基于域 $GF(2^m)$ 上的椭圆曲线中标量乘的快速算法*)

张 宁 牛志华 肖国镇

(西安电子科技大学 ISN 国家重点实验室 信息保密研究所 西安 710071)

摘 要 标量乘法的快速运算是椭圆曲线密码学中研究的一个焦点。本文讨论基于域 $GF(2^m)$ 的非超奇异椭圆曲线上 $2P+Q$ 运算,给出了在域 $GF(2^m)$ 中的椭圆曲线点此类运算的一个完整的改进算法,并对算法做了简单的分析。得出结论:我们所给出的算法比 IEEE 给出的标准算法效率提高 10%以上。

关键词 $GF(2^m)$ 上的椭圆曲线,标量乘法,快速算法

Fast Scalar Multiplication of Elliptic Curves over Finite Field $GF(2^m)$

ZHANG Ning NIU Zhi-Hua XIAO Guo-Zhen

(Information Security & Privacy Institute in ISN, Xidian University, Xi'an 710071, P. R. C)

Abstract The paper discussed the algorithm for computing Scalar Multiplications on none-supersingular elliptic curves defined over $GF(2^m)$. Two algorithms to compute $2P+Q$ was given and an analysis of the algorithms implemented in the addition-subtraction method in IEEE was made. At last, we conclude that the new method makes it over 10% higher in efficiency.

Keywords Elliptic curve over $GF(2^m)$, Scalar multiplication, Fast algorithm

1 引言

1985 年,Kibitz 和 Miller 分别独立地提出了椭圆曲线密码体制(ECC)。因其具有较高的安全性,椭圆曲线密码在当今的密码学界引起了广泛的关注,但是在实现上因为速度比较慢,不能让人满意,影响了实际的应用,所以优化椭圆曲线的算法就成为 ECC 研究的一个重要课题。

标量乘是 ECC 中最常用的一种计算,相当于有限域中的指数运算。现有的计算标量乘的方法中,最常用的是 IEEE P1363[3] 给出的加-减(addition-subtraction)法,使用了数的 NAF(non-adjacent-form)表示法。在这种方法中,有近乎 1/3 的运算是 $2P+Q$ 形式的运算。本文就这种形式的运算,对 $GF(2^m)$ 上的椭圆曲线中的算法做详细的讨论。

以下文章如下安排:第 2 节介绍 $GF(2^m)$ 域上标量乘的基本算法,第 3 节提出一种计算 $2P+Q$ 的改进方法,第 4 节对这种新算法进行分析,给出具体的例子。

2 标量乘的基本算法

$GF(2^m)$ 中的非超奇异椭圆曲线 waitrass 一般表示为

$$E: y^2+xy=x^3+ax^2+b$$

其中 $x,y,a,b\in GF(2^m)$,且 $b\neq 0$。

根据椭圆曲线中点的运算法则,对于 E 上的点 $P=(x_1,y_1)$ 和 $Q=(x_2,y_2)$,令 $T=P+Q=(x_3,y_3)$,则加法运算定义如下:

①$x_1=x_2$ 时,有两种情况:当 $P=-Q$ 时,$P+Q=O$,O 定义为椭圆曲线上无穷远点;$P=Q$ 时为倍点运算:

$$\begin{cases} x_3=\lambda^2+\lambda+a \\ y_3=(x_1+x_3)\lambda+x_3+y_1 \end{cases} \tag{1}$$

其中 $\lambda=\dfrac{y_1}{x_1}+x_1$

②$x_1\neq x_2$ 时,即为点加运算

$$\begin{cases} x_3=\lambda^2+\lambda+x_1+x_2+a \\ y_3=(x_1+x_3)\lambda+x_3+y_1 \end{cases} \tag{2}$$

其中 $\lambda=\dfrac{y_1+y_2}{x_1+x_2}$

在上述运算下,椭圆曲线中的点构成了一个加法 Abel 群。在这两种情况下,各用了一次平方、两次乘法、一次求逆(忽略加法运算)。用 S 表示平方,M 表示域乘,I 表示求逆,则点加及倍点的运算量均为 $1S+2M+1I$。给定椭圆曲线上点 P 和一个大整数 k(一般而言,k 为长度为 m 的二进制数),计算 kP,即用椭圆曲线上加法将 P 与自身连续相加 k 次,称为椭圆曲线上标量乘法。一般的标量乘法就是根据上面的两个基本算法构造起来的,定义为对于椭圆曲线上的点 $P=(x_1,y_1)$、大整数 $k\in(0,2^m)$,计算 kP。下面的算法是 IEEE P1363 给出的加-减(addition-subtraction)法。

用 NAF 法表示 k,$k=(e_{m-1}e_{m-2}\cdots e_0)$,其中 $e_i=-1,0,1$,$0\leqslant i<m$。算法如下:

输入 $P=(x_1,y_1)$, $k=(e_{m-1}e_{m-2}\cdots e_0)_{NAF}$
$Q\leftarrow P$
For $i=m-2$ downto 0 do
$Q\leftarrow 2Q+e_iP$
输出 Q

平均情况下,NAF 表示法中非零位的个数为 $\dfrac{1}{3}(m-1)$。在上面的算法中,有 $m-1$ 次倍点运算和 $\dfrac{1}{3}(m-1)$ 点加运

*)基金项目:国家重点基础研究发展规划项目(97-3 项目)(项目编号:G1999035804)。张 宁 在读博士。

算，这样所有的运算为$\frac{4}{3}(1S+2M+1I)(m-1)$次。对于这种算法，效率的提高有三种不同的思路：第一为改变k的表示方法，改二进制为其它进制的表示；第二为提高倍点运算的效率；第三是考虑将$2P+Q$的运算量减小，将它作为一种独立的运算方式，那么这样的运算方式要做$\frac{1}{3}(m-1)$次，倍点运算减少为$\frac{2}{3}(m-1)$次。

3 域$GF(2^m)$中$2P+Q$的快速算法

椭圆曲线上点的标量乘的运算有两类：一为减少某种域$GF(2^m)$上的运算，比如域上的乘法次数；第二种考虑求逆运算一般是乘法运算量的3～10倍，以增加乘法运算的代价将求逆运算次数减少。本节参考文[1]和[2]中的思路，给出针对$GF(2^m)$上点的标量乘法的快速算法。

3.1 计算$2P+Q$的基本算法

输入点$P=(x_1,y_1)$和点$Q=(x_2,y_2)$，这里$P,Q\neq O$，计算$2P+Q$。先考虑一般情况，$P\neq\pm Q$时，先计算$P+Q=(x_3,y_3)$，再计算$2P+Q=P+(P+Q)=(x_4,y_4)$。利用(1)、(2)式，得出

$$\begin{cases}x_3=\lambda_1^2+\lambda_1+x_1+x_2+a\\ y_3=(x_1+x_3)\lambda_1+x_3+y_1\end{cases},\lambda_1=\frac{y_1+y_2}{x_1+x_2}$$

或者

$$\begin{cases}x_3=\lambda_1^2+\lambda_1+a\\ y_3=(x_1+x_3)\lambda_1+x_3+y_1\end{cases}\quad\lambda_1=\frac{y_1}{x_1}+x_1$$

$$\begin{cases}x_4=\lambda_2^2+\lambda_2+x_1+x_3+a\\ y_4=(x_1+x_4)\lambda_2+x_4+y_1\end{cases},\lambda_2=\frac{y_1+y_3}{x_1+x_3}$$

这样，直接计算的运算量为$2(1S+2M+1I)$。

3.2 改进算法

在计算两个λ时进行了两次求逆运算。考虑使求逆运算的次数减少，只做一次逆运算。要求有$I=f_1/f_2(x_1+x_2)(x_1+x_3)$，有$(x_1+x_2)^2(x_1+x_3)(y_1+y_2)(y_1+y_2+x_1+x_2)+(x_2+a)(x_1+x_2)^2$。令$A=(x_1+x_2)^2(x_1+x_3)$，$B=(x_1+x_2)A$，$I=B^{-1}$，则$\lambda_1=(y_1+y_2)AI$，$\lambda_2=\lambda_1+1+x_1(x_1+x_2)^3I$。同时注意到$x_4=\lambda_1^2+\lambda_2^2+\lambda_1+\lambda_2+x_2=(\lambda_1+\lambda_2)^2+\lambda_1+\lambda_2+x_2$。

$P=Q$时，有$\lambda_1=y_1/x_1+x_1$，$x_3=\lambda_1^2+\lambda_1+a$，$A=x_1^4+x_1^3+b$。令$B=x_1A$，$I=B^{-1}$，则$\lambda_1=y_1AI+x_1$，$\lambda_2=\lambda_1+1+x_1^2x_1^2I$，$x_4=\lambda_1^2+\lambda_2^2+\lambda_1+\lambda_2+x_1=(\lambda_1+\lambda_2)^2+\lambda_1+\lambda_2+x_1$。

改进算法如下：

输入$P=(x_1,y_1),Q=(x_2,y_2)$
输出$T=(x,y)$
if$(x_1=x_2)$\{
　if$(y_1\neq y_2)$
　　输出$T=P=(x,y)=(x_1,y_1)$
　　else　\{$S\leftarrow x_1^2$
　　$A\leftarrow S^2+x_1S+b$
　　$B\leftarrow x_1A$
　　$I\leftarrow B^{-1}$
　　$\lambda_1\leftarrow y_1AI+x_1$
　　$\lambda_2\leftarrow\lambda_1+1+S^2I$
　　\}
else\{
　　$S\leftarrow(x_1+x_2)^2$
　　$A\leftarrow(y_1+y_2)(y_1+y_2+x_1+x_2)+(x_2+a)S$
　　$B\leftarrow(x_1+x_2)A$
　　$I\leftarrow B^{-1}$
　　$\lambda_1\leftarrow(y_1+y_2)AI$
　　$\lambda_2\leftarrow\lambda_1+1+x_1(x_1+x_2)SI$
　　\}
$x\leftarrow(\lambda_1+\lambda_2)^2+\lambda_1+\lambda_2+x_2$
$y\leftarrow(x_1+x)\lambda_2+x+y_1$
输出$T=(x,y)$

以上算法考虑了三种P和Q的关系。当$P=Q$时，运算量为$4S+6M+1I$；$P\neq Q$时，运算量为$2S+9M+1I$。

4 算法分析

本节根据有限域$GF(2^m)$上各种运算的计算时间，分析上面给出的算法的效率。表1取自Lydia[4]（时间单位：μs）。

表1 Lydia表

m的值	加法运算时间	平方运算时间	乘法运算时间	求逆运算时间
163	0.6	2.3	10.05	96.2
191	0.7	2.0	10.9	118.1
239	0.8	2.6	14.6	162.8

改进算法用于加-减(addition-subtraction)法中，总运算量为：

$$[\frac{2}{3}(1S+2M+1I)+\frac{1}{3}(2S+9M+I)](m-1)$$

在m取163，191，239时，同标准算法相比，算法效率分别提高了8.8%，14.6%和14.9%。

本算法用在域$GF(2^m)$上椭圆曲线上点的标量乘法时，的确提高了算法效率，缩短了计算时间。并且，随着m值的增大，求逆运算时间与乘法运算时间的比值增大，改进算法的运用会使运算效率提高的比例越来越大。

参考文献

1 Eisentrager K, Lauter K, Montgomery P L. Fast elliptic curve arithmetic and improved Weil pairing evaluation. In Joye M, ed. Topics in Cryptology, Vol. 2612 of Lecture Notes in Computer Science, Springer-Verlag, 2003. 343～354

2 Ciet M, Joye M, Lauter K, et al. Trading Inversions for Multiplications in Elliptic Curve Cryptography. http://eprint.iacr.org/2003/257

3 IEEE P1363: Editorial Contribution to Standard for Public Key Cryptography, draft, 1998

4 LiDIA Group LiDIA v1.3- A library for computational number theory. THDarmstadt