大数幂剩余的二进制冗余数 Montgomery 算法*

陈 运** 龚耀寰

(电子科技大学通信与信息工程学院，电子工程学院 成都 610054)

【摘要】 介绍了大数幂剩余的 Montgomery 算法，提出了基于二进制冗余数的大数幂剩余 Montgomery 算法模型。理论分析表明，采用二进制冗余数可减少乘法的进位传播，同时使算法的迭代步数减少17.2%。进一步提高了大数幂剩余的运算速度。

关 键 词 密码学； 公钥密码体制； 幂剩余； Montgomery 算法； 二进制冗余数

中图分类号 TN911.2; TN911.7

目前，许多网络黑客事件在新闻媒体上频频曝光，使人们对网络信息安全产生了极大的担心，解决网络通信和电子商务系统的信息安全问题成为当务之急。

身份认证、数字签名和密钥分配是网络通信尤其是电子商务系统中信息安全必不可少的内容，解决这些问题的一体化方法是采用公钥密码体制。

大数幂剩余运算是许多国际流行公钥密码体制的核心运算，由于数值常常大到十进制的100~200位，运算速度非常缓慢。目前，大数幂剩余快速算法很多，在硬件实现的算法中，以 Montgomery 算法映射的各种方案速度较快[1]。本文提出了一种新的基于二进制冗余数的 Montgomery 算法，可进一步提高大数幂剩余的运算速度。

1 Montgomery 算法简述

大数幂剩余运算一般用 BR(Binary Representations)算法将乘幂求模运算分解成一系列乘同余和平方剩余的迭代，即 n 位二进制指数

$$E=e_{n-1}e_{n-2}\cdots e_1e_0 \qquad e_i\in\{0,1\},\ i=0,1,\cdots,n-1$$

$$\begin{aligned}Y\equiv(((\cdots((((X^{e_{n-1}} \bmod M)^2 \bmod M)X^{e_{n-2}} \bmod M)^2 \bmod M)\cdots\\ X^{e_1} \bmod M)^2 \bmod M)X^{e_0} \bmod M)\end{aligned} \tag{1}$$

式(1)的每步迭代实质都是模乘，包括乘法和求模两种基本运算。

$$\begin{cases}X=x_{n-1}x_{n-2}\cdots x_1x_0\\ Y=y_{n-1}y_{n-2}\cdots y_1y_0 \qquad x_i、\ y_i、\ m_i\in\{0,1\},\ i=0,1,\cdots,n-1,m_0=1\\ M=m_{n-1}m_{n-2}\cdots m_1m_0\end{cases}$$

式中 X、Y、M 分别为 n 位二进制被乘数、乘数和奇数模，且 $X<M$, $Y<M$。

Montgomery 模乘算法采用模加右移的方法避免了求模运算中费时的除法。基-2的 Montgomery 算法 $MM(X,Y,M)=XY2^{-(n+2)} \bmod M$ 步骤如下：

1) $y_0=0$；

2) 对 $i=0\sim n$，计算

$$\begin{aligned}q_i&=y_i \bmod 2\\ s_{i+1}&=(s_i+q_iM)\mathrm{div}2+y_iX\end{aligned} \tag{2}$$

3) 输出 $S=s_{n+1}$，其中 mod 表示求余数；div 表示求商。

1999年5月16日收稿
* 电子部预研基金资助项目
** 女 42岁 硕士 副教授

用 Montgomery 算法计算式(1)的每步迭代，则可以构造从左到右的二进制幂剩余算法（基于Montgomery算法的BR算法），其步骤如下：

1) 预计算 $\widetilde{X} \equiv X2^{n+2} \bmod M$；

2) $Y=\widetilde{X}$, $i=n-1$。若$e_i=0$, $i=i-1$(找到 E 中第一个1)；

3) 计算$Y=MM(Y,Y,M)$(自乘得到$Y \equiv Y^2 2^{-(n+2)} \bmod M$)；

4) 若$e_i=1$, 计算$Y=MM(Y,\widetilde{X},M)$(乘 X 得到$Y \equiv Y\widetilde{X}2^{-(n+2)} \bmod M$)；

5) $i=i-1$，若 $i \geqslant 0$，返回到步骤3)(得到$Y \equiv X^E 2^{n+2} \bmod M$)；

6) 计算$Y=MM(1,Y,M)$，得到$Y \equiv X^E \bmod M$。

后处理：若 $Y \geqslant M$, 输出$Y=Y-M$。

快速算法分为以下三个部分：1) 预计算 $\widetilde{X}=X2^{n+2} \bmod M$；2) 幂剩余运算，由 BR 迭代算法完成，每步迭代运用 Montgomery 算法；3) 后处理，使最终结果 $Y<M$。

2 二进制冗余数及其对模乘算法的影响

用 $\bar{1}$ 表示-1，二进制序列中的连1可表示成如下形式

$$\underbrace{11\cdots1}_{n个}=1\underbrace{0\cdots0}_{(n-1)个}\bar{1} \tag{3}$$

式(3)右边用 $\{1,0,\bar{1}\}$ 表示的二进制数即为二进制冗余数。当连1个数大于或等于3时，二进制冗余数的汉明重量——非零位个数小于二进制数。

在式(2)的模加右移计算中，当 s_i、M 和 X 的连1个数较多时，加法常常出现进位传播，影响运算速度。

如果用二进制冗余数表示被乘数，其非零位将减少，运算时产生的进位也随之减少。在同一个位置上出现两个极性相反的非零位1和 $\bar{1}$ 时，相加结果为0，也不产生进位，从而减少进位传播，使 Montgomery 算法运算速度进一步提高。

在式(1)的 BR 算法中，迭代的步数 L 取决于指数 E 的二进制位数 n 和汉明重量 $h(E)$[2]

$$L=n+h(E)-2 \tag{4}$$

将指数 E 变换成二进制冗余数，通过降低指数的汉明重量减少 BR 算法的迭代步数，可进一步提高大数幂剩余的运算速度。

3 基于二进制冗余数的大数幂剩余 Montgomery 算法

二进制冗余数 $R=r_{n-1}r_{n-2}\cdots r_1r_0$，$r_i \in \{1,0,\bar{1}\}$，$i=0,\cdots,n-1$。用 R 代替 E，可将式(1)变为

$$\begin{aligned} Y \equiv X^R \bmod M \equiv (((\cdots((((X^{r_{n-1}} \bmod M)^2 \bmod M)X^{r_{n-2}} \bmod M)^2 \bmod M)\cdots \\ X^{r_1} \bmod M)^2 \bmod M)X^{r_0} \bmod M) \end{aligned} \tag{5}$$

当迭代中出现 $r_i=\bar{1}$ 时，操作可分两步进行：1) 对中间结果求平方剩余；2) 计算中间结果与 X^{-1} 的乘同余。其中 X^{-1} 表示 X 对模 M 的乘逆，即

$$XX^{-1} \equiv 1 \bmod M \tag{6}$$

只要

$$(X,M)=1 \tag{7}$$

则 X^{-1} 必存在，而几乎所有的 $X \in \{0,1,\cdots,M-1\}$ 都满足式(7)[3]。

用式(5)构造一种新的大数幂剩余 Montgomery 算法，其步骤如下：

1) 用欧几里德算法计算 X^{-1}[4]；

2) 将加密指数 E 变换成二进制冗余数 R。变换原则是从 E 的最高位开始，当 E 的二进制序列中有 k 个($k \geqslant 3$)连1时，用 $1\underbrace{0\cdots0}_{(k-1)个}\bar{1}$ 将这 k 个连1替换掉；同理将 X、X^{-1} 也变换成二进制冗余数 X_T 和 X_T^{-1}；

3) 预计算 $\widetilde{X}_T \equiv X_T 2^{(n+2)} \bmod M$，$\widetilde{X}_T^{-1} \equiv X_T^{-1} 2^{(n+2)} \bmod M$；

4) $Y_T = \widetilde{X}_T$，$i = n-1$，若$r_i = 0$，$i = i-1$(找到 R 中第一个1)；

5) $Y_T = MM(Y_T, Y_T, M)$(自乘得到$Y_T \equiv Y_T^2 2^{-(n+2)} \bmod M$)。

6) 若$r_i = 1$，计算$Y_T = MM(Y_T, \widetilde{X}_T, M)$(乘 X_T 得到$Y_T \equiv Y_T \widetilde{X}_T 2^{-(n+2)} \bmod M$)；

7) 若$r_i = \overline{1}$，计算$Y_T = MM(Y_T, \widetilde{X}_T^{-1}, M)$(乘 X_T^{-1} 得到$Y_T \equiv Y_T \widetilde{X}_T^{-1} 2^{-(n+2)} \bmod M$)；

8) 若 $i = i-1$，$i \geqslant 0$，返回步骤5)(得到$Y_T \equiv X_T^R 2^{n+2} \bmod M$)；

9) 计算$Y_T = MM(1, Y_T, M)$(得到$Y_T \equiv X_T^R \bmod M$)；

10) 将二进制冗余数表示的Y_T反变换为二进制表示的Y。

后处理：若$Y \geqslant M$，输出$Y = Y - M$。

式(5)的迭代步数由指数二进制冗余数的位数 n 和汉明重量 $h(R)$决定。迭代时从指数的左边开始，每遇到一个为“0”的位，计算一次平方剩余(迭代一步)；若遇到非零位，则计算一次平方剩余和一次乘同余(迭代两步)。由于式(5)迭代算法的初始值为“1”，而指数最左边的一位总是非零位“1”。因此，最开始的两步迭代分别是$1^2 \pmod M$和$1X \pmod M$，其结果为 X 本身，可以省掉，所以式(5)的迭代步数为

$$L' = n + h(R) - 2 \tag{8}$$

例如计算$32^{31} \pmod{35}$，首先用欧几里德算法计算出$32^{-1} \pmod{35} = 23$[4]，然后分别用基-2的 Montgomery 算法和新算法进行计算，结果如表1所示，其中每步迭代均用 Montgomery 算法实现。

表1　基-2 Montgomery 算法与新算法迭代步数比较

指　数		基-2 Montgomery 算法	新算法
		11111	$10000\,\overline{1}$
迭代步数及结果	1	$32^2 \pmod{35} = 9$	$32^2 \pmod{35} = 9$
	2	$9 \times 32 \pmod{35} = 8$	$9^2 \pmod{35} = 11$
	3	$8^2 \pmod{35} = 29$	$11^2 \pmod{35} = 16$
	4	$29 \times 32 \pmod{35} = 18$	$16^2 \pmod{35} = 11$
	5	$18^2 \pmod{35} = 9$	$11^2 \pmod{35} = 16$
	6	$9 \times 32 \pmod{35} = 8$	$16 \times 23 \pmod{35} = 18$
	7	$8^2 \pmod{35} = 29$	
	8	$29 \times 32 \pmod{35} = 18$	

由表1可见，两种算法的运算结果相同，而新算法的迭代步数减少了25%。

如果指数 E 的长度达到512 bit，假定0、1个数各占1/2，迭代步数将达到700多步。用 R 代替 E，必然有

$$L' \leqslant L \tag{9}$$

新算法用二进制冗余数进行模乘运算，使 Montgomery 算法的进位传播减少，提高了模乘运算速度。为了避免额外的时间开销，冗余数变换仅在幂剩余算法开始之前进行一次。

4　新算法的速度分析

新算法的速度改善体现在以下两个方面：1) 迭代步数的减少；2) Montgomery 模乘算法中进位传播的减少。RSA(Rivest, Shamir & Adleman)公钥密码体制的核心运算即是大数幂剩余运算。文献[4]提出的 RSA 快速算法是同时利用乘同余对称特性快速算法和二进制冗余数得到的。根据其理

论分析，二进制冗余数带来的速度改善是17.2%(迭代步数减少了17.2%)。在此基础上减少Montgomery算法的进位传播，将使运算速度进一步提高。

用Montgomery算法映射、并由脉动阵列实现的512 bit RSA算法，在125 MHz时钟下工作速率可达164 kbps[5]。如果再采用二进制冗余数，在理论上，其工作速率将达到192 kbps以上。

5 结束语

在用硬件实现的大数幂剩余快速算法中，以Montgomery算法映射而来的脉动阵列实现最为快速。采用本文提出的新算法后，能够进一步提高17.2%以上的运算速度。但新算法事先计算模数乘法逆需要一定的时间，故只适于迭代步数很多的大数幂剩余算法。

参考文献

1 Montgomery P L. Modular multiplication without trial division. Math Comp, 1985,44(170):11~15

2 陈 运. 一种新的快速RSA算法. 电子科技大学学报, 1995, 24(增): 223~228

3 Chen Yun. An improved SMM algorithm. Journal of Electronic Science, 1999,16(1): 93~96

4 陈 运. 信息加密原理. 成都: 电子科技大学出版社, 1996

5 Yang Chingchao, Jen Cheinwei, Chang Tiansheuan. The IC design of a high speed RSA processor. APCCAS'96. IEEE Asia Pacific Conference on Circuits and Systems'96 Proceedings, New York, USA,1996:33~36

A New Montgomery Algorithm Based on Binary Redundant Representations for Modular Exponentiation with Very Large Operand

Chen Yun Gong Yaohuan

(Telecommunication Institute, Electronic Engineering Institute, UEST of China Chengdu 610054)

Abstract The Montgomery algorithm used in modular exponentiation with very large operand is briefly introduced. A new Montgomery algorithm is presented for modular exponentiation computation with very large operand based on binary redundant representations. Computation of modular exponentiation with very large operand is furthermore speeded up by decreasing recursive steps and carry propagation in modular multiplication.It is shown by theoretical analysis that recursive stops are also decreased by 17.2% on average.

Key words cryptography; public-key cryptosystem; modulo exponentiation; Montgomery algorithm; binary redundant number