大整数模幂的固定基窗口组合算法*

瞿云云[1a], 包小敏[2], 刘 花[2], 徐 洋[1b]

(1. 贵州师范大学 a. 数学与计算机科学学院; b. 贵州省信息与计算科学重点实验室, 贵阳 550001; 2. 西南大学数学与统计学院, 重庆 400715)

摘 要: 模幂乘运算是实现公钥密码体制的一个很重要的运算,其运算速度从整体上决定了公钥密码体制的实现效率。通过采用预处理技术,将椭圆曲线的定点标量乘的固定基窗口方法应用在模幂运算中,与 SMM 算法进行组合得到一种新的求模幂乘算法——固定基窗口方法。对算法的原理与效率进行了分析,实验结果表明,算法的运算速度得到了有效提高。

关键词: RSA; 模幂运算; SMM 算法; 固定基窗口方法

中图分类号: TP309 **文献标志码**: A **文章编号**: 1001-3695(2013)03-0679-03

doi:10.3969/j.issn.1001-3695.2013.03.008

Fixed base windowing combination algorithm for large integer modular exponentiation

QU Yun-yun[1a], BAO Xiao-min[2], LIU Hua[2], XU Yang[1b]

(1. *a. School of Mathematics & Computer Science, b. Key Laboratory of Information & Computing Science of Guizhou Province, Guizhou Normal University, Guiyang* 550001, *China*; 2. *School of Mathematics & Statistics, Southwest University, Chongqing* 400715, *China*)

Abstract: Modular exponentiation is an important operation of public-key cryptosystems, which heavily determines the overall implementation of the efficiency of a public-key cryptosystems. This paper proposed a new modular exponentiation algorithm named fixed base windowing algorithm. By precomputation, this algorithm utilized the fixed base windowing algorithm of scalar multiplication of elliptic curve combined with SMM algorithm to compute $g^k \bmod n$. Furthermore, it presented the principle and efficiency analysis of the new algorithm. At last, experimental results show that the computational efficiency has been increased effectively.

Key words: RSA; modular exponentiation; SMM algorithm; fixed base windowing algorithm

0 引言

在现代密码学中,公钥密码体制占有重要的地位。安全性基于大整数分解问题的 RSA 算法[1]是目前应用得最为广泛的公钥密码算法,其公钥和私钥都是两个大素数的函数。为了获得可以接受的安全性,电子商务的 SET(secure electronic transaction)协议规定 CA(certificate authority)使用 2 048 bit 的 RSA 密钥,其他实体使用 1 024 bit 的 RSA 密钥, 在大多数公钥密码算法中,都需要处理高达 1 024 bit 的大整数。这样,在使用公钥密码算法对数据进行处理时,需要进行大量的模幂运算。大整数的模幂运算是极其关键且非常耗时的,模幂运算的速度直接影响到公钥密码的有效实现。因此,研究模幂运算的快速实现引起了研究者的广泛关注[2]。对于模幂运算 $g^k \bmod n$,根据密码算法的实际应用可分为以下三类[3]:

a)指数 k 变化,底数 g 固定。主要应用于 ElGamal 加密和签名以及 Diffie-Hellman 密钥分配协议等。

b)底数 g 变化,指数 k 固定。主要应用于 RSA 加/解密等。

c)底数 g 和指数 k 均可变化。

对 a)类模幂运算,主要技术是预处理。由于底数 g 固定,可以消耗一些存储空间预计算存储一些 g 的固定指数的值来提高效率。对此,Brickell 等人[4]提出了 BGMW 算法;Rooij[5]提出了一种改进算法,在 BGMW 算法的基础上采用了加法链术。Dimitrov 等人[6]给出了将指数表示成双基数形式的预处理方法。其后,Dimitrov 等人[7]给出将指数表示成形如 2^i3^j 的和的改进方法,即 $k=\sum_{i,j} d_{i,j}2^i3^j, d_{i,j}\in\{0,1\}$,在此基础上通过预计算求 $g^k \bmod n$,称为双基记数系统(double-based number system,DBNS)。对于 b)类模幂运算,算法主要有 Knuth 于 1969 年提出的加法链算法及其一系列改进[8],由于指数 k 固定,主要方法是寻找 k 的一个最短的加法链。对于 c)类模幂运算,算法主要有二进制方法、m-ary 方法[9]、因子方法、窗口算法[10]、并行算法[11]等。Lou 等人[12]给出了 m-ary 方法的改进方法,称为 2^ω-ary 方法。

本文通过采用预处理技术,将椭圆曲线的定点标量乘的固定基窗口方法[13]应用在模幂运算中,组合 SMM 算法提出一种求 $g^k \bmod n$ 模幂剩余的算法——固定基窗口方法。该算法适合于 a)类模幂运算,将该算法与二进制平方乘算法作比较,

收稿日期: 2012-08-04; **修回日期**: 2012-09-30 **基金项目**: 国家自然科学基金资助项目(11001061,61070243,41161065);贵州省科学技术厅、贵州师范大学联合科技基金资金资助项目(黔科合 J 字 LKS[2011]15 号);贵州省科学技术基金资助项目(黔科合 J 字[2011]2213)

作者简介: 瞿云云(1983-),男,贵州金沙人,讲师,硕士,主要研究方向为密码学(qucloud@163.com);包小敏(1959-),男,新疆巴楚人,教授,博士,主要研究方向为密码学、信息安全、组合数学、群论;刘花(1987-),女,云南个旧人,硕士研究生,主要研究方向为密码学;徐洋(1983-),男,山东聊城人,副教授,博士(后),主要研究方向为信息安全.

进行了算法的原理与效率分析,当固定窗口宽度 ω,即固定基 2^ω,若指数 k 的二进制长度越长,模幂运算的效率提高程度越大;在进行大整数的模幂运算时,为保证较高的运算速度,笔者推荐选取窗口宽度 $\omega=2,3,4$。

1 预备知识

1.1 二进制算法

对于二进制算法[3]求 $g^k \bmod n$,所采用的是重复平方求模和相乘后求模的迭代方法来实现,k 的二进制表示的非零数字个数及长度分别表示乘法运算的次数与平方运算的次数,是影响算法实现速度的主要因素。

1.2 基于乘同余对称特性的 SMM 算法

SMM(symmetry of modulo multiplication)算法[14]的原理是基于平方剩余和乘同余的对称性。在求 $x^c \bmod n$ 的迭代运算中,实际只包含两种基本运算 $A_i^2 \bmod n$ 和 $A_i \times x \bmod n, A_i, x \in \{0,1,\cdots,(n-1)/2,\cdots,n-1\}$,由平方剩余和乘同余的对称性,有

$$(n-A_i)^2 \equiv A_i^2 (\bmod n)$$

$$(n-A_i)\times(n-x) \equiv A_i \times x (\bmod n)$$

$$A_i \times (n-x) \equiv (n-A_i)\times x \equiv -A_i \times x (\bmod n)$$

在求 $x^c \bmod n$ 的每步迭代计算中进行有条件代换,根据对称特性,如果 $A_i, x > (n-1)/2$,则用 $(A_i - n)$ 或 $(x-n)$ 代替 A_i,或 x 进行乘同余或平方剩余计算,其计算结果不变。但由于每次运算时减小了部分被乘数和乘数的绝对值,从而也就减少了求模运算量和乘法运算时间,使整个模幂算法的运算速度得到提高,即可构成一种快速算法。

2 模幂组合算法——固定基窗口方法的实现

2.1 模幂运算的固定基窗口方法的实现原理

令 $(k_{d-1}, k_{d-2}, \cdots, k_1, k_0)_{2^\omega}$ 是 k 的以 2^ω 为基的表达式,其中 $d = \lfloor t/\omega \rfloor$,$t$ 是 k 的二进制位长。再令 $Q_j = g^{\sum_{i;k_i=j} 2^{\omega i}} = \prod_{i;k_i=j} g^{2^{\omega i}}$,对于每个 $1 \leqslant j \leqslant 2^\omega - 1$,则

$$g^k (\bmod n) = g^{\sum_{i=0}^{d-1} k_i 2^{\omega i}} = \prod_{j=1}^{2^\omega-1} (g^{\sum_{i;k_i=j} 2^{\omega i}})^j =$$

$$\prod_{j=1}^{2^\omega-1} Q_j^j = Q_{2^\omega-1} \times (Q_{2^\omega-1} \times Q_{2^\omega-2}) \cdots$$

$$(Q_{2^\omega-1} \times Q_{2^\omega-2} \cdots \times Q_1) (\bmod n)$$

固定基窗口方法的实现就是基于这一推证。

2.2 模幂运算的固定基窗口方法的算法实现

本节采用 SMM 算法结合模幂运算的固定基窗口方法的实现原理构成模幂运算的固定基窗口方法,使得算法整体效率得到提高,且算法简洁。

下面描述这种算法,算法分两步进行。

算法 1 固定基窗口方法

input:一个正整数 k,窗口宽度 ω,g,$d = \lceil t/\omega \rceil$,模 n。

output:$g^k \bmod n$。

a)预计算 $g_i = g^{2^{\omega i}}$,若 $g_i \geqslant (n-1)/2$,则 $g_i \leftarrow g_i - n, 0 \leqslant i \leqslant d-1$。

b)利用 SMM 算法与结合模幂运算的固定基窗口方法的实现原理进行模幂运算。

(a)$A \leftarrow 1$,$B \leftarrow 1$。

(b)对于 j 从 $2^\omega - 1$ 到 1,重复执行:

①对于每一个 i,若 $k_i = j$,$B \geqslant (n-1)/2$,则 $B \leftarrow B - n$;$B \leftarrow B \times g_i (\bmod n)$。

②若 $A \geqslant (n-1)/2$,则 $A \leftarrow A - n$;若 $B \geqslant (n-1)/2$,则 $B \leftarrow B - n$;$A \leftarrow A \times B (\bmod n)$。

(c)返回 A。

2.3 与平方乘算法的比较

一个 t bit 整数的二进制表示中含有 $t/2$ 个零,平方乘算法需要 t 次平方运算,需要 $t/2$ 次乘运算;模幂运算的固定基窗口方法需要的运算次数分析如下:忽略掉做减法运算所消耗的时间,在算法 1 的 b)(b)的①步中,对于每一个 $0 \leqslant i \leqslant d-1$,$k_i = j$ 的概率为 $1/(2^\omega - 1)$,这一步乘模运算的平均次数为 $d/(2^\omega - 1)$,所以在算法 b)中所做的模乘运算的次数 $d/(2^\omega - 1)$,整个算法的模乘运算次数为 $(d/(2^\omega - 1) + 1) \times (2^\omega - 1) = d + (2^\omega - 1)$,所以两种算法花费的时间比率近似为

$$\frac{t + \frac{t}{2}}{d + (2^\omega - 1)} = \frac{\frac{3t}{2}}{\lceil t/\omega \rceil + (2^\omega - 1)}$$

所以运算速度大致提高为

$$\frac{\frac{3t}{2} - \lceil t/\omega \rceil - (2^\omega - 1)}{\frac{3t}{2}} \approx 1 - \frac{2}{3\omega} - \frac{2(2^\omega - 1)}{3t} \tag{1}$$

由式(1)可知固定 ω,即固定基 2^ω 的情形下,t 越大,所需的预存储空间越大,运算速度提高越大。当进行大整数的模幂运算时,若窗口宽度 ω 取得较小时,指数 k 的二进制长度 $t >> 2^\omega - 1$,则式(1)中的 $\frac{2(2^\omega - 1)}{3t}$ 项比较小,可以忽略掉,此时固定基窗口方法相对于二进制平方乘算法的运算速度大致提高 $1 - \frac{2}{3\omega}$。此时若 ω 在一个较小的值范围内取值时,ω 越大,运算速度提高越快;但是当 ω 取得较大时,会导致 $\frac{2^\omega - 1}{t}$ 的值大于 1,式(1)中的 $\frac{2(2^\omega - 1)}{3t}$ 的项不能忽略掉,此时固定基窗口方法的运算速度将会大大降低。

本文实验环境为 Intel Core2 T5670 1.80 GHz CPU、2 GB 内存的 PC 机,实现平台为 Windows XP Wolfram Mathematica 8.0,取指数 k 分别为 $k_1 = 10^{100} \times 2^{100}$,$k_2 = 10^{100} \times 2^{200}$,$k_3 = 10^{100} \times 2^{300}$,$k_4 = 10^{100} \times 2^{400}$,$k_5 = 10^{100} \times 2^{500}$,用 l_i 表示 k_i 的 2^ω 进制长,$i = 1,2,3,4,5$,固定 $g = 2^{100} - 1$,取模 $n = 2^{10000} - 1$,分别采用窗口宽度 $\omega = 1,2,3,4,5,6$,分别采用固定基窗口方法与二进制平方乘算法计算 $g^k (\bmod n)$,程序重复运行 1 000 次,得到如表 1~4 所示的实验数据。

从表 1 可以看出,在固定窗口宽度 ω 的情形下,对应的指数 k 越大,其 2^ω 进制长越长;在同一指数 k 下,ω 越大,指数 k 的 2^ω 进制长越短。从表 2 可以看出,在固定窗口宽度 ω 的情形下,对应的指数 k 越大,固定基窗口方法所需的运算时间越多;在同一指数 k 下,若 ω 取较小的值 $\omega = 2,3,4$ 时,程序运行的速度较快;但是当 ω 取得较大时,会导致算法 1 的 b)中 j 的循环次数 $2^\omega - 1$ 增多,算法的运行速度变慢。从表 4 可以看出,在固定窗口宽度 ω 的情形下,对应的指数 k 越大,固定基窗口方法相对于二进制平方乘算法的运算速度提高程度越大;在同一指数 k 下,若 ω 取较小的值 $\omega = 2,3,4$ 时,固定基窗口方法相对于二进制平方乘算法的运算速度提高越大;但是当 ω 取得较大时,会导致 $\frac{(2^\omega - 1)}{t}$ 的值大于 1,式(1)中的 $\frac{2(2^\omega - 1)}{3t}$ 的项不能忽略掉,此时固定基窗口方法的运算速度将会大大降低。这些实验数据证实了本文对式(1)的分析,所以在选取

窗口宽度时,应根据所求的模幂的指数 k 的大小,适当选取窗口宽度 ω,以保证 $t>>2^{\omega}-1$。在进行大整数的模幂运算时,笔者推荐选取窗口宽度 $\omega=2,3,4$。

表 1　指数 k_i 在不同的 ω 下的 2^{ω} 进制长

ω	l_1	l_2	l_3	l_4	l_5
1	433	533	633	733	833
2	217	267	317	367	417
3	145	178	211	245	278
4	109	134	159	184	209
5	87	107	127	147	167
6	73	89	106	123	139

表 2　固定基窗口方法在不同的 k、ω 下的运行时间　/s

ω	k_1	k_2	k_3	k_4	k_5
1	26.234 4	26.687 5	27.203 1	27.718 8	28.25
2	21.796 9	22.546 9	23.328 1	24.156 3	24.968 8
3	21.812 5	23.937 5	24.390 6	25.421 9	27.609 4
4	24.125	26.109 4	28.078 1	29.984 4	31.953 1
5	31.546 9	34.812 5	38.00	41.203 1	44.531 3
6	46.50	52.00	57.703 1	63.109 4	68.671 9

表 3　平方乘算法在不同的 k 下的运行时间　/s

k_1	k_2	k_3	k_4	k_5
88.953 1	110.828	130.344	152.813	171.516

表 4　固定基窗口方法相对于平方乘算法的提高程度/%

ω	k_1	k_2	k_3	k_4	k_5
1	70.507 6	75.919 9	79.129 7	81.860 9	83.529 2
2	75.496 2	79.656	82.102 6	84.192 2	85.442 3
3	75.478 7	78.401 2	81.287 5	83.364	83.902 7
4	72.879	76.441 6	78.458 4	80.378 3	81.370 1
5	64.535 4	68.588 7	70.846 3	73.036 8	74.036 6
6	47.725 3	53.080 5	55.73	58.701 4	59.961 7

3　结束语

模幂运算是公钥密码体制中一个很重要的运算,其运算速度直接影响到公钥密码体制能否有效地实现。本文与其他解决 a)类模幂问题的算法类似,也采用了预处理技术,将椭圆曲线的定点标量乘的固定基窗口方法应用在模幂运算中,与 SMM 算法进行组合,得到一种新的关于模幂运算的固定基窗口方法,并以牺牲存储空间为代价,获得了较快的运算速度。预存储空间中存放的是固定底数 g 的模幂,当固定窗口宽度 ω,存储空间大小与指数 k 的二进制长度 t 有关,t 越大,所需的预存储空间越大,相对于二进制的平方乘算法,运算速度提高越大。为保证较高的运算速度,窗口宽度 ω 不能选得太大,笔者推荐选取窗口宽度 $\omega=2,3,4$。

参考文献:

[1] RIVEST R,SHAMIR A,ALDEMAN L. A methoed for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM,1978,21(2):120-126.

[2] GORDON D M. A survey of fast exponenfiafion methods[J]. Journal of Algorithms,1998,27(1):129-146.

[3] MENEZES A J,Van OORSCHOT P C,VANSTONE S A. Handbook of applied cryptography[M]. Boca Raton:CRC Press,1996.

[4] BRICKELL E F, GORDON D M, McCURLEY K S,*et al*. Fast exponentiation with precomputation[C]//Proc of Eurocrypt on Advances in Cryptology. New York:Springer-Verlag,1993:200-207.

[5] De ROOIJ P. Efficient exponentiation using precomputation and vector addition chains[C]//Proc of Eurocrypt on Advances in Cryptology. New York:Springer-Verlag,1995:389-399.

[6] DIMITROV V S, COOKLEV T V. Two algorithms for modular exponentiation using nonstandard arithmetics[J]. IEICE Trans on Fundamentals,1995,E78-A(1):82-87.

[7] DIMITROV V S,JULLIEN G A, MILLER W C. An algorithm for modular exponentiation[J]. Information Processing Letters,1998,66(3):155-159.

[8] GATHEN J,NOCKER M. Exponentiation using addition chains for finite fields[EB/OL]. (2008-3-18). http://citeseer. nj. nec. com/vonzurgathe-n00exponentiation. html.

[9] KNUTH D E. The art of computer programming[M]. 2nd ed. [S. l.]: Addison-Wesley,1981.

[10] KOC C K. Analysis of sliding window techniques for exponentiation [J]. Computers & Mathematics with Applications, 1995, 30(10):17-24.

[11] CHIOU C W. Parallel implementation of the RSA public-key cryptosystem [J]. International Journal of Computer Mathematics, 1993,48(3):153-155.

[12] LOU D C, CHANG C C. An adaptive exponentiation method [J]. Journal of Systems and Software,1998,42(1):59-69.

[13] HANKERSON D, MENEZES A, VANSTONE S. Guide to elliptic curve cryptography[M]. New York:Springer-Verlag,2004:104.

[14] CHEN Y. A new fast RSA algorithm[J]. Journal of University of Electronic Science and Technology of China,1996,24(2):223-228.

(上接第 675 页)

参考文献:

[1] 王先发,禹化龙,张碧雄. 我国未来卫星导航信号的优先选择——BOC 调制信号[J]. 中国电子科学研究院学报,2009,4(3):307-312.

[2] LOHAN E S. Statistical analysis of BPSK-like techniques for the acquisition of Galileo signals[J]. AIAA Journal of Aerospace Computing Information and Communication,2006,3(5):234-243.

[3] JULIEN O, MACABIAU C, CANNON M E, *et al*. ASPeCT: unambiguous sine-BOC(n,n) acquisition/tracking technique for navigation applications[J]. IEEE Trans on Aerospace and Electronic Systems,2007,43(1):150-162.

[4] NUNES F D, SOUSA F M G, LEITAO J M N. Multipath mitigation technique for BOC signals using gating functions[C]//Proc of the 2nd European Space Agency Workshop on Satellite Navigation User Equipment Technologies. 2004.

[5] 邢兆栋,张其善,杨东凯. GALILEO 接收机中 BOC(1,1)信号的捕获[J]. 北京航空航天大学学报,2006,32(6):687-690.

[6] 陈佳品,齐佳敏,陈翔,等. 二进制偏置载波信号精确同步装置及其同步方法:中国,201110108183[P]. 2011-09-14.

[7] 刘哲,倪少杰,牟卫华,等. CCRW 技术在 MBOC 调制信号下多径抑制性能[J]. 全球定位系统,2011,36(4):29-33.

[8] 杨力,潘成胜,冯永新,等. 一种 BOC 调制信号的同步主峰检测新算法[J]. 宇航学报,2010,31(8):2008-2014.