计算环境 MPI 在 RSA 加密中的应用

韩 健,张 乐,蔡瑞英

(南京工业大学 信息科学与工程学院,江苏 南京 210009)

摘 要:加密过程中常用的 RSA 算法涉及到比较多的大数运算和一些复杂判断。实现起来比较困难,对计算机的速度、容量等要求都比较高。为了改变这种情况,采用 MPI 并行计算环境来改进 RSA 算法,结果表明能够较好地提高加密的速度和加密效果。

关键词:MPI;RSA;分布式计算;并行计算*

中图分类号:TP309.2 **文献标识码**:A **文章编号**:1671-7643(2003)03-0049-03

公开密钥密码体制是一种非对称密钥密码体制,它的加密数据和解密数据的密钥是不相同的,因此它的加密密钥不象以往的对称密钥密码体制那样需要保密,也就不需要专门提供一个安全渠道来交换密钥。RSA 密码系统就是公钥密码体制中的一个典型代表。RSA(由 Ronald Rivist、Adi Shamir 和 Len Adlemar 开发出的第一个公钥密码体制)密码体制是基于群 Z_n(包含 0,1,…,$n-1$ 这些元素的集合,还包含'+'运算和其他一些定义性质)中大整数因子分解的困难性,然而这种密钥密码体制涉及到比较多的大数运算和复杂判断,对计算机的速度、容量等要求都比较高,因此实现起来比较困难。然而并行处理在解决一些复杂计算过程中体现出来的优越性,为改进 RSA 加密算法提供了一种可能。MPI(Message Passing Interface 消息传递接口)是在分布式计算中广泛使用的一种基于消息传递的编程模型,它的使用和实现非常简单方便,特别是对算法过程比较明确的问题,实现起来有很强的优势性。本文提出了将 MPI 用到改进 RSA 加密算法中,事实证明这样克服了原来 RSA 算法的上述缺点,提高了算法的加密速度,并增强了安全性。

1 RSA 密码算法的原理[1]

RSA 是对明文采用分组的方式加密。它随机地选择两个大素数 p 和 q(需保密),计算乘积 $n=p\times q$(不用保密)然后利用欧拉函数 $\phi(n)=(p-1)\times(q-1)$ 计算小于 n 并与 n 互质的整数个数 选择一个奇数 K,令 $1<K<\phi(n)$ 且 K 与 $\phi(n)$ 互质,将 K 作为 SK(加密密钥)或 PK(解密密钥)。用 Euclid 算法计算模数为 $\phi(n)$ 时 K 的乘法逆元,等价于求满足 $u\times\phi(n)+v\times K$ 的 v,即求 v,使 $v\times K=1 \bmod \phi(n)$,将 v 作为 PK 或 SK,u 是求模得到的商 保密 SK、p 和 q,而公开 n 和 PK,任何想给本机传送加密信息的人都可以得到 n 和 PK 而加密时对于一个分组的明文 x,通过计算 $c=x^{PK} \bmod n$ 得到相应的密文 c 解密时候则利用 $x=c^{SK} \bmod n$ 重新计算出明文。由于 RSA 涉及大数计算,无论是硬件实现还是软件实现的效率都比较低,其硬件实现的效率是 DES 加密算法的 1/1 000,软件实现的效率是 DES 算法的 1/100。因此,如何采用适当的方法来改进 RSA 算法,以提高它的效率,就显得很重要。

2 并行计算环境 MPI[2~5]

MPI 是伴随着多处理机系统的发展而被广泛使用的。MPI 的优越性很大程度体现在它的可移植性和可扩展性上。随着网络速度的提高和分布共享体系结构的完善,这种特点可以得到更好的发挥。

MPI 可以确保异构环境下数据格式的统一。对于分布存贮的网络环境,不同节点上的程序可以通过消息传递来实现同步及数据共享,而传递原语中,

* 收稿日期:2002-12-16
作者简介:韩 健(1979-),男,江苏南通人,硕士生,主要研究方向为网络安全与网络管理。

消息的大小是一个很重要的参数,以往的编程平台要求程序员给出消息的总字节数以确定消息的大小,特别在并行加密运算这类应用中,需要明确指明数据位数。由于不同类型的计算机在表示同一类型的数据时,所采用的机器字长可能不一样,这样就不可避免地给消息传递带来问题,即表示相同类型的同样个数的数据所占缓冲区内存总字节数随机器的不同而变化,从而使得并行程序对硬件有极强的依赖性,在程序移植时,用户不仅需要深入了解机器特性,还有可能需要修改每一条消息传递原语,不利于并行程序的交流及并行计算技术的推广。MPI 针对上述问题,将消息的大小信息用所传递的数据类型及个数来确定,这样程序员只要在通讯原语中指明数据类型及个数,MPI 内核将自动完成机器间字长的转换,这为并行程序的移植及在异构网络上开发、运行并行程序提供了有力的支持。

MPI 在进行通信时是采用的点对点的通讯方式。从语义上讲,MPI 的发送和接收可以分为阻塞方式和非阻塞方式。在阻塞方式中,发送操作的完成意味着用户可以重用该操作的消息缓冲区,接收完成意味着所要接收的消息已存在于用户的缓冲区中。为了支持通信、计算和并行,MPI 也支持非阻塞方式的发送接收操作。这时操作分为两个阶段,即首先发出请求,要求 MPI 发送或接收,然后用户可能通过测试或等待操作来最终完成该发送接收。在采用并行加密的过程中,包含了大量的计算和进程间的协调和通信,因此 MPI 这种兼顾的通信方式,带来了很大的便利。

3 分布式并行计算环境 MPI 在 RSA 中的应用[1]

根据 MPI 在并行处理中的优点和 RSA 算法在一般实现中的困难,于是提出了利用 MPI 这样的消息传递系统,对 RSA 算法进行优化,以提高算法的实现效率并提高加密的安全性。采用并行算法以后的 RSA 算法具体实现如下。

RSA 算法中后一步过程往往需要使用到前一步运算结果,如果整体处理过程完全使用并行计算,在数据交换和通信过程上会耗费大量的系统开销,而且实现起来也比较困难。反而影响效率。从这一点考虑,采用全局串行处理,局部并行处理的方法更加实际。主要对随机产生质数过程,欧拉函数的计算、计算逆元的过程,以及加密时的大数取模过程采用并行处理。RSA 加密中有两个关键问题:

一是大质数 P、Q 的取得。传统的判断素数的方法是使用所有小于$\sqrt{p}$的整数去除 p,这对这里采用的大素数是不可取的,因此选择利用取大数(100 位以上的整数)p 和整数 a($a < p$),计算 $a^{p-1} \bmod p$,若结果不为 1,则 p 必定不是素数;若结果为 1,则 p 不为素数的概率大约只有 10^{-13}。

二是逆元的计算。计算逆元的过程比较困难,本文选择 Euclid 算法来计算逆元。

通过分析,对整个系统最后确定了 4 个调用单元,系统就通过这些单元模块的有效组合来实现。在建立这个系统时,为了使消息传递和计算方便,使用数组存放需要计算的每一位数据,将需要计算的数的每一位对应存放到 1 个整型数组的每一位中。以模块中计算随机产生的 2 个大素数的乘积来说明:

大数相乘的时候将乘数和被乘数通过 MPI 消息传递的方式传递给每 1 台机器或每 1 个进程上,然后在 MPI 中设置 1 个步长,以确定并行计算时的每台机器计算的相应位。下面以 $ABCD \times EFGH$ 为例进行计算。

(1)初始化阶段

通过发送消息,使得每个进程中都存放了乘数和被乘数:$ABCD$ 和 $EFGH$。

设置存储单元 1 存放乘数:

A	B	C	D

设置存储单元 2 存放乘数:

E	F	G	H

(2)计算阶段

设置 1 个 MPI 的步长。可以使用乘数的位数除以开设的进程数(或机器数)得到的商作为步长。这里是 4/2 即 2。使用第一个进程专门计算与 2 求模余 1 的位数上的数,第二个进程则计算与 2 求模余 0 的位数上的数,每个进程独立运算后得到各自的结果。

(3)汇总阶段

将每个进程的结果再次通过 MPI 消息发送到一号进程进行汇总计算,同时进行进位调整,可以得到最终的结果。

其中 MPI 主要的代码如下:

```
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
  /* MPI 系统步长 */
MPI_Comm_size(MPI_COMM_WORLD, &ranksize);
  /* MPI 系统大小 */
……  /* 给数组赋值 */
MPI_Barrier(MPI_COMM_WORLD);
t1 = realtime();
MPI_Bcast(FirstArray,NUM * NUM,MPI_INT,0,MPI_
COMM_WORLD);  /* 将被乘数数组广播给每个
进程 */
MPI_Bcast(SecondArray, NUM * NUM, MPI_INT, 0,
MPI_COMM_WORLD);  /* 将乘数数组广播给每
个进程 */
……  /* 并行计算乘法的过程 */
for(i = 0;i < NUM;i + +)
    for(j = 0;j < NUM;j + +)
     {
       int Result,temp1;
       Result = ResultArray[i][j];
       MPI_Reduce(&Result,&temp1,1,MPI_INT,
            MPI_SUM,0,MPI_COMM_WORLD);
            /* 将各个进程的计算结果发回给一号
            进程 */
       ResultArray[i][j] = temp1;
    }
```

其他几个功能函数的设计思想与上述类似,就是充分利用 MPI 在多台机器或多个进程中的消息传递机制,将本来需要在 1 台机器上或 1 个进程上完成的作业,分发到多台机器或多个进程中,提高整体加密速度。最终系统实现时,其流程如图 1 所示。

4　算法实验和结果

在现实管理系统中往往涉及到不同地区间的数据传送,包括了一些需要保密的财务数据的传送,需要对其中的一些数据加密。以我们所做的一个医保系统为模型,选用 512 位长度的 p,q 生成的公钥,密钥和模 n。对其中的部分财务数据文件进行加密。其步骤如下:

在主频为 333 MHZ 的 Intel 中央处理器、128 MB 内存、Win2000 操作系统的测试环境下的加密时间如下:使用 1 个进程需要 4.52 s,使用 2 个进程需要 3.13 s,使用 2 台机器各开一进程需要 3.87 s。

当时文件中数据量较少,随着数据量增大,文件容量变大,在时间上的优越性会进一步显示出来。同时表明在小数据量加密时候,机器间的网络通信占用的开销会比较明显,但数据计算量增大以后,网络通信时间所占比例会下降。

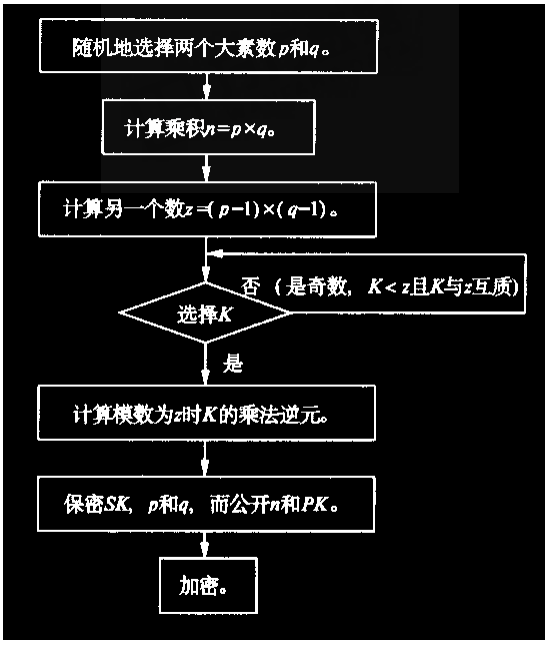

图 1　算法流程图

Fig.1　The process of the algorithm

5　结　论

采用分布式计算的方法处理 RSA 加密,既完全满足了加密算法的要求:安全,高效,准确;又发挥了并行处理的优点:快速,充分利用资源。而 MPI 正是实现这一结合的良好的环境。

参考文献:

[1] Carlton R Davis. IPSec:VPN 的安全实施[M]. 北京:清华大学出版社,2002.

[2] 都志辉. 高性能计算并行编程技术—MPI 并行程序设计[M]. 北京:清华大学出版社,2001.

[3] 李　东,李晓明. MPI 并行编程环境若干技术研究[J]. 哈尔滨工业大学学报,1996,20(3):25-28.

[4] 赵军锁,周恩强,杨学军. 主动消息与 MPI[J]. 小型微型计算机系统,1999,20(3):209-213.

[5] 赵军锁,周恩强. 消息传递、PVM 及 MPI[J]. 电脑与信息技术,1998,2:11-15.

(下转第 61 页)

[6] Crampon C, Charbit G, Neau E. High-pressure apparatus for phase equilibria studies: solubility of fatty acid esters in supercritical CO_2 [J]. Journal of Supercritical Fluids1999,16:11-20.

Calculation of phase equilibrium of systems containing supercritical carbon dioxide and fatty acid esters using artificial neural networks

GUO Ning[1], YUN Zhi[1], SHAO Rong[2], SHAO Hua-guang[1], SHI Mei-ren[1]

(1. College of Chemistry and Chemical Engineering, Nanjing University of Technology, Nanjing 210009, China;
2. Department of Chemical Engineering, Yancheng Institute of Technology, Yancheng 224000, China)

Abstract: Based on the data of phase boarders of systems containing supercritical carbon dioxide and fatty acid esters, the phase equilibria data for these systems are obtained using BP artificial neural network with $2 \times 2 \times 1$ frame, and correlated well using BP artificial neural network with $2 \times 4 \times 2$ frame. The parameters involved in the calculations are discussed.

Key words: artificial neural network; supercritical; carbon dioxide; fatty acid ester; phase-equilibrium

(上接第 51 页)

Application of parallel computing environment MPI in RSA

HAN Jian, ZHANG Le, CAI Rui-ying

(College of Information Science and Engineering, Nanjing University of Technology, Nanjing 210009, China)

Abstract: RSA is a kind of encrypt algorithm in common use. But this kind of algorithm involves a lot of large numbers and some difficult estimates. So it is hard to turn it into reality and always requires a computer of high speed and large capacity. MPI is a kind of parallel computing environment based on message passing. So a good result and high speed by using MPI in RSA, could be achieved.

Key words: MPI; RSA; distributed computing; parallel computing