}(jQ_j)=Q_{2^\omega-1}+(Q_{2^\omega-1}+Q_{2^\omega-2})+\ldots+(Q_{2^\omega-1}+Q_{2^\omega-2}+\ldots+Q_1).$$

于是有

$$\sum_{i=1}^{m}k_iP_i=\sum_{j=1}^{t}\left(j\sum_{i:k_{i_j}=j}d_iP_{i_j}\right)=\sum_{j=1}^{t}jQ_j=Q_t+(Q_t+Q_{t-1})+\ldots+(Q_t+Q_{t-1}+\ldots+Q_1),$$

其中,$t=\left\lfloor\frac{s}{2}\right\rfloor$,$Q_j=\sum_{i:k_{i_j}=j}d_iP_{i_j}$. 当 $k_{i_j}>0$ 时,$d_i=1$;当 $k_{i_j}<0$ 时,$d_i=-1$.从而步骤(5)得证,即而可得出步骤(6):$A=\sum_{i=1}^{m}k_iP_i$.证毕. □

2.3 新算法的复杂度

标量乘运算是椭圆曲线密码的核心运算,因此,椭圆曲线密码系统的实现效率在很大程度上取决于标量乘运算的简单、快速.新算法的复杂度可行性分析分为如下 4 个部分:

2.3.1 预置表的复杂度分析

考虑算法 2 的步骤(2),因为 $2!P,3!P,\ldots,S!P$ 之间不相互独立,即 $i!P=i\cdot(i-1)!P,i\in[2,S]$,记 $P_{i-1}=(i-1)!P$,易得 $i!P=iP_{i-1}$.由 NAF[10]标量乘算法可知,计算 $i!P$ 比计算$(i-1)!P$ 需要多进行 $t-1$ 次倍点和 $t/3$ 次点加运算,其中 $t=\lceil\log_2 i\rceil$.又由于 i 远远小于大数 k,因此,t 通常为很小的数.例如:一个 200 比特位的 $k=2^{200}$,而它的阶乘展开式最大的 i 仅仅是 47,$NAF(47)=(10\bar{1}0001)$,即预置点的每次循环最多增加 6 次倍点运算和 1 次点加运算.因此,$i!P$ 的实现是容易的;其次,在多标量乘的实际运算过程中,通常 m 的取值应小于 8.同时,又结合整数阶乘展开式将使得存储点个数得以减少的特点(相应的结果见表 1 和表 2),于是可以得到,算法 2 中步骤(1)的预置点的复杂度将会极大地降低;再次,由于相互间存在线性关系,因此在程序的实现方面,只需进行相同的循环迭代.由表 1 易见,预置点的增加速度小于比特的增加速度,而这一特点对于预置表增加时间的计算是极为有利的.

2.3.2 展开式的复杂度分析

在步骤(1)中,由于阶乘展开式的收敛速度较快,于是对于相对复杂的大数运算而言,小数运算的运行时间可以忽略不计.此外,普通的加减运算相对于椭圆曲线上标量乘运算来说,其运行时间也可以忽略不计.因此,求标量的阶乘展开式的计算时间相对于步骤(5)的计算时间而言可以忽略不计.

2.3.3 步骤(5)的复杂度分析

这是新算法计算效率得以提高的核心.由于大数 k_i 用带符号的整数阶乘展开式展开后,其展开式中的系数 k_{i_j} 远远小于 k_i,这样就使得步骤(5)中的步骤③最多需要进行 $\left\lfloor\frac{s}{2}\right\rfloor-1$ 次点加运算.而步骤(5)中的步骤①、步骤②

© 中国科学院软件研究所 http://www.jos.org.cn

所需要的加法次数则依赖于展开式中系数 k_{i_j} 的个数,显然,系数 k_{i_j} 的个数小于预置点的个数.再根据以上对预置点的复杂度分析可知,步骤(5)中的步骤①、步骤②是可行的,且最多需要进行 $mS-1$ 次点加运算.

2.3.4 算法 2 的复杂度

由于带符号整数阶乘展开式的运行时间相对椭圆曲线的点乘运算的运行时间来说可以忽略不计,因此,算法 2 的用时主要涉及到步骤(1)和步骤(5),其中,步骤(1)需要预置 mS 个点,步骤(5)用时约为 $\left(\left\lfloor\frac{s}{2}\right\rfloor+mS-2\right)M$.

2.4 算法2与已有算法的比较

首先给出 Shamir 窮门方法和新算法预置点的比较.

由表 2.1 和表 2.2 易知,Shamir 窮门方法由于其存在预置点个数呈指数级增长的缺陷(预置点的个数为 $2^{m\omega}-1$),无法适应多标量乘算法的运算,并且通过表 2.1 可以发现,当 $m\geqslant 4$ 时,Shamir 窮门方法在实际应用中是不适合的.而与 Shamir 窮门方法相比,新算法在预置点个数上具有极大的优越性(t 是 NIST 推荐的安全椭圆曲线参数的比特长度).

本文中其他多标量乘算法的复杂度分别为:Shamir 窮门方法是 $(9+15t/32)M+(t+2)D$;联合稀疏形方法(JSF)[8]约为 $\left(\frac{t}{2}+2\right)M+tD$;T-多标量乘方法[9]为 $(1-p(S_0))tM+(1+t)D$,其中,$t=\lceil\log_2 k\rceil$(见表 3).

Table 3 Algorithm complexity analysis

表 3 算法的复杂度分析

Computational step \ Method	Shamir's method	JSF method	T-Multi-Scalar multiplication method	Signed integer factorial expansion of multi-scalar multiplication method
Point addition	$\frac{2^{m\omega}-1}{2^{m\omega}}d-1$	$(1-(2/3)^m)t$	$(1-p(S_0))t$	$\left(\left\lfloor\frac{s}{2}\right\rfloor+mS-2\right)$
Point doubling	$t+1$	$t+1$	$t+1$	0

从表 4 中我们可以发现,当参数 m 取不同值时,在忽略计算预置点所需时间的情况下,新算法与本文中其他方法在点乘效率上近似相等.但由于新算法省略了倍点运算,且根据椭圆曲线上的点乘运算 M 和倍点运算 D 有 $D\approx 0.8M$ 的近似关系,所以,本文所提出的新算法在多标量乘的复杂度上得到大幅度的提高(相应的结果见表 4,$t=200$,$\omega=3$,$d=67$,$S=48$,$p(S_0)$同上).

Table 4 Comparison of multi-scalar multiplication algorithm's complexity

表 4 多标量乘算法复杂度的比较

m \ Method	2		3		4	
Shamir's method	66	201	67	201	67	201
JSF method	111.1	201	140.7	201	160.5	201
T-Multi-Scalar multiplication method	100	201	118	201	128.4	201
Signed integer factorial expansion of multi-scalar multiplication method	118	0	168	0	214	0
Percentage (%)	48~56		26~44		6~34	

考虑 NIST 推荐的安全椭圆曲线[12],其参数设置如下(利用 Miracl 大数包,运行环境为:Pentium(R)4 CPU2.00 GHz 256MB),相应的计算结果见表 5.

$P=2^{192}-2^{64}-1$ $a=-3$

b=0x64210519*E*59C80*E*70*FA*7*E*9*AB*72243049*FEB*8*DEECC*146*B*9*B*1

n=0x*FFFFFFFFFFFFFFFFFFFFFFFF*99*DEF*836146*BC*9*B*1*B*4*D*22831

x=0x188*DA*80*EB*03090*F*67*CBF*20*EB*43*A*18800*F*4*FF*0*AFD*82*FF*1012

y=0x07192*B*95*FFC*8*DA*78631011*ED*6*B*24*CDD*573*F*977*A*11*E*794811

k_1=0x1*F*40*CBFC*5570506*E*70*B*594*E*72039*DE*5*D*07*C*1*E*9831*ECF*0*C*

k_2=0x1*F*4*BC*80*FBB*7*EB*5*F*12*F*312642*F*5407*EAEAAD*995092*FB*70*F*

© 中国科学院软件研究所 http://www.jos.org.cn

k_3=0x1F4BC80FBB7E16CCAD716B3C1ED000B6D66A2D0CAFF267

k_4=0x1F4BC81471637C48CBAD04FA243BAB7DD60B7E78691568

Table 5 Experimental data (ms)

表 5 实验数据 (毫秒)

Method \ m	2	3	4
Shamir's method	29.71	29.53	30.01
JSF method	36.10	40.05	37.84
T-Multi-Scalar multiplication method	31.34	33.05	35.87
Signed integer factorial expansion of multi-scalar multiplication method	11.16	17.68	18.61

通过表 5 我们可以发现,在 NIST 推荐的安全椭圆曲线参数条件下,本文所提出的带符号整数阶乘展开式多标量乘算法在运算时间上明显优于现有的多标量乘算法.

3 结束语

本文给出了一种新的多标量乘算法——带符号的整数阶乘展开式多标量乘算法.同时,本文还详细给出了新算法的正确性证明、可行性以及复杂度分析.该算法结合了固定基窗口标量乘方法和带符号的整数阶乘展开式的优势.与本文中所提其他算法相比,新算法在点加运算上效率相当,但省略了倍点运算.因此,新算法极大地提高了多标量乘的运算效率.值得注意的是,在实际应用且不失安全性的情况下,选取"好的"私钥 k_i(即 k_i 展开式中系数为 0 的个数接近于 k_i 展开式中系数总数的一半),可以使新算法的运算效率更高.

References:

[1] Brickell EF, Gordon DM, McCurley KS, Wilson DB. Fast exponentiation with precomputation. In: Rueppel RA, ed. Advances in Cryptology—Proc. of the EURO-CRYPT'92. LNCS 658, Berlin: Springer-Verlag, 1993. 200−207. [doi: 10.1007/3-540-47555-9_18]

[2] Roberto MA, Mathieu C, Francesco S. Faster scalar multiplication on koblitz curves combining point halving with the frobenius endomorphism. In: Bao F, *et al*., eds. Proc. of the PKC 2004. LNCS 2947, 2004. 28−40. [doi: 10.1007/978-3-540-24632-9_3]

[3] Yong KL, Ingrid V. A compact architecture for montgomery elliptic curve scalar multiplication processor. In: Kim S, Yung M, Lee HW, eds. Proc. of the Int'l Workshop on Information Security Applications (WISA 2007). LNCS 4867, Springer-Verlag, 2007. 115−127. [doi: 10.1007/978-3-540-77535-5_9]

[4] Institute of Electrical and Electronics Engineers, Inc. IEEE P1363/D4. In: Proc. of the Standard Specifications for Public-Key Cryptography. New York, 2001. http://www.jablon.org/passwordlinks.html

[5] Chen TS, Huang KH, Chung YF. Digital multi-signature scheme based on the elliptic curve cryptosystem. Journal of Computer Science and Technology, 2004,19(4):570−573. [doi: 10.1007/BF02944760]

[6] Lim CH, Lee PJ. More flexible exponentiation with precomputation. In: Pitt DH, *et al*., eds. Proc. of the 14th Annual Int'l Cryptology Conf. on Advances in Cryptology. LNCS 389, New York: Springer-Verlag, 1994. 95−107. [doi: 10.1007/3-540-48658-5_11]

[7] Clavier C, Joye M. Universal exponentiation algorithm. In: Koc CK, Naccache D, eds. Proc. of the CHES 2001. LNCS 2162, New York: Springer-Verlag, 2001. 300−308. [doi: 10.1007/3-540-44709-1_25]

[8] Solinas JA. Low-Weight binary representations for pairs of integer. Technical Report, CORR 2001-41, Centre for Applied Cryptographic Research. 2001. http://www.cacr.math.uwaterloo.ca

[9] Liu D, Dai YQ. A new algorithm of elliptic curve multi-scalar multiplication. Chinese Journal of Computers, 2008,31(7): 1131−1138 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2008.01131]

[10] Zhang YJ, Zhu YF, Kuang BJ. Low-Weight JSF3 representations for pairs of integers. Journal of Software, 2006,17(9):2004−2012 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/2004.htm [doi: 10.1360/jos172004]

[11] Shi RH, Zhong C. A fast method for scalar multiplication on elliptic curves. Computer Engineering and Applications, 2006,42(2): 156−169 (in Chinese with English abstract).

© 中国科学院软件研究所 http://www.jos.org.cn

[12] National Institute of Standards and Technology. Recommended elliptic curves for federal government use. 1999. http://www.csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

附中文参考文献:

[9] 刘铎,戴一奇.计算椭圆曲线上多标量乘的快速算法.计算机学报,2008,31(7):1131–1138. [doi: 10.3724/SP.J.1016.2008.01131]

[10] 张亚娟,祝跃飞,况百杰.整数对的低重量表示.软件学报,2006,17(9):2004–2012. http://www.jos.org.cn/1000-9825/17/2004.htm [doi: 10.1360/jos172004]

[11] 石润华,钟诚.一种快速的椭圆曲线标量乘方法.计算机工程与应用,2006,42(2):156–169.

陈厚友(1979－),男,山东微山县人,硕士生,主要研究领域为椭圆曲线密码学.

马传贵(1962－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为密码学,无线网络安全.

****

2011 年全国开放式分布与并行计算学术年会

征 文 通 知

由中国计算机学会开放系统专业委员会主办、华中科技大学计算机学院承办的“2011 全国开放式分布与并行计算学术年会(DPCS 2011)”将于 2011 年 8 月 16-19 日在湖北恩施召开.本次大会欢迎中英文投稿.录用的英文文章将由 IEEE 出版(EI 检索,优秀论文推荐到 SCI 国际期刊);录用的中文论文将以正刊方式发表在《微电子学与计算机》,优秀论文推荐到一级学报发表.有关征文事宜通知如下:

1、征文范围(包括但不限于):

(1) 开放式分布与并行计算模型、体系结构、编程环境、算法及应用;

(2) 开放式网络、数据通信、网络与信息安全、业务管理技术;

(3) 开放式海量数据存储与 Internet 索引技术,分布与并行数据库及数据/Web 挖掘技术;

(4) 开放式网格计算、云计算、Web 服务、P2P 网络及中间件技术;

(5) 开放式无线网络、移动计算、传感器网络与自组网技术;

(6) 分布式人工智能、多代理与决策支持技术;

(7) 开放式虚拟现实技术与分布式仿真;

(8) 开放式多媒体技术与流媒体服务,媒体压缩、内容分送、缓存代理、服务发现与管理技术.

2、论文必须是未正式发表的、或者未正式等待刊发的研究成果.稿件格式应包括题目、作者、所属单位、摘要、关键词、正文和参考文献等,具体格式参照网站提供的样式.务必附上第一作者简历(姓名、性别、出生年月、出生地、职称、学位、研究方向等)、通信地址、邮政编码、联系电话和电子信箱.并注明论文所属领域.来稿一律不退,请自留底稿.

3、中文投稿截止日期: 2011 年 6 月 1 日

中文投稿论文录用通知日期: 2011 年 6 月 15 日.

4、中文论文投稿: DPCS2011@gmail.com [务必在邮件标题中指明“DPCS2011 中文投稿”字样];英文论文投稿:按照 IEEE 格式撰写,不超过 6 页,详情参见网站 http://grid.hust.edu.cn/HumanCom2011/PDC2011.htm

5、会议承办方联系人、联系电话及 E-mail 信箱

华中科技大学计算机学院 廖小飞

电话: 027-87557047-8007, 13871453610

电邮: hustliaoxf@gmail.com

© 中国科学院软件研究所 http://www.jos.org.cn