第 24 卷　第 3 期　　华侨大学学报(自然科学版)　　Vol. 24　No. 3
2003 年 7 月　　Journal of Huaqiao University (Natural Science)　　Jul. 2003

文章编号　1000-5013(2003)03-0321-04

用类的 RSA 体制实现方案

郑子伟　　李翠华

(厦门大学计算机科学系, 福建 厦门 361005)

摘要　通过对 RSA 算法原理的分析和实现方法的研究, 构造了数字签名软件的数据结构. 用类的对象对自定义函数设计模块进行调用, 实现任意长度数据的运算. 针对 RSA 实现算法运算速度慢的特点, 在生成密钥对的过程中采用小素数翻番、欧几里得扩展算法、二元法等一系列方法, 以加快算法实现速度. 从而, 在微机上实现了数字签名的软件开发.

关键词　公钥密码, RSA, 数字签名, 长整数

中图分类号　TP 309.7　　文献标识码　A

数字签名作为一种近年来迅速发展的实用认证技术, 已成为数字媒体上取代传统印鉴、手写的认证技术. 它最早是于 1976 年由 Diffie 和 Hellman 提出[1], 特别是 1978 年由于 RSA 公钥体制以及 RSA 签名方案的提出[2], 因而突破了传统加密算法的界限. 它很好地解决了密钥分发方面的困难, 证实了实际可行的数字签名方案的存在. 目前, 全世界的诸多商业、研究机构, 甚至政府机构, 大家都在使用它. 这种广泛的应用, 从另一侧面反映了 RSA 体制的实用性、可靠性及广阔的发展前景. 但是, RSA 体制实现过程存在着运算量大, 要求计算机具有高速度、大容量的缺点. 因而, 它使得如何根据 RSA 算法的基本原理, 寻找其实用的方法研究及实现软件的开发, 成了一个重要的研究课题.

1　RSA 算法原理

RSA 算法的安全依赖于大数因子分解的困难性, 算法的基础是数论的欧拉定理[3].

1.1　系统参数

p, q 均为 100 位以上十进制数的素数(秘密, 这里的 p, q 均为强素数. 为了安全性, 它们应相差几位以上)[4]. 有

$$n = p \cdot q \text{(公开)}, \qquad \Phi(n) = (p-1)\cdot(q-1) \text{ (秘密)}.$$

公钥(K_p)为 $E=(n,e)$, 其中 e 为任选的整数, 它应满足(e, $\Phi(n)$) = 1(公开). 私钥(K_s)为 $D=(d, \Phi(n))$, 其中 K_p 满足 e 与 $\Phi(n)$ 互素, K_s 满足 $K_p \cdot K_s \equiv 1 (\mathrm{mod}\,\Phi(n))$ 的条件. 与此同时, $M^{K_p \cdot K_s} \equiv M (\mathrm{mod}\ n)$, 即明文 M 运算 $K_p \cdot K_s$ 次幂后对 n 取模, 仍是明文本身. M 为明文(秘密), C 为密事(公开).

收稿日期　2003-01-04

作者简介　郑子伟(1969-), 男, 讲师, 在职硕士研究生, E-mail: ziweizhen@163.com

1.2 加密签名过程

(1) 加密. $C = E(M) = M^e \pmod n$. (2) 解密. $M = D(C) = C^d \pmod n$. (3) 签名. $D(M) = S = M^d \pmod n$. (4) 验证. $E(S) = S^e = M \pmod n$. 如果 $M' \neq M$ 则验证正确, 否则验证失败. 由于任何人均可用爱丽斯的公钥验证, 且只有爱丽斯本人知道用 d 来产生 S. 因此发生争论时, 公正的第三方可做出公正的仲裁.

2 数字签名方案的软件开发

RSA 应用系统开发, 须按一定的步骤进行.

2.1 系统参数产生过程

根据 RSA 算法原理, 产生系统参数 $p, q, e, d, n, \Phi(n)$, (n, e) 存放于 Public.rsa 中为公开文件, $(p, q, d, \Phi(n))$ 可在生成后, 另存在其它的安全地方后消毁它.

2.2 签名(加密) 验证(解密) 过程

具体过程, 如图 1 所示.

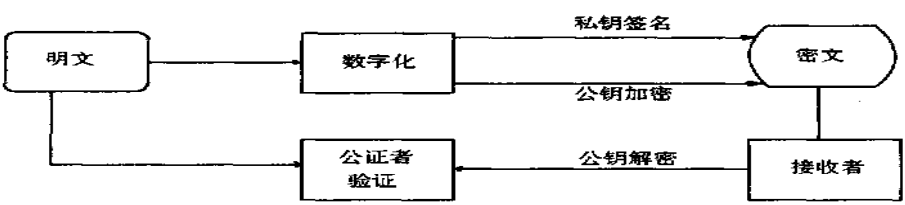

图 1 签名(加密) 验证(解密)过程示意图

2.3 高精度长整数基本函数定义

在实现 RSA 算法及其整个运算过程中, 参与运算的均是高精度, 长度在 100 位以上的运算. 在 C 语言中表示整数的最大限度(Unsigned Long Int), 其二进制为 32 位, 即只能为 9 位的十进制. 所以, 对于 100 位以上长整数的实际处理, 必须重新定义其算术运算子程序. 本系统中我们采用了字符串结合数组的形式, 表示并处理这些长整数. 其数据结构为

```
class strdata: {char* str1, char* str2, string1[110], string2[110]};
```

下面介绍 6 种自定义基本函数模块. (1) stradd: 求 2 个数的和. (2) strsub: 求 2 个数的差. (3) strmul: 求 2 个数的积. (4) strdiv: 求 2 个数的商. (5) strmod: 求 2 个数的余数. (6) strpow: 求 2 个数的幂. 利用上述自定义函数子程序, 我们实现了对任意长度整数的加减乘除、求余、求幂运算. 例如, $2^{1\,009}$ 可通过 strpow(2, 1 009) 运算, 因而得到一个长度为 304 个十进制位的长整数.

2.4 素数生成算法

素数的生成方法, 可分为概率法和确定性法两大类. 其中, 确定性方法如 1989 年 Demytko 对 1983 年 Adleman, Millert 和 1981 年 Trbovich 的方法, 做了一些改进和扩充. 由此, 产生了一种能够产生并验证大素数高效的非概率方法, 即 MTD 法. MTD 法[5], 它对于一个给定的正整数有 $p_{i+1} = h_i \cdot p_i + 1$. 只要这里的 p_i 是个奇素数, h_i 是一个小于 p_i 的偶数. 并且, P_{i+1} 又能通过以下两种检测: $2^{h_i p_i} \equiv 1 \pmod{p_{i+1}}$, $2^{h_i} \neq 1 \pmod{p_{i+1}}$, 则 p_{i+1} 必为素数. 其产生翻番素数的流程图, 如图 2 所示. 利用如图 2 的流程, 由一个十进制的小素数 1 009 通过 5 次素数翻番, 可

以得到一个 106 位十进制素数, 如表 1 所示.

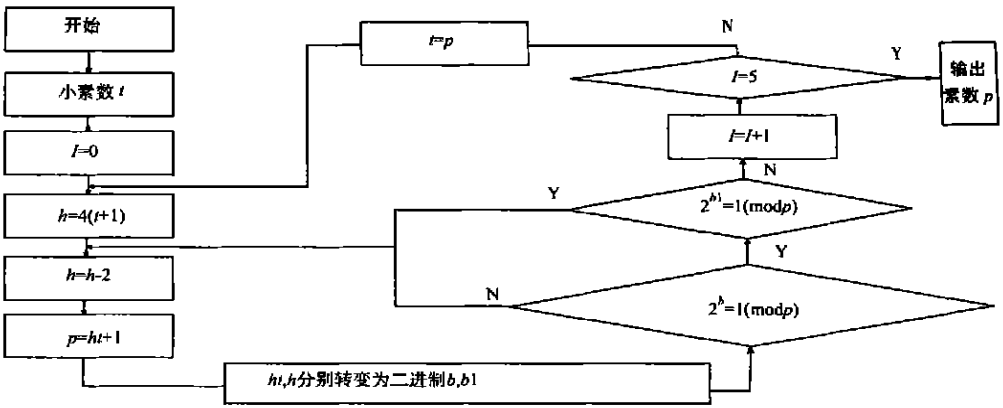

图 2　产生翻番素数的流程图

表 1　翻番后的素数表

翻番次数	结　　果
0(4 位)	1 009
1(7 位)	4 036 001
3(29 位)	16 981, 76 297 825, 42 573 566, 48 472 807
4(58 位)	11, 53 521 095, 39 842 761, 93 643 960, 05 210 505, 59 311 359, 63 429 319, 79 275 597
⋮	⋮

2.5　计算密钥对

求解 K_s 的算法流程图, 如图 3 所示. 图中, a 代表 $\varphi(n)$, b 代表 K_s. RSA 算法的公钥(n, e)

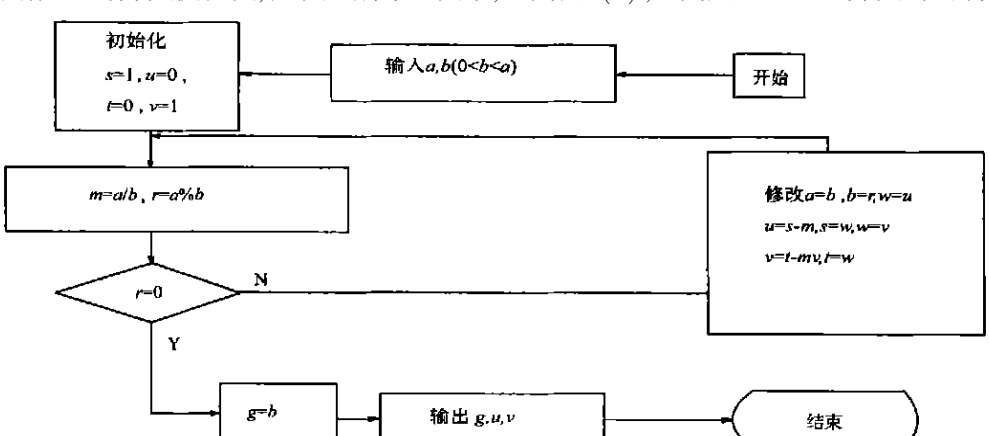

图 3　求解 K_s 的算法流程图

为公开的, 其中 $n = p \cdot q$, e 为随机选取的整数且满足$(e, \Phi(n)) = 1$. 为了安全, e 不能太小, 兼顾加密速度一般选它的长度为 16 位[4]. 私钥的产生多采用 Euclid 算法, 但 Knuth 等人又提出了扩展的 Euclid 算法[6]. 后者比前者速度有所提高. 私钥的产生是在 K_p 产生之后, 通过求解同余方程 $K_p \cdot K_s \equiv 1 (\mathrm{mod}\,\Phi(n))$, 解出 K_s 值. 图 3 为求解 K_s 的算法流程.

2.6 快速指数运算

高次幂的运算是数字签名软件中运算量最大的计算,可用二元法[7]来加快指数运算.即将十进制指数转化为二进制指数,再进行运算.具体方法是从左边的第2位开始向右,遇到"0"就将A平方,遇到"1"就将A平方后再乘以A.其目的是减少循环的次数,以此大大加快指数运算.例如,A^{23}转变为$A^{10\,111}$及计算过程为

$$A \rightarrow A^2 \rightarrow A^4 \cdot A \rightarrow A^{10} \cdot A \rightarrow A^{22} \cdot A \rightarrow A^{23}.$$

它只需7次循环而不是23次,大大加快了指数运算.

3 结束语

本文采用小素数翻番、欧几里得扩展算法、二元法等一系列方法,加快了算法实现速度,能在微机上实现数字签名软件的开发.今后我们将继续研究:(1) 用并行处理方式提高RSA运算的速度;(2) 找出程序中较费时的模块,进行优化;(3) 研究寻找更新更好的算法,以期获得更高效的应用系统.

参 考 文 献

1 Diffe W, Hellman M E. New direction in cryptography[J]. IEEE Trans Information Theory, 1976, 22: 644～654

2 Revest R L. A method for obtraining digital signature and public key cryptosystems[J]. Commun, ACM, 1978, 21: 120～126

3 朱文余,孙 琦. 计算机密码应用基础[M]. 北京:科学出版社, 2000. 113～114

4 赖溪松,韩 亮,张真诚. 计算机密码学及其应用[M]. 北京:国防工业出版社,2001. 91～91

5 覃 颖. 基于手写签名的网络责任文件认证系统[D]:[硕士学位论文]. 武汉:华中理工大学计算机科学系,1999. 16～17

6 Knuth D E. The art of computer programming (volume 2) seminumerical Algorithms[M]. London: Addison-Wesley, 1973. 1～100

7 张玉清,肖国镇. 数字签名方案的软件开发[J]. 西安公路交通大学学报, 1999, 19(4): 119～119

Realization of Class-Based RSA System

Zheng Ziwei Li Cuihua

(Dept. of Computer Science, Xiamen Univ., 361005, Xiamen, China)

Abstract By analysing the principle of Rivest-Shamir-Adelman(RSA) system or RSA system for short and by studying the realization of RSA algorithm, the data structure of digital signature software is constructed. By using class as object to call the module of self-defining function design, the operation of data with arbitrary length is realized. In view of slow realization of RSA algorithm operation, a series of methods including doubling of small prime and Euclidean algorithm extension and binary algorithm are adopted during the generation of private key cryptosystem. By which the realization of algorithm can be accelerated; and software of digital signature can be developed and realized on microcomputer

Keywords public key cryptosystem, RSA, digital signatare, long integer