基于乘同余对称特性的快速 RSA算法的改进*

陈 运**
(电子科技大学电子技术系 成都 610054)

【摘要】 提出了一种新的快速 RSA算法,这种算法是基于乘同余对称特性的快速 RSA算法(SMM算法)的改进。理论分析表明,新算法的迭代计算步数平均减少了 17.2%。与传统的二进制算法(BR算法)相比,新算法的计算速度平均提高了 42%左右。

关 键 词 密码学; 公钥密码体制; RSA; 幂剩余; 快速算法
中图分类号 TN911.2; TN911.7

RSA(Rivest, Shamir & Adleman)是一种国际公认的理想公钥密码体制。它表达方式简单、保密性强、没有密钥管理的麻烦,并且具有数字签名、认证和鉴别等功能,特别适合于现代保密通信的需要。这些优越性都归功于其核心——大整数幂剩余计算。但是大整数幂剩余运算会使其计算速度非常缓慢,难以实际应用。为了提高 RSA公钥密码体制的运算速度,人们一直在寻找它的各种快速算法[1~7]。由于目前国际上流行的其他几种公钥密码体制的关键也都是大整数幂剩余计算,因此,研究快速 RSA算法是十分有意义的。

本文提出的快速 RSA算法是在文献[6]提出的快速算法的基础上,进一步减少运算的迭代步数,从而使 RSA算法的运算速度进一步提高。

1 BR算法简述

几乎所有的快速 RSA算法都是建立在 BR(Binary Reresentations)算法的基础上,以下对 BR算法进行简要介绍。

RSA公钥密码体制的数学表示为

$$y \equiv x^e \pmod{M} \tag{1}$$

$$x \equiv y^d \pmod{M} \tag{2}$$

其中

$$M = pq \tag{3}$$

且有

$$ed \equiv 1 \pmod{\varphi(M)} \tag{4}$$

式中 x为明文; y为密文; "≡"为同余号; p、q均为大素数; e、d均为正整数且满足式(4)的关系; $\varphi(M)$为 M的欧拉函数[9]。式(1)是加密算式,式(2)是解密算式。

将指数 e表示成二进制形式

$$e = \sum_{i=0}^{n-1} e_i 2^i \qquad e_i \in \{0, 1\} \tag{5}$$

BR算法是将幂剩余变成一系列平方剩余和乘同余的迭代,即式(1)变成

⑳ 1996年 11月 6日收稿, 1997年 1月 9日修改定稿
* 电子部预研基金资助项目
** 女 38岁 硕士 副教授

$$y \equiv (((((1 \cdot x^{e_{n-1}})_M^2 x^{e_{n-2}})_M)_M^2 \cdots x^{e_1})_M)_M^2 x^{e_0})_M \tag{6}$$

式中 $(\circ)_M$表示括弧中的数对 M求模; $(\circ)_M^2$表示先对括弧中的数求平方再对 M求模.

BR算法的迭代步数为[6]

$$l = n + h(e) - 2 \tag{7}$$

式中 $h(e)$表示 e的汉明重量.

2 基于乘同余对称特性的快速 RSA算法

令 $i \notin \{0, 1, \cdots, \frac{M-1}{2}, \frac{M+1}{2}, \cdots, M-1\}$由乘同余和平方剩余的对称性有

$$(M - i)^2 \equiv i^2 \pmod M \tag{8}$$

$$(M - i)(M - j) \equiv ij \pmod M \tag{9}$$

$$i(M - j) \equiv (M - i)j \equiv - ij \pmod M \tag{10}$$

式(6)的迭代运算实际只含两种基本运算. 令 A_i为第 i步迭代后的中间结果, x为待加密明文,则

$$A_i, x \in \{0, 1, \cdots, \frac{M-1}{2}, \cdots, M-1\}$$

两种基本运算分别是 $A_i^2 \pmod M$和 $A_i x \pmod M$, $i = 1, 2, \cdots, l$.

在式(6)的每步迭代计算中进行有条件代换,即可构成一种快速算法. 代换原则是: 如果 $A_i, x > \frac{M-1}{2}$,则用 $(M - A_i)$或 $(M - x)$代替 A_i或 x进行平方剩余或乘同余计算. 根据式(8)~(10),其计算结果不变,但由于减少了乘法时间和求模运算量,使算法速度平均提高了 30%以上[6]. 为了方便起见,称这种算法为 SMM(Symmetry of Modulo Multiplication)算法.

3 改进的 SMM算法

上述算法的迭代步数 l取决于指数的二进制长度及其汉明重量—— 即非零元素的个数. 如果能够降低二进制指数的汉明重量,迭代步数必将减少.

用“$\bar{1}$”表示“ - 1”,二进制的连“ 1”可表示成如下形式

$$\begin{cases} (11)_2 = (100)_2 - (1)_2 = 10\bar{1} \\ (111)_2 = (1000)_2 - (1)_2 = 100\bar{1} \\ (1111)_2 = (10000)_2 - (1)_2 = 1000\bar{1} \\ \cdots \\ (11\cdots1)_2 = 10\cdots0 \end{cases} \tag{11a}$$

当“ 110”和“ 1110”码型之后出现 k个($k \geqslant 2$)连“ 1”时,也可进行如下二次代换

$$\begin{cases} (11011)_2 = (100000)_2 - (1)_2 - (100)_2 = 10000\bar{1} - (100)_2 = 100\bar{1}0\bar{1} \\ \cdots \\ (1110\,1\cdots1)_2 = 1000\bar{1}0\cdots0\bar{1} \end{cases} \tag{11b}$$

式(11a)和式(11b)等号右边由 $1\ 0\ \bar{1}$组成的数称为二进制冗余数. 容易看出,当连“ 1”个数大于等于 3或“ 110”及“ 1110”码型之后的连“ 1”个数大于等于 2时,二进制冗余数的汉明重量低于二进制

数的汉明重量。

用式(11a)和式(11b)的方法对二进制序列中三个以上连“1”及“110”和“1110”码型之后两个以上连“1”进行替换。例如: $e=(101110101111)_2$,若 $R(e)$表示对应于 e的二进制冗余数,则 $R(e)=1\bar{1}00\bar{1}0\bar{1}0\bar{1}000\bar{1}$。不难验证 $e=R(e)$。但是 e的汉明重量是 9,而 $R(e)$只有 5,明显小于 e。

如果 e的最高位含有 3个以上连“1”,则 $R(e)$的位数比 e多 1,其他情况下两者位数相等。

用 $R(e)$代替 e,式(1)变为

$$y\equiv x^{R(e)} \pmod{M} \tag{12}$$

其中,$R(e)=\sum_{i=0}^{k} e_i 2^i$　$k=n$或 $n-1$,$e_i\in\{0,1,\bar{1}\}$。

用式(6)的方法对式(12)进行迭代计算时,含有三种基本运算,即 $A_i^2 \pmod{M}$、$A_i x \pmod{M}$和 $A_i x^{-1} \pmod{M}$。x^{-1}是 x对模 M的乘逆,即

$$xx^{-1}\equiv 1 \pmod{M} \tag{13}$$

已知模数 $M=pq$的欧拉函数 $\varphi(M)$表示小于 M又与 M互素的数的个数,则 $\varphi(M)=(p-1)(q-1)$[9]。欧拉函数与模数之比

$$\frac{\varphi(M)}{M}=\frac{(p-1)(q-1)}{pq}=1-\frac{1}{p}-\frac{1}{q}+\frac{1}{pq} \tag{14}$$

当 M是 200位的十进制数时,式(14)的比值趋近于 1。这说明几乎所有的 $x\in\{0,1,\cdots,M-1\}$都与 M互素,即满足

$$(x,M)=1 \tag{15}$$

满足式(15)的 x,对模 M的乘逆 x^{-1}必存在[9]。据此,可以构造一种新的快速 RSA算法,步骤如下:

1) 将加密指数 e的二进制形式变换成二进制冗余数形式,记为 $R(e)$。变换原则是,从 e的高位开始,当 e的二进制数序列中有 k个($k\geq 3$)连“1”时,用 $1\,0\cdots 0\,\bar{1}$将这 k个连“1”替换掉。如果序列中“110”或“1110”码型之后有两个以上连“1”时,也可进行连续代换。如(11011)。第一次代换成 $1\bar{1}0\bar{1}1$,第二次代换成 $1\bar{0}0\bar{1}0\bar{1}$。再如(111011),可连续代换成 $1\bar{0}00\bar{1}0\bar{1}$。

2) 用欧几里德算法[9],由式(15)求出 x对模 M的乘逆 x^{-1}。

3) 令

$$x=\begin{cases} x & x\leqslant \dfrac{M-1}{2} \\ M-x & x>\dfrac{M-1}{2} \end{cases}$$

$$x^{-1}=\begin{cases} x^{-1} & x^{-1}\leqslant \dfrac{M-1}{2} \\ M-x^{-1} & x^{-1}>\dfrac{M-1}{2} \end{cases}$$

4) 从 $R(e)$的最高位开始,按式(6)迭代计算幂剩余 y。若 A_i为第 i步迭代的中间结果,则具体算法为:

$$(1)\,A_i\begin{cases} A_i & A_i\leqslant (M-1)/2 \\ M-A_i & A_i>(M-1)/2 \end{cases} \qquad A_{i+1}=A_i^2 \pmod{M}$$

$$(2)\,A_i\begin{cases} A_i & A_i\leqslant (M-1)/2 \\ M-A_i & A_i>(M-1)/2 \end{cases} \qquad A_{i+1}=A_i x \pmod{M}$$

$$(3)\ A_i=\begin{cases} A_i & A_i\leqslant (M-1)/2 \\ M-A_i & A_i>(M-1)/2 \end{cases} \qquad A_{i+1}=A_i^2x^{-1}(\mathrm{mod}\ M)$$

当 $R(e)$的第 i位为“0”时,进行(1)的运算;为“1”时,进行(1)、(2)的运算;为“$\bar{1}$”时进行(1)、(3)的运算。从 $R(e)$的最高位开始,直至将 $R(e)$的所有位都计算完为止。

容易推知,新算法的迭代步数为

$$\dot{l}=n+h[R(e)]-2 \tag{16}$$

式中 $h[R(e)]$为 $R(e)$的汉明重量。

4 改进算法的速度分析

改进算法是在 SMM算法的基础上又减少迭代步数得到的。SMM算法比传统的 BR算法速度提高了 30%以上[6]。下面仅分析采用二进制冗余数得到的速度改善。

假设 e的二进制数是一个随机的二进制序列。根据戈龙随机性公设[10],k个连“1”出现的概率是 $1/2^k$。在实际应用中,e的长度大约是 300多 bit,那么出现三个以上连“1”的概率大约是

$$P_1=\sum_{k=3}^{n}\frac{1}{2^k} \tag{17}$$

当 n很大时,式(17)的极限是 1/4。

在 e是随机二进制序列的假设条件下,“110”和“1110”码型之后出现 k个($k\geqslant 2$)连“1”的条件概率仍是 $1/2^k$。由于“110”和“1110”码型出现的概率分别为 $1/2^3$和 $1/2^4$,“110”和“1110”与 k个($k\geqslant 2$)连“1”发生的联合概率分别为 $1/2^{k+3}$和 $1/2^{k+4}$,所以对“110”和“1110”码型之后出现两个以上连“1”的情况进行二次代换的可能性分别为

$$P_2=\sum_{k=2}^{n-3}\frac{1}{2^{k+3}} \tag{18}$$

$$P_3=\sum_{k=2}^{n-4}\frac{1}{2^{k+4}} \tag{19}$$

式(18)、(19)的极限分别为 1/16和 1/32。

忽略多次代换的情况,一、二次代换比特数占总比特数的比率为

$$P=P_1+P_2+P_3=34.4\% \tag{20}$$

计入一、二次代换减少的非零位个数,再考虑到 k个连“1”发生的概率,可以推算出在超过 1/3的代换中,非零位个数减少了一半。每减少一个非零位,迭代步数就减少一步,故迭代步数平均减少量约为

$$34.4\%\times\frac{1}{2}=17.2\% \tag{21}$$

选择模数 $M=97\times 157=15\,229$,其欧拉函数 $\varphi(M)=14\,976$。RSA公钥密码体制要求公开密钥和秘密密钥必须满足

$$(d,\varphi(M))=1 \tag{22}$$

$$ed\equiv 1\ (\mathrm{mod}\ \varphi(M)) \tag{23}$$

用欧几里德算法随机产生出满足式(22)、(23)的 20个加密和解密密钥对,分别用二进制数和二进制冗余数表示每个密钥对 (e_i,d_i) $(i=1,2,\cdots,20)$,用式(7)和式(16)计算 BR算法和新算法的迭代步数,并加以比较,结果列于表 1。

表 1　BP算法和新算法迭代步数的比较

序号 i	密钥对 (e_i, d_i)	BR算法迭代步数 l	新算法迭代步数 l	迭代步数减少量 $(1-\frac{l'}{l})\times 100\%$
1	191, 5 567	34	25	26. 5%
2	251, 179	24	21	12. 5%
3	127, 10 495	34	24	29. 4%
4	347, 3 539	31	29	6. 5%
5	485, 2 285	30	27	10%
6	509, 3 413	32	28	12. 5%
7	935, 6 551	34	32	5. 9%
8	1057, 8 161	32	27	15. 6%
9	1021, 11 221	38	31	18. 4%
10	1259, 1 475	31	28	9. 7%
11	1 871, 13 103	38	34	10. 5%
12	2 045, 725	33	27	18. 2%
13	1 939, 7 963	36	32	11. 1%
14	3 059, 12 347	38	32	15. 8%
15	3 743, 11 231	41	31	24. 4%
16	4 607, 6 911	43	30	30. 2%
17	7 487, 7 487	42	34	19%
18	7 897, 11 113	40	36	10%
19	8 189, 7 253	41	33	19. 5%
20	9 215, 10 367	43	31	27. 9%
迭代步数平均减少量	16. 68%			

由表 1可以看出,迭代步数平均减少量与理论分析结果接近。虽然上述参数与实际参数相比小得多,数量也很少,仅 20组。但因这些参数是随机选取的,足以说明新算法的迭代步数确实有一定程度的减少。

由于新算法是在 SMM算法的基础上进一步减少迭代运算步数得到的,而 SMM算法已经使每步迭代的求模运算量平均减少了 30% 以上,故与传统的 BR算法相比,新算法的速度改善为

$$30\% + 17.2\% \times 70\% = 42\% \tag{24}$$

考虑到 SMM算法除了使求模运算量减少,还使乘法时间缩短这一因素,实际获得的速度改善要大于 42%。

5 结 束 语

RSA公钥密码体制的快速算法研究一直受到密码学界的高度重视, RSA的速度问题不解决,就难以实际应用,本文提出的快速算法是在 SMM算法的基础上进一步改进得到的。由于 RSA算

法的模数大到十进制的200位以上,指数大到100位以上,其运算速度非常缓慢,因而获得42%以上的速度改善是十分可观的。它所付出的代价是在进行迭代计算之前,先将指数的二进制 e 按一定规则变换成冗余二进制数 $R(e)$,这种变换可用简单的移位寄存器加逻辑电路的方法完成,因而易于实现。如果用软件模拟,则以汇编语言程序最为有效。

参 考 文 献

1 Vandemuleulebroeke Ardi, e, Vangielehem Etienne, Denayer Tony. A new carry-free division algorithm and its application to a single-chip 1024-bRSA processer. IEEE Journal of Solid-State Circuit, 1990, 25(3), 748~756

2 徐大专,邹深昌. RSA公钥保密系统的实现研究. 数据通信, 1990, (4): 1~7

3 Kawamara Chinich, Hiraro Kyoko. A fast modular arithmetic algorithm using a residue table. Advances in cryptology-Euro CRYPTO'88, 1988: 245~250

4 Zhang C N. An improved binary algorithm for RSA. Compter Math Applic, 1993, 25(6): 15~24

5 Nguyen Phong. Public-key cryptography: hardware implementation and novel nerual network-based approach. AD-A257103, sept, 1993: 25~29

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

7 陈 运. 一种组合快速 RSA算法. 电子科技大学学报, 1996, 25(2): 114~119

8 陈 运. 信息理论与编码. 电子科技大学讲义, 1996

9 陈 运. 信息加密原理. 成都: 电子科技大学出版社, 1996

10 贝克 H, 派普尔 F著, 通信保密编辑部译. 密码体制—通信保护. 成都: 电子部三十所出版社. 1982

An Improved Algorithm for RSA Based on Symmetry of Modulo Multiplication

Chen Yun

(Dept. of Electronic Tech., UEST of China Chengdu 610054)

Abstract A new fast RSA algorithm is presented in this paper, which is an improvement of a fast RSA algorithm based on symmetry of modulo multiplication (SMM algorithm). It is shown by theoretical analysis that the proposed algorithm decreases the recursive steps by 17.2% on average. Compared with traditional binary representations (BR algorithm), the new algorithm obtains the speed improvement by about 42% on average.

Key words cryptograph; public-key cryptosystem; RSA; modulo multiplication; fast algorithm

编辑 黄 辛