文章编号:1672 - 058X(2012)06 - 0047 - 03

基于 NAF_w 的二进制域乘法算法*

蒋洪波[1], 吴 岩[1], 冯新宇[1], 杜艳秋[1], 杨庆江[1], 史克英[2], 刘艳伟[2]
(1. 黑龙江科技学院 电气与信息学院,哈尔滨 150027;2. 黑龙江大学,哈尔滨 150001)

摘 要:椭圆曲线上的乘法运算速度是提高椭圆曲线加密(ECC)性能的一个关键;分析了宽度 w 的非相邻表示型(NAF)算法和多项式乘法算法,提出了一个基于 NAF_w 的二进制域乘法算法;算法减少了运算中的异或运算次数和预计算个数,缩短了运算时间且节省了存储空间;经建模仿真,结果表明本算法运算效率较 comb 多项式乘法平均快 14.7% 左右,预计算只需要计算 $2^{w-1}-1$ 个,从存储预计算个数和时间消耗综合考虑 $w=4$ 也是较优的窗口宽度选择。

关键词:椭圆曲线;非相邻表示型;二进制域
中图分类号:TN918.4 **文献标志码**:A

信息化的高速发展促使了密码学的快速发展,而椭圆曲线以其安全性能高、处理速度快、计算量小及存储空间占用小的优点和其离散对数的难解性在密码学中占有重要的地位。椭圆曲线又分为质数曲线和二元曲线,通常二元曲线上的基本运算也称做二进制域的运算。二进制域($GF(2^m)$)上的基本运算由于实现起来简单快捷而受到青睐,椭圆曲线二进制域上的乘法算法主要有移位加和多项式乘法等。目前,最快的乘法算法之一是多项式乘法,但该算法需要存储 2^w-1 个预计算值。在此算法是在窗口宽度 w 的非相邻表示型(NAF_w)算法基础上提出的,它可以在计算乘法时减少运算次数,从而达到快速计算的要求,同时它需要存储的预计算个数也是多项式乘法的一半,节省了存储单元。

1 二进制数的 NAF 表示

设实现平台的字长是 W 位,且 W 为 8 的整数倍。一个字 U 的 W 位,分别以 0 到 $W-1$ 标记,且第 0 位为最高有效位。

设 $f(z)$ 是一个 m 次的二进制既约多项式,记做 $f(z)=z^m+r(z)$,则 $GF(2^m)$ 上的元素是次数最多为 $m-1$ 的二进制多项式。因此,域元素的加法即二进制多项式的加法。域元素的乘法即两个二进制多项式相乘再对 $f(z)$ 取模的结果。

一个域元素 $a(z)=a_{m-1}z^{m-1}+\cdots+a_2z^2+a_1z+a_0$ 与一个 m 维向量 $a=(a_{m-1},\cdots,a_2,a_1,a_0)$ 相对应。令 $s=\lfloor m/W \rfloor$,$t=Ws-m$。软件实现中 a 可以用 s 个 W 字的一个数组 $A=(A[s-1],\cdots,A[1],A[0])$ 来存储,最高有效位 a_0 存储到 $A[0]$ 中,并且 $A[s-1]$ 的最左 t 位没有用(总是设置为 0)。

收稿日期:2012 - 01 - 15;**修回日期**:2012 - 02 - 17.
* **基金项目**:黑龙江省教育厅科学技术研究(指导)项目(11553092).
作者简介:蒋洪波(1978-),女,黑龙江省鸡西人,讲师,硕士,从事 SoC 和椭圆曲线加密研究.

NAF 称作非相邻表示型。宽度为 w 的 NAF 记为

$$\mathrm{NAF}_w(n) = (k_{t-1},\cdots,k_0)$$

令 $w\geqslant 2$,那么每一个正整数 k 都有惟一的宽度为 w 的 NAF 表达式

$$k = \sum_{i=0}^{l-1} k_i 2^i$$

其中每一个非零系数 k_i 都是奇数,$0<k_i<2^w$,$k_{l-1}\neq 0$,并且任何连续 w 个数字中最多 1 位为非零。文献[3]中给出了如算法 1 所示的计算正整数窗口宽度为 w 的 NAF 算法。

算法 1 计算一个正整数窗口宽度为 w 的 NAF 算法。

输入:一个正整数 k;输出:非相邻表示型 $\mathrm{NAF}_w(k)$。

(1) $c\leftarrow k$;

(2) 当 $c>0$,重复执行,如果 c 是奇数,那么 $u\leftarrow c \bmod 2^w$,$c\leftarrow c-u$,否则 $u\leftarrow 0$。在 S 中将 u 追加在前 $c=c/2$;

(3) 返回 S,算法 2 中若令 $k=629$,那么 k 的二进制($w=2$)表示为$(k)_2=10\quad 0111\quad 0101$,当 $w=4$ 时用算法 2 计算的结果为 $\mathrm{NAF}_4(k)=10\quad 0007\quad 0005$。

从中可以看出两个非零数字之间 0 的个数至少是 3,对于任意的 $a,b\in \mathrm{GF}(2^m)$,将 a 用算法 1 转换成窗口宽度为 w 的 NAF 序列,这时序列中的非零数值的取值范围是$\{1,3,5,\cdots,2^w-1\}$,因此预计算只需计算 $2^{w-1}-1$ 个值即可,该计算量比 comb 少了一半。

2 二进制域乘法算法分析

用文献[1]中的计算方法计算多项式乘法,则从序列中的第 l 位开始,每遇到一个非零数值就查表取得预计算的乘积,做按位异或和累加处理,然后再查找两个非零数字之间的间隔 j,将刚得到的累加结果左移 $j+1$ 位。这样就得到了如算法 2 所示的基于 NAF_w 的二进制域乘法算法。

算法 2 基于 NAF_w 的二进制域乘法算法。

输入:次数低于 m 的二进制多项式 $a(z),b(z)$;输出:多项式乘积 $S(z)=a(z)\cdot b(z)$。

(1) 对于 $u\in(1,3,5,\cdots,2^w-1)$,计算 $B[u]=u(z)\cdot b(z)$;

(2) $i=0,S=0$;

(3) 若 $i<=m-1$ 或 $a(z)\neq 0$,则①取 $a(z)$ 的第 $i+\omega-1,i+\omega-2,\cdots,i$ 位为 u;若 u 为奇数则 $C=B[u]+C$;$i=i+4$;否则 $i=i+1$;$C=0$;将 C 的低 i 位追加到 S 前,C 右移 i 位,将 $a(z)$ 的第 i 位之后清零。

(4) 返回 S。

3 实验结果分析

López 和 Dahab 在文献[2]中提出了 comb 多项式乘法算法。comb 多项式乘法中需要的异或运算和移位运算分别为预计算的异或次数 $+(W/w)\times s$ 次 m/W 字异或和预计算的移位次数 $+2(W/w-1)$ 次 m/W 字移位。NAF_w 乘法的非零元素的平均密度近似为 $1/(w+1)$[3],因此平均异或次数为预计算的异或次数 $+m(w+1)$,平均移位次数为预计算的移位次数 $+m/(w+1)-1$。

在域 GF(2233)上对 comb 乘法算法和基于 NAF_w 的二进制域乘法算法用 C 建模,在 linux 平台下用 gcc 编译仿真,测试万组随机数据,得到如表 1 所示的实验结果。

表 1　$m=233$ 时 comb 多项式乘法和 NAF_w 乘法算法预计算个数和运行时间比较

窗口宽度 w	预计算个数、消耗时间/μs	comb 乘法	NAF_w 乘法
3	预计算个数	6	3
	消耗时间	487	291
4	预计算个数	14	7
	消耗时间	259	221
5	预计算个数	30	15
	消耗时间	357	198

本仿真实现平台为 $W=32$。实验结果分析如下:首先算法 2 的时间复杂度是 $O(m/w)$,其对算法 3 的时间复杂度影响并不大。第二,由于实现平台为 32 位,所以当窗口宽度为 3 或 5 这样的非 32 的公约数时,comb 乘法计算时需要进行位的拼接,从而影响了运算速度,而 NAF_w 乘法除去了位拼接的麻烦,因此在实验中 comb 多项式乘法的窗口宽度为 3 和 5 时的运算时间都较 $w=4$ 时长,而 NAF_w 乘法却随着窗口宽度的加大运算时间也在变短。

从异或移位运算和预计算角度看,comb 多项式乘法算法中当窗口宽度是 3 时,其异或运算和移位运算都要较 $w=4$ 时多,所以消耗时间上较多。$w=5$ 时虽然异或和移位次数减少,但预计算量增大的数量要比减少的异或和移位次数多。

NAF_w 乘法算法中随着窗口宽度的增大,移位和异或运算次数都在减少,且减少量比增加的预计算量要多。因此表 1 中的"NAF_w 乘法"列运算时间是递减趋势,而预运算个数却是呈递增趋势。当 NAF_w 中的窗口宽度 w 越大运行时间就会越小,预运算个数越多。综合预计算要存储的数值个数和消耗时间,从理论和实验结果分析可以得出窗口宽度 $w=4$ 时有较强的时间和空间上的优越性。

4　总　结

在对文献[3]提出的 NAF_w 算法进行了拓展应用,将其应用在二进制域乘法中。提出了一种基于 NAF_w 的二进制域乘法算法,通过理论分析和建模仿真验证,该算法只需要 $2w-1-1$ 个预计算值,同时计算效率相对于 comb 多项式乘法(窗口宽度 w)平均提高 14.7% 左右,在存储空间和计算时间上有很大的改进。

参考文献:

[1] MllER B. Improved techniques for fast exponentiation [C]. Information Security and Cryptology 2002 (LNCS 2587) [277], 2003:298-312

[2] LOPEZ J, DAHAB R. High-speed software multiplication in F2m [C]. Progress in Cryptology — INDOCRYPT2000(LNCS1977) [393],2000:203-212

[3] JEROME A,SOLINA S. Efficient Arithmetic on Koblitz Curves [J]. Designs,Codes and Cryptography,2000(19):195-249

[4] DARREL H,ALFRED M,SCOTT V. Guide to Elliptic Curve Cryptography [M]. 2004:92-97

[5] 李忠,王毅,彭代渊. 基于滑动窗口技术的有限域 GF(2n)乘法算法[C]. 何大可,黄月江. 密码学进展:中国 密码学会 2007 年会论文集. 成都:西南交通大学出版社出版,2007:123-129

(下转第 56 页)