基于 Karatsuba 递归思想的 Montgomery 模乘算法 *

羊红光[1,3]，黄世中[1,2]，张利民[4]
(1. 河北省科学院应用数学研究所，河北 石家庄 050081；
2. 河北省数字研究中心，河北 石家庄 050016；
3. 石家庄冀科双实科技公司，河北 石家庄 050081；
4. 衡水学院数学与计算机学院，河北 衡水 053000)

摘 要：采用大数的高基表示方法和 Karatsuba 递归思想改进了 Montgomery 模乘中的 IFIOS 实现算法，该算法可以应用于 RSA 公钥体制下的模乘法器的设计。模乘运算的速度决定了公钥加密系统和众多通信系统的系统性能，通过与 IFIOS 算法的比较分析发现，改进后的算法具有使用的乘法次数少、并行性能高等优点，是一种适合设计硬件的高效算法。此算法也适用于其他公钥体制的加解密处理器。

关键词：Montgomery 模乘；Karatsuba 递归思想；IFIOS
中图分类号：TP301.6 **文献标识码**：A **文章编号**：1674-7720(2010)16-0021-03

A novel Montgomery modular multiplication algorithm based on Karatsuba recursive idea

YANG Hong Guang[1,3], HUANG Shi Zhong[1,2], ZHANG Li Min[4]
(1. Institute of Applied Mathematics, Hebei Academy of Sciences, Shijiazhuang 050081, China;
2. Mathematics Research Center, Hebei Province, Shijiazhuang 050016, China;
3. SJZ JKSS Technology Co.,Ltd, Shijiazhuang 050081, China;
4. Department of Mathematics and Computer Science, Hengshui University, Hengshui 053000, China)

Abstract: A high-radix technique and Karatsuba recursive idea is employed to improve the IFIOS-Montgomery's algorithm, and the improved algorithm is adapted to design of modular multiplier for RSA cryptosystem. Modular multiplication operation is a key factor of determining performance system of publickey cryptography systems and most of communication systems. Compared with IFIOS algorithm, the improved algorithm require less multiplications and has a higher parallel performance. So the improved algorithm is an efficient algorithm suitable for hardware design and adapted to several public-key system encryption/decryption processors.

Key words: Montgomery modular multiplication; Karatsuba recursive idea; IFIOS

基于大整数分解的 RSA 体制是公钥密码体制中的一类，是目前应用最广泛的公钥密码体制之一。对于 RSA 来说，人们关心的一个重要问题是大整数模幂运算。为此，许多学者一直在对此进行研究[1-2]。加快模幂运算主要考虑减少模幂的次数和提高大整数模乘的运算速度。1985 年，MONTGOMERY P.L.提出的大整数模乘算法是一种非常有效的方法。之后 KOC C.C. 对几种 Montgomery 算法的实现方法进行分析比较，发现 FIOS 算法是一种空间复杂度低的实现方法[3]。其他学者也对 Montgomery 算法进行了各式各样的改进，取得了一些进步[4-5]。基于 Karatsuba 递归思想的 Karatsuba 快速乘法算法已经被证实在计算机系统中比传统算法更具优势[6]。本文运用 Karatsuba 递归思想对 Montgomery 模乘算法进行了改进，给出了一种改进算法描述，对其复杂度和并行度做了分析比较。

1 Montgomery 算法

1.1 Montgomery 算法简介

Montgomery 构造了一个特殊的剩余系，借此把普通的模乘转换成该剩余系中的模乘，把对 n 的模简化变为对一个任意 $r(r>n)$ 的模简化，当选取为 2 的幂时，对 r 模简化只需简单的移位操作即可，这样在计算 $a\cdot b \pmod n$ 时避免了除法运算，从而实现了在计算机中快速计算

* 基金项目：河北省自然科学基金数学研究专项(08M009)；河北省科学院重大科技攻关项目(D2009614)

模乘的目的。

Montgomery 算法主要是计算 $a\cdot b\cdot r^{-1}\bmod n$，设 n 为 k 比特的整数，$2^{k-1}<n<2^k$，$a<n$，$b<n$；$(n,r)=1$，通常取 $r=2^k$；r^{-1} 是 r 模 n 的逆，即 $r\cdot r^{-1}=-1(\bmod n)$；$r\cdot r^{-1}-n\cdot n'=1$。Montgomery 算法描述为：

function Monpro (a,b,r,n)

(1) $t:=a\cdot b$

(2) $u:=(t+(t\cdot n'\bmod r)n)/r$

(3) if $u\geqslant n$ then

return $u-n$

else return u

因为 r 为 2 的幂，所以算法在计算机中主要使用乘法、加法和移位操作，不用除法，因而能够快速实现。Montgomery 算法共需要 $3s^2$ 次乘法、$6s^2+2s+2$ 次加法和 $3s+3$ 个存储空间。由于 Montgomery 算法计算的是 $a\cdot b\cdot r^{-1}\bmod n$ 的值，因此还需要进行相应的预计算和后处理来去除 r^{-1} 的影响。

1.2 Karatsuba 递归算法

计算环 R 上的次数小于 n 的两个多项式 $f(x),g(x)\in R[x]$ 的乘积，如果 f_i、g_i、$h_k\in R$，分别是 f、g 和 $h=fg$ 的系数，经典算法将进行 $O(n^2)$ 次运算，在环上 f_i 和 g_i 计算 h_k，对所有 $h_k=\sum_{i+j=k}f_ig_j$ 有 n^2 个乘法 f_ig_j 加上 $(n-1)^2$ 个加法。1962 年，Karatsuba 提出的算法降低了多项式的乘法个数，增加了加法个数。由于乘法比加法慢，当 n 足够大时，就可以节省很多消耗。Karatsuba 乘法可表示为这样的形式：$fg=F_1G_1x^n+((F_0+F_1)(G_0+G_1)-F_0G_0-F_1G_1)x^m+F_0G_0$，其中 $f=F_1x^{n/2}+F_0$，$g=G_1x^{n/2}+G_0$，F_0、F_1、G_0、$G_1\in R[x]$，且次数小于 $n/2$。经计算可知 Karatsuba 算法比次数小于 $2n$ 的多项式乘积多做 $9n^{\log 3}$ 或 $O(n^{1.59})$ 次环运算。Karatsuba 算法也适用于 r 进制的整数乘积运算。

由于 Karatsuba 算法是递归的，在计算中需要存储中间状态、临时变量等比较多。所以本文局部运用这种递归思想，并采用高基来改进算法，避免了这些问题的出现。

1.3 IFIOS 算法

FIOS 算法是 Montgomery 实现算法中的一种，需要的加法次数和读写次数稍多，但空间复杂度最低，因此适用于设计高速专用 RSA 密码芯片件。参考文献[5]给出了 FIOS 的改进算法——IFIOS 算法，算法的性能进一步得到优化。

IFIOS 算法共需要 $2s^2+s$ 次乘法、$4s^2+4s+2$ 次加法、$5s^2+6s+2$ 次读、s^2+5s+1 次写和 $s+4$ 个存储单元，是一个较好的方法。

2 利用 Karatsuba 思想改进 Montgomery 模乘算法

2.1 Karatsuba- IFIOS 模乘算法

为了进一步提高实现的效率，利用了 Karatsuba 思想对 IFOS 算法进行了改进，提出了 Karatsuba- IFIOS 模乘算法。首先设算法中的大整数采用基 2^{2w} 表示，存储空间以 $2w$ 比特的存储单元进行存储。设 n 为 k 比特的整数，r 表示为 $r=2^k=2^{sw}$。所以 a,b,n 可分别表示为：

$a=a[s/2-1]\cdot 2^{2w(s/2-1)}+\cdots+a[0]\cdot 2^0$，

$b=b[s/2-1]\cdot 2^{2w(s/2-1)}+\cdots+b[0]\cdot 2^0$，

$n=n[s/2-1]\cdot 2^{2w(s/2-1)}+\cdots+n[0]\cdot 2^0$。

$n'[0]=n'\bmod 2^{2w}=-n_0^{-1}\bmod 2^{2w}$，通过扩展的 Euclidean 算法计算。$C$ 表示 $2w$ 位的进位寄存器，S 表示 $2w$ 位的求和寄存器，m 是 $2w$ 位的临时存储单元，$(t[s/2+1],t[s/2],\cdots,t[0])$ 是 $2w$ 位的存储单元。因为 w 是乘法器的长度，所以 w 一般取值为 32、16、8 等。为了保证 RSA 的安全性，模 n 的长度 k 常取 768、1 024、2 048。由 $k=sw$，可以得出 $s/2$ 是整除的。

Karatsuba- IFIOS 算法描述为：

```
for  i=0 to s/2-1
  {(R0,S):=t[0]+Karatsuba(a[0],b[i],0);
  m:=Karatsuba(S,n[0],1)mod 2^w;
  (R1,S):=S+Karatsuba(m,n'[0],0);
  for j=1 to s/2-1
  {(R0,S):=t[j]+Karatsuba(a[j],b[i],0)+R0;
  (R1,S):=S+Karatsuba(m,n[j],0)+R1;
  t[j-1]:=S;}
  t[s/2-1]:=S
  (C,S):=R0+R1;
  t[s/2+1]:=t[s/2+1]+C;
  (C,S):=t[s/2]+S;
  t[s/2-1]:=S;
  t[s/2]:=t[s/2+1]+C;
  t[s/2+1]:=0;
  }
  if (t≥n) t:=t-n
       return t
  fuction Karatsuba(x,y,type)
  // x=x[1]·2^w+x[0], y=y[1]·2^w+y[0]
  temp1=x[1]×y[1];
  temp2=x[0]×y[0];
  temp3=x[1]×y[0]-x[0]×y[1];
if type=0 then
     return (2^2w temp1+2^w temp3+temp2)
     else return (2^w temp3+temp2)
```

由于 $m:=t[0]\times n'[0]\bmod 2^{2w}$ 带有 $\bmod 2^{2w}$ 的运算，因此 Karatsuba(x,y,type) 只返回 $(2^w$ temp3+temp2$)$ 即可 $(\text{temp}=1$ 的情况$)$。

2.2 性能分析及对比

首先分析 Karatsuba-IFIOS 算法的时间复杂度与空间占用资源分析。对于该算法的乘法数，因为 $t[j]$、m 等都是 $2w$ 位的变量，通过调用 Karatsuba(x,y,type) 函数计算 $2w$ 位的乘法，实际的乘法还是 w 位的标准乘法。Karatsuba(x,y,type) 算法中还包含 3 次 w 位的标准乘法，然后是两层循环，可以算出 Karatsuba-IFIOS 算法共需要 $\frac{3}{2}s(s+1)$ 次 w 位的标准乘法。

对于加法的次数，由于减法运算可以通过补码转换

为加法运算，所以可把减法看做加法来计算。Karatsuba-IFIOS 算法中均为 $2w$ 位的加法，这样改算法共需要 $s(3s+9)+2$ 次 $2w$ 位加法和 $s(s+3)$ 次 w 位加法。

最后计算临时存储单元的数目。只考虑如下一些临时存储单元：一部分是 $2w$ 位的存储单元 $t[s/2+1],t[s/2],\cdots,t[0]$，一个 $2w$ 位的 m 和3 个 $2w$ 位的存储变量(Karatsuba$(x,y,type)$算法中需要的)，这样总数目为$(s/2+6)\times 2=s+12$ 个 w 位的存储单元。与 IFIOS 算法相比，增加了 9 个 w 位的存储单元。需要说明的是，由于初始输入 a,b,n 的存储单元数目不变，只是单元存储由 w 位变为 $2w$ 位，$n'[0]$是在预计算中得到的，还有 C,S,B 均为计算中运算器的寄存器，不占用存储单元，所以这些均不考虑在本算法内。

由于 k 比特乘法运算的时间复杂度是 $O(k^2)$加法运算的复杂度是 $O(k)$。一般可认为 1 次 k 比特的乘法运算需要 k 次 k 比特的加法运算来实现。因此，提高算法的运算速度关键是看其乘法次数的多少。

通过与 IFIOS 算法的比较发现：加法的次数增加了，如果按照 1 次 $2w$ 位加法等于 2 次 w 位加法计算，加法次数比 IFIOS 方法增加 $2s^2+14s$ 次，近似于 $2s+14$ 次乘法；综合看来乘法的次数明显减少，约减少了$(s^2-5s)/2$ 次。存储空间仅增加了 9 个 w 位的存储单元，变化不大。因此改进后的算法还是很有效的。

2.3 并行性分析与比较

算法并行性的优劣，在密码芯片设计中至关重要，它决定了硬件芯片设计的可扩展性：一个具有较好并行性的算法，可以通过增加一定的硬件资源，利用并行和流水线技术，设计出高速的密码芯片。

下面对改进后的 IFIOS 算法进行并行性分析：

(1) IFIOS 算法的并行性

首先观察循环内的运算，算法内层循环中的$(R0,S):=t[j]+a[j]\times b[i]+R0$ 和$(R1,S):=S+m\times n[j]+R1$ 运算可分别并行处理：$t[j]+R0$ 和 $a[j]\times b[i]$以及 $S+R1$ 和 $m\times n[j]$在同一个时钟周期内并行处理，然后在下一个时钟内完成两个计算结果的累加。这样可在两个时钟周期内完成这两步的运算，提高了运算效率。

由于 s 内循环之间的数据是无关的，因此可以在并行的基础上建立两级流水线，当前一个循环$(j=k)$进行$(R0,S):=t[j]+a[j]\times b[i]+R0$ 的最后结果累加时，后一循环$(j=k+1)$计算 $t[j]+R0$ 和 $a[j]\times b[i]$的值。

(2) Karatsuba-IFIOS 算法的并行性

Karatsuba-IFIOS 算法的并行处理主要有 Karatsuba$(x,y,\text{type})t[j]+\text{Kartsuba}(a[j],b[i],0)+R0$ 的计算。由于数据无关，在 Karatsuba$(x,y,type)$乘法算法中 $\text{temp1}=x[1]\cdot y[1]$和 $\text{temp2}=x[0]\cdot y[0]$，还有 $x[1]-x[0]$与 $y[0]-y[1]$可以在一个时钟周期内并行处理；同样 temp1+temp2 的计算和 $x[1]-x[0]$与 $y[0]-y[1]$乘积也可以在一个时钟周期内并行处理。因此，函数 Karatsuba$(x,y,type)$在硬件芯片设计中采用双乘法器和双加法器，只要 3 个时钟周期即可。而 $t[j]+\text{Karatsuba}(a[j],b[i],0)+R0$ 的并行计算类似于 IFIOS 的处理方法，在同一个时钟周期里计算 $t[j]+R0$ 和 Karatsuba$(a[j],b[i],0)$，所以只要 2 个时钟周期。

在改进算法的主循环里有 $t[j]+\text{Karatsuba}(a[j],b[i],0)+R0$ 和 $S+\text{Karatsuba}(m,n[j],0)+R1$ 这样两个类似的计算，在硬件设计中共需要 10 次加法的顺序，如果采用双加法器就能够获得更好的并行处理性能。另外需要注意，w 的取值应根据不同的应用需求和不同的设计结构选取。由于 w 值决定了乘法器和加法器的位数，所以 w 值不能太大，否则会导致芯片的实现面积和时延较大，也会使硬件实现的复杂度增加。

利用 Karatsuba 递归思想成功地将 IFIOS 算法进行了改进，并给出算法的具体描述。通过分析和比较，改进后的 Karatsuba-IFIOS 算法比 IFIOS 算法的乘法次数明显减少，并行处理性能更高，是一种适合硬件设计的好算法。Karatsuba 递归思想也可以改进其他的算法，并能不同程度地减少乘法次数。

参考文献

[1] IPUT H K, ASEP B N, RANDY S P, et al. Very fast pipelined RSA architecture based on montgomery's algorithm[C]. 2009 International Conference on Elec- trical Engineering and Informatics. Bangi, Malaysia IEEE Computer Socitey Press, 2009:491-495.

[2] FANG X Y, ZHANG J H. The researcher and implement of high-speed modular multiplication algorithm basing on parallel Pipelining[C]. 2009 AsiaPacific Conference on Information Processing. Shenzhen, China, IEEE Computer Socitey Press, 2009:398-403.

[3] KOC C C, ACAR T, KALISKI B S. Analyzing and comparing montgomery multiplication algorithms[J]. IEEE Micro. 1996,16(3):26-33.

[4] WU C L. An efficient common-multiplicand-multiplication method to the montgomery algorithm for speeding up exponentiation[J]. Information Scien- ces. 2009,179(4):410-421.

[5] 王红霞,王金荣,赵宪生. Montgomery 模乘算法的改进及其应用[J]. 计算机工程与应用, 2007,43(20):52-55.

[6] 蔡风景, 李涛.一种快速乘法算法—Karatsuba 乘法算法[J]. 湘潭师范学院学报, 2004,26(1):55-56.

(收稿日期：2010-04-30)

作者简介：

羊红光，男，1979 年生，研究实习员，硕士，主要研究方向：密码算法及实现。

黄世中，男，1973 年生，副研究员、博士生，主要研究方向：密码学。

张利民，男，1982 年生，助教、硕士，主要研究方向：计算机安全。