基于滑动窗口技术的快速标量乘法

李 忠[1,2] 彭代渊[1]
(西南交通大学信息科学与技术学院 成都 610031)[1] (宜宾学院计算机与信息工程学院 宜宾 644000)[2]

摘 要 标量乘法是椭圆曲线密码体制的核心运算,它的有效实现是近年来信息安全领域研究的一个热点内容。借助于标量的 wMOF 表示思想,利用混合坐标表示下直接计算 2^kQ+P 的策略,改进了基于滑动窗口技术的标量乘法算法。分析表明,所得算法效率明显提升,并降低了存储需求,能有效提升 ECC 的实现效率。

关键词 椭圆曲线密码,标量乘法,滑动窗口技术,直接计算
中图法分类号 TP309.7 **文献标识码** A

Fast Scalar Multiplication Based on Sliding Window Technology

LI Zhong[1,2] PENG Dai-yuan[1]
(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China)[1]
(School of Computer & Information Engineering, Yibin University, Yibin 644000, China)[2]

Abstract Scalar multiplication is the heart of elliptic curve cryptosystems. In recent years, how to realize efficient scalar multiplication is a research focus of information security field. By means of the wMOF representations of scalar and the direct computation 2^kQ+P strategy, we modified the scalar multiplication algorithm based on sliding window technology. The analysis results indicate that the efficiency of the algorithms is improved obviously and the storage requirements are reduced, and it can enhance the ECC's efficiency.

Keywords *Elliptic curve cryptosysytem, Scalar multiplication, Sliding window technology, Direct computation*

RSA 是目前广泛使用的公钥密码体制,但密钥长度为 512bits 的 RSA(RSA-512)已被认为不安全了,很多场合都要使用 RSA-1024,预计 10 年后需要用到 RSA-2048 才能保证安全性。相对于 RSA 密码体制,椭圆曲线密码体制(ECC)具有高安全强度,ECC-160、ECC-224、ECC-256 和 ECC-384 分别提供 RSA-1024、RAS-2048、RAS-3072 和 RSA-8192 同等安全级别,ECC 还具有密钥短、功耗低、计算速度快等突出优点,特别适用于 WSN、PDA 等处理能力、存储空间、带宽、功耗受限的环境[1]。

标量乘法是 ECC 的核心运算,它的有效实现是近年来信息安全领域研究的一个热点内容,迄今已提出许多标量乘法算法,其中,基于 NAF 方法的标量乘法是无预计算类的典型代表,如果有额外存储空间可用,可通过预计算的方式大大提升标量乘法运算的效率,其中,基于 wNAF 方法的标量乘法是带预计算类的典型代表。文献[2-5]等对这些算法进行了讨论。文献[6]结合最新研究成果,讨论了基于窗口技术的标量乘法算法的最佳坐标选择,并利用直接计算 $2Q+P$ 策略对基于窗口技术的标量乘法算法进行了优化,提升了标量乘法运算的效率。本文在此基础上,借助于标量的 wMOF 表示思想,利用混合坐标表示下直接计算 2^kQ+P 的策略,对基于滑动窗口技术的标量乘法算法进行了优化,所得算法效率明显提高,并降低了存储需求,能有效提高 ECC 的实现效率。

1 椭圆曲线群运算

素域 $\mathbf{F}_p$ 上椭圆曲线 E 由简化的 Weierstrass 方程 $E: y^2=x^3+ax+b$ 确定,其中 $a、b\in\mathbf{F}_p$,$4a^3+27b^2\neq 0 \pmod p$。

点集 $E(\mathbf{F}_p)=\{(x,y)\in\mathbf{F}_p\times\mathbf{F}_p \mid y^2=x^3+ax+b\}\cup\{O\}$ 构成一个 Abelian 群,其中 O 为无穷远点,是群的单位元,称该群为椭圆曲线群。

1.1 椭圆曲线群运算法则

群 $E(\mathbf{F}_p)$ 运算依赖于曲线 E 上点的表示,在仿射坐标系下,点 $P=(x,y)$ 的负元 $-P=(x,-y)$。$P=(x_1,y_1)$、$Q=(x_2,y_2)\in E(\mathbf{F}_p)$,若 $P\neq\pm Q$,点加运算 $P+Q=(x_3,y_3)$ 定义为:

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_1-x_3)-y_1 \\ \lambda=(y_1-y_2)/(x_1-x_2) \end{cases} \tag{1}$$

若 $P\neq -P$,倍点运算 $2P=P+P=(x_3,y_3)$ 定义为:

$$\begin{cases} x_3=\lambda^2-2x_1 \\ y_3=\lambda(x_1-x_3)-y_1 \\ \lambda=(3x_1^2+a)/(2y_1) \end{cases} \tag{2}$$

分别用 I、M、S 表示域 $\mathbf{F}_p$ 上的求逆、乘法、平方运算,在上述仿射坐标表示下,点加运算的时间消耗为 $1I+2M+1S$,倍点运算的时间消耗为 $1I+2M+2S$。

本文受四川省教育厅重点科研项目(07ZA145)资助。

李 忠(1963—),男,博士生,副教授,主要研究方向为密码学、信息安全,E-mail:lz8056859@163.com;彭代渊(1955—),男,博士,教授,博士生导师,主要研究方向为密码学、信息安全、编码理论。

1.2 标量乘法运算

$k\in F_p$，$P=(x,y)\in E(F_p)$，称 $kP=P+\cdots+P$（k 个 P 相加）为椭圆曲线 E 上的标量乘法运算。对于二进制表示的标量 $k=(k_{l-1}\cdots k_2k_1k_0)_2$，有：

$$kP=(\sum_{i=0}^{l-1}k_i2^i)P=\sum_{i=0}^{l-1}(k_iP)2^i$$
$$=2(2\cdots2(2(2k_{l-1}P+k_{l-2}P)+k_{l-3}P)+\cdots+k_1P)+k_0P$$

可得从左向右的标量算法。

算法 1 基于二进制方法的从左向右的标量乘法[6]

输入：$k=(k_{l-1}\cdots k_2k_1k_0)_2$，$k_{l-1}=1$，$P\in E(K)$.

输出：$Q=kP$.

1. $Q=P$;
2. for i from l−2 downto 0 do
   2.1 if $k_i=1$ then $Q=2Q+P$;
   2.2 else $Q=2Q$;
3. return Q.

由算法 1、式(1)及式(2)可知，标量乘法运算的效率依赖于基域的运算，其中求逆是最耗时的运算，可以使用投影坐标表示将求逆运算转换为乘法等运算以提高效率。文献[6]根据最新研究成果，对比分析了点加运算、倍点运算在不同坐标表示下的效率，如表 1 所列。

表 1 不同坐标表示下点加运算、倍点运算的时间消耗[6]

点加运算	时间消耗	倍点运算	时间消耗
A+A=A	1I+2M+1S	2A=A	1I+2M+2S
J+J=J	11M+5S	2J=J	1M+8S
J+A=J	7M+4S	2A=J	1M+5S
$J^m+J^m=J^m$	11M+7S	$2J^m=J^m$	3M+5S
$J^m+A=J^m$	7M+6S	$2A=J^m$	2M+5S

说明：(1)A、J、J^m 分别代表仿射坐标、雅可比坐标、优化的雅可比坐标；(2)$J+A=J$ 代表点加运算 $Q+P$，其中 Q 为雅可比坐标，P 为仿射坐标，计算结果为雅可比坐标，其余类似。

由表 1 可知，平均情况下，采用 $J+A=J$ 进行点加运算，$2J=J$ 进行倍点运算是提升标量乘法运算效率的较佳组合选择[6]。

2 基于滑动窗口技术的快速标量乘法

2.1 基于滑动窗口技术的标量乘法

标量的 NAF 表示是数字集{1,0,−1}上具有最小平均汉明重量的标量表示法，基于滑动窗口技术的标量乘法算法依赖于标量的 NAF 表示，通过预计算的方式提升标量乘法运算的效率。

算法 2 基于滑动窗口技术的标量乘法[7]

输入：窗口宽度 w，正整数 k 的 NAF 表示 $k=(k_{l-1}\cdots k_1k_0)_{NAF}$，$P\in E(K)$.

输出：kP.

1. compute $P_i=iP$ for $i\in\{1,3,\cdots,2(2^w-(-1)^w)/3-1\}$;
2. Q=O, i=l−1;
3. while i≥0 do
   3.1 if $k_i=0$ then t=1, u=0;
   3.2 else: find the largest t≤w such that $u=(k_i\cdots k_{i-t+1})$ is odd;
   3.3 $Q=2^tQ$;
   3.4 if u>0 then $Q=2^tQ+P_u$;
   3.5 else if u<0 then $Q=2^tQ-P_{-u}$;
   3.6 i=i−t;
4. return Q.

算法 2 期望的运行时间近似为[7]：

$$1D+\left(\frac{2^w-(-1)^w}{3}-1\right)A+(l-1)D+\left(\frac{l}{w+v(w)}-1\right)A$$
$$=lD+\left(\frac{l}{w+v(w)}+\frac{2^w-(-1)^w}{3}-2\right)A \quad (4)$$

式中，$v(w)=\frac{4}{3}-\frac{(-1)^w}{3\cdot2^{w-2}}$，代表‘0’在窗口之间移动的平均长度；$A$ 表示点加运算；D 表示倍点运算。

2.2 优化的基于滑动窗口技术的标量乘法

文献[6]讨论了不同坐标表示下，直接计算 $2Q+P$ 的效率，如表 2 所列。

表 2 不同坐标表示下直接计算 2Q+P 的时间消耗[6]

2Q+P 运算	时间消耗	2Q+P 运算	时间消耗
2A+A=A	1I+9M+2S	2J+A=J	13M+5S
2A+A=J	11M+3S	2J+J=J	17M+6S
2A+J=J	15M+4S	$2J+A=J^m$	14M+7S

说明：表中各量的含义同表 1

文献[6]通过在仿射坐标系下进行预计算，采用 $2J+A=J$ 直接计算 $2Q+P$，$2J=J$ 计算 $2Q$，提升了基于滑动窗口技术的标量乘法运算的效率。

对于算法 2，如果某 $k_i=0$，则进行一次倍点运算（步骤 3.3），如果连续出现 s 个‘0’，则要连续进行 s 次倍点运算，再进行一次点加/减运算。文献[8]给出雅可比坐标下直接计算 2^kQ 的算法，其时间消耗为 $4kM+(4k+2)S$，我们借助于该思想对算法 2 进行优化。

算法 3 优化的基于滑动窗口技术的标量乘法

输入：窗口宽度 w，正整数 k 的 NAF 表示 $k=(k_{l-1}\cdots k_1k_0)_{NAF}$，$P\in E(K)$.

输出：kP.

1. compute $P_i=iP$ for $i\in\{1,3,\cdots,2(2^w-(-1)^w)/3-1\}$;
2. Q=O, i=l−1;
3. while i≥0 do
   3.1 s=0;
   3.2 while i≥0 and $k_i=0$ do
      3.2.1 s=s+1, i=i−1;
   3.3 if i≥0 then
      3.3.1 find the largest t≤w such that $u=(k_i\cdots k_{i-t+1})$ is odd;
      3.3.2 if u>0 then $Q=2^{s+t}Q+P_u$;
      3.3.3 else $Q=2^{s+t}Q+(-P_{-u})$;
      3.3.4 i=i−t;
   3.4 else $Q=2^sQ$;
4. return Q.

为便于算法效率比较，我们仍按文献[6]的假设：$1S=0.8M$，$1I=30M$。

于是有：

(1)在仿射坐标表示下，一次点加运算、倍点运算的时间消耗分别为：$1A=32.8M$，$1D=33.6M$。

(2)我们采用直接计算 2^kQ 的算法计算 $2^{k-1}Q$，再利用直接计算 $2Q+P$ 的算法计算 $2^kQ+P=2(2^{k-1}Q)+P$（点 Q 为雅可比坐标、点 P 为仿射坐标，计算结果为雅可比坐标），一次计算 2^kQ+P 的时间消耗为 $(4k+9)M+(4k+3)S=(7.2k+11.4)M$。

算法 3 的预计算（步骤 1）仍采用仿射坐标表示，按以下方式计算：

$2P,3P=P+2P,5P=3P+2P,7P=5P+2P,\cdots$

时间消耗为：

$$1D+\left(\frac{2^w-(-1)^w}{3}-1\right)A=33.6M+32.8\left(\frac{2^w-(-1)^w}{3}-1\right)M$$

在 NAF 表示下，经步骤 3.3.1，所得标量表示的平均汉明密度为 $1/(w+v(w))$，不失一般性，可假设表示结果为：

$$x\ \underbrace{0\cdots0}_{w+v(w)-1}\ x\ \underbrace{0\cdots0}_{w+v(w)-1}\ x\cdots\ \underbrace{0\cdots0}_{w+v(w)-1}\ x$$

平均情况下需要进行 $l/(w+v(w))-1$ 次 $2^{w+v(w)-1}Q+P_u$（或 $2^{w+v(w)-1}Q+(-P_{-u})$）运算，其时间消耗近似为：

$$\left(\frac{l}{w+v(w)}-1\right)[7.2(w+v(w)-1)+11.4]M$$

进而，算法 3 的平均时间消耗近似为：

$$33.6M+32.8\left(\frac{2^w-(-1)^w}{3}-1\right)M+\left(\frac{l}{w+v(w)}-1\right)[7.2(w+v(w)-1)+11.4]M \quad (5)$$

2.3 基于滑动窗口技术的快速标量乘法

利用算法 2、算法 3 进行标量乘法运算，必须先计算并存储标量 k 的 NAF 表示，这需要消耗额外的存储空间，同时，存储预计算结果也需要消耗较多的存储空间。我们借助类似于标量的 wMOF 表示思想[9]，从左向右扫描标量的各位，根据 $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})$ 及 (k_1,k_0) 的值按表 3 的规则对标量进行转换。

表 3 标量表示中位段转换规则

位段	结果	位段	结果	位段	结果	位段	结果
i00	i00	i10	0i0	i1$\bar{i}$	00$\bar{3}$	i01	00$\bar{3}$
1i0	010	1i1	003	10$\bar{i}$	003	100	100
00	00	01	01	10	i0	11	0i

说明：最后一行仅针对 (k_1,k_0) 的值使用，其中 $\bar{c}$ 代表 $-c$。

对于标量 $k=\sum_{i=0}^{l-1}k_i2^i$（其中 $k_i\in\{0,\pm3,\pm1\}$，且 $k_{l-1}\neq0$）及椭圆曲线 E 上的点 P，有：

$$\begin{aligned}k&=\sum_{i=0}^{l-1}k_i2^i\\&=k'_{t-1}2^{e_{t-1}}+k'_{t-2}2^{e_{t-2}}+\cdots+k_1'2^{e_1}+k_0'2^{e_0}\\&=2^{e_0}(\cdots(2^{e_{t-2}-e_{t-3}}(2^{e_{t-1}-e_{t-2}}k'_{t-1}+k'_{t-2})+k'_{t-3})+\cdots+k_0')\end{aligned}$$

其中，$k_i'\neq0$，$e_0<e_1<\cdots<e_{t-2}<e_{t-1}$。

于是：

$$kP=2^{e_0}(\cdots(2^{e_{t-2}-e_{t-3}}(2^{e_{t-1}-e_{t-2}}k'_{t-1}P+k'_{t-2}P)+k'_{t-3}P)+\cdots+k_0'P) \quad (6)$$

由表 3 及式(6)可得基于滑动窗口技术的快速标量乘法算法。

算法 4 基于滑动窗口技术的快速标量乘法

输入：二进制整数 $k=(k_{l-1}\cdots k_1k_0)_2$，$P\in E(\mathbf{F}_p)$.

输出：kP.

1. Q=(1:1:0),R=3P,k_l=0,k_{-1}=0,i=l,t=0;
2. while i≥2 do
   2.1 if $k_{i-1}=k_i$ then t=t+1,i=i−1;
   2.2 else if $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})=(-1,0,0)$ then Q=2^tQ-P,t=2;
   2.3 else if $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})=(-1,1,0)$ then Q=$2^{t+1}Q-P$,t=1;
   2.4 else if $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})=(-1,1,-1)$ or $(-1,0,1)$ then Q=$2^{t+2}Q-R$,t=0;
   2.5 else if $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})=(1,-1,0)$ then Q=$2^{t+1}Q+P$,t=1;
   2.6 else if $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})=(1,-1,1)$ or $(1,0,-1)$ then Q=$2^{t+2}Q+R$,t=0;
   2.7 else if $(k_{i-1}-k_i,k_{i-2}-k_{i-1},k_{i-3}-k_{i-2})=(1,0,0)$ then Q=2^tQ+P,t=2;
   2.8 i=i−3;
3. if i=1 then
   3.1 if $(k_1,k_0)=(0,0)$ then t=t+2;
   3.2 else if $(k_1,k_0)=(0,1)$ then Q=$2^{t+1}Q+P$,t=0;
   3.3 else if $(k_1,k_0)=(1,0)$ then Q=2^tQ-P,t=1;
   3.4 else if $(k_1,k_0)=(1,1)$ then Q=$2^{t+1}Q-P$,t=0;
4. else if i=0 then
   4.1 if k_0=0 then t=t+1;
   4.2 else if k_0=1 then Q=2^tQ-P,t=0;
5. if(t≠0) then Q=2^tQ;
6. return Q

3 算法效率对比分析

由文献[6]的表 4.3 可知，对于 NIST 素数 P-192、P-224、P-256，基于滑动窗口技术的标量乘法运算的最佳窗口宽度均为 $w=4$，而对于 NIST 素数 P-384 和 P-521 的最佳窗口宽度分别为 $w=5$ 和 $w=6$

根据式(5)，当窗口宽度 $w=4$ 时，算法 3 的时间消耗为 $(8l+122.8)M$，最后将雅可比坐标表示的点 Q 转换为仿射坐标表示，其时间消耗为 $1I+3M+1S=33.8M$，所以，取 $w=4$ 时按算法 3 的方式进行标量乘法运算，其时间消耗为 $(8l+156.6)M$，同理可得其它情形的时间消耗。

算法 4 的预计算只须计算一个点 $R=2P+P$，时间消耗为 $1I+9M+2S=40.6M$，标量按表 3 的方式转换的平均汉明密度为 1/4，平均情况下需进行 $l/4-1$ 次混合坐标下的 2^3Q+P 运算，时间消耗为 $(l/4-1)(7.2\times3+11.4)M=(8.25l-33)M$，同理，需要 $33.8M$ 的时间消耗将点 Q 转换为仿射坐标表示，所以，按算法 4 的方式进行标量乘法运算，其时间消耗为 $(8.25l+41.4)M$。

文献[6]、算法 3、算法 4 的时间消耗如表 4、图 1 所示。

表 4 基于滑动窗口技术标量乘法算法效率对比分析

	文献[6]	算法 3	
P-192、P-224	M-cost	(9.57l+69.6)M	(8l+156.6)M
P-256,w=4	#Pre	5	5
P-384,w=5	M-cost	(9.29l+149.35)M	(7.859l+338.1)M
	#Pre	11	11
P-521,w=6	M-cost	(9.08l+283.1)M	(7.774l+666.55)M
	#Pre	21	21
算法 4	时间消耗为(8.25l+41.4)M，仅预计算 R=3P		

说明：M-cost、#Pre 分别代表相应的时间消耗和预计算量(点数)

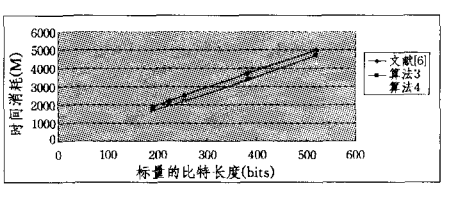

图 1 基于滑动窗口技术的标量乘法效率分析

结束语 本文利用混合坐标下的直接计算技术对基于滑

(下转第 64 页)

员的编程习惯不同，某些应用程序可能会随着用户输入不同产生可变的 SQL 语句，这给本方法带来很大挑战，一方面导致知识库无法完全学习到所有合法 SQL 语句，另一方面由于可变语句的存在，应用程序产生的 SQL 语句的数量不可控，因此大量可变语句的出现将导致知识库巨大。虽然可以通过配合特征过滤方法来解决大部分的可变 SQL 语句，但是依然无法从根本上解决。我们也注意到可变语句的产生在很大程度上有其特有的共性，比如相关的语句结构具有很大的相似性，变化的只是 SQL 语句的局部。因此，在未来的研究工作中，我们考虑采用基于 SQL 语法树的相似度进行模式匹配，实现对可变 SQL 语句的处理。

参 考 文 献

[1] Patel N, Mohammed F, Soni S. SQL Injection Attacks: Techniques and Protection Mechanisms[J]. International Journal on Computer Science and Engineering, 2011, 3(1)

[2] Das D, Sharma U, Bhattacharyya D K. An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching [J]. 2010 International Journal of Computer Applications, 2010, 1(25)

[3] Howard M, LeBlanc D. Writing Secure Code (Second edition) [M]. Microsoft Press, Redmond, Washington, 2003

[4] Huang Y, Huang S, Lin T, et al. Web Application Security Assessment by Fault Injection and Behavior Monitoring[C]// Proceedings of the 11th International World Wide Web Conference (WWW 03). May 2003

[5] Gould C, Su Z, Devanbu P. JDBC Checker: A Static Analysis Tool for SQL/JDBC Applications[C]// Proceedings of the 26th International Conference on Software Engineering (ICSE 04). Formal Demos, 2004: 697-698

[6] Maor O, Shulman A. Sql injection signatures evasion[EB/OL]. http://www.imperva.com/application_defense_center/white_papers/sql_inje%ction_signatures_evasion.html, White paper, 2004-04

[7] Ashcraft K, Engler D. Using programmer-written compiler extensions to catch security holes[C]// Proceedings of the IEEE Symposium on Security and Privacy. May 2002: 143-159

[8] Buehrer G, Weide B, Sivilotti P. Using Parse Tree Validation to Prevent SQL Injection Attacks[C]// Proceedings of the 5th International Workshop on Software Engineering and Middleware. Lisbon, Portugal, September 2005

[9] Su Z, Wassermann G. The Essence of Command Injection Attacks in Web Applications[C]// Proceedings of the 33rd Symposium on Principles of Programming Languages. Charleston, South Carolina, January 2006

[10] 周敬利，等. 一种新的反 SQL 注入策略的研究与实现[J]. 计算机科学，2006，133(111)

[11] 余静，等. 基于 SQL 注入的渗透性测试技术研究[J]. 计算机工程与设计，2007，28(15)

[12] 陈小兵，等. SQL 注入攻击及其防范检测技术研究. 计算机工程与设计，2007，43(11)

(上接第 56 页)

动窗口技术的标量乘法算法进行了改进。由表 4 和图 1 可知，所获得的改进算法的效率有明显提高，并降低了空间消耗，能有效提升 ECC 的实现效率。

参 考 文 献

[1] Noroozi E, Kadivar J, Shafiee S H. Energy analysis for wireless sensor networks[C]// IEEE International Conference on Mechanical and Electronics Engineering (ICMEE 2010). IEEE, 2010(2): 382-386

[2] Longa P, Gebotys C. Setting speed records with the (fractional) multibase non-adjacent form method for efficient elliptic curve scalar multiplication[C]// Department of Electrical and Computer Engineering University of Waterloo, Canada, 2009

[3] Shah P G, Huang X, Sharma D. Algorithm Based on One's Complement for Fast Scalar Multiplication in ECC for Wireless Sensor Network[C]// IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA). 2010: 571-576

[4] Okeya K, Schmidt-Samoa K, Spahn C, et al. Signed Binary Representations Revisited[C]// Advances in Cryptology- CRYPTO 2004(LNCS3152). New York: Springer-Verlag, 2004: 123-139

[5] Huang X, Shah P G, Sharma D. Minimizing Hamming Weight Based on 1's Complement of Binary Numbers Over $GF(2^m)$[C]// The 12th International Conference on Advanced Communication Technology (ICACT). Gangwondo, Korea, Feb. 2010, 2: 1226-1230

[6] Mahdavi R, Saiadian A. Efficient Scalar Multiplications for Elliptic Curve Cryptosystems Using Mixed Coordinates Strategy and Direct Computations[C]// Cryptology and Network Security (LNCS6467). 2010: 184-198

[7] Hankerson D, Menezes A, Vanstone S. Guide to elliptic curve cryptography[M]. Springer-Verlag Professional Computing Series, 2004

[8] Sakai Y, Sakurai K. Efficient Scalar Multiplications on Elliptic Curves with Direct Computations of Several Doublings[J]. IEICE Transaction on Fundamentals, 2001, E84(A): 120-129

[9] Okeya K, Schmidt-Samoa K, Spahn C, et al. Signed binary representations revisited[C]// Andances in Cryptology- CRYPTO' 04, LNCS 3152. 2004: 123-139