RSA 算法的研究和改进

陈春玲,齐年强,余 瀚
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:为了提高 RSA 算法的安全性,避免 RSA 密码体制中的模 n 被因子分解,导致私钥 d 泄露,采取消除密钥中 n 的分布方法以及对三个因素因子的加密方法。采用消除密钥中 n 的分布方法可以成功避免在公布的公钥和保密的私钥中出现 n,防止攻击者通过因子分解法分解出公钥中 n 的因子,推导出解密密钥 d;采用三因子的加密算法,即使攻击者知道了 $\varphi(n)$,但对于分解 n 的三个素数没有一个具体的公式,所以加大了分解的困难性,并且因为大素数适当减小了位数,降低了计算量。实验结果表明,该方法提高了 RSA 算法的安全性,时间复杂度与传统 RSA 算法相同。

关键词:RSA;安全性;公钥;私钥;时间
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2016)08-0048-04
doi:10.3969/j.issn.1673-629X.2016.08.010

Research and Improvement of RSA Algorithm

CHEN Chun-ling, QI Nian-qiang, YU Han
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: In order to improve the security of the RSA algorithm and avoid the mathematical factorization of the module n in RSA crytosystem which leads to private key d leaking, the method of eliminating the distribution of n and three-factor encryption algorithm are adopted. The results show that the former can successfully avoid the appearance of n in public key and private key, preventing the attacter from using the method of mathematical factorization to get the n factors and deducing the decryption key d. Adopting the latter, even if the attacker knows the $\varphi(n)$, but it doesn't have a specific formula to break down the three prime number for decomposition of n. So the difficulty of factoring increases and the computation is reduced due to the reduction of large prime numbers's bits. The experimental results indicate that the method increases the security of the RSA algorithm and its time complexity is same to the traditional algorithm.

Key words: RSA; security; public key; private key; time

0 引 言

在 RSA 算法中两个密钥之间存在一种数学上的联系。基于这个事实,如果某个人发现了这种数学联系并成功获取了私钥,这样体制就会被破坏。算法中公钥和私钥都包含大数 n,n 可以被分解为因子 p 和 q。众所周知,公钥是公开的,因此,如果某人猜到了 n 的因子那就很容易获取私钥。为了阻止该事件的发生,算法中尝试消除私钥和公钥中 n 的分布,在 n 上运用一种数学变换,通过使用 X 来替代 n,使得攻击者无法分解到 n 的因子。文中同时提出了三因子的 RSA 加密算法,指的是选取三个素数因子 a_1、a_2、a_3,它们的乘积为 n,即 $n = a_1 \times a_2 \times a_3$。该算法可以降低大素数选择所需要的时间,适当地减少大素数的位数,降低计算量。

1 传统 RSA 密码算法的研究与分析

RSA 算法[1] 是 1978 年由三位数学家 Rivest、Shamir 和 Adleman 根据 Whitfield 与 Martin Hellman 的理论框架设计出的一种非对称加密算法。RSA 密码体制的安全性取决于其加密算法的数学函数的求逆困难性,称之为大数因子分解的困难性[2-4]。RSA 包含了产生密钥、加密信息和解密信息三个步骤:

步骤 1:产生密钥。

选取两个大素数 p 和 q,$p \neq q$。计算乘积:

收稿日期:2015-11-23 **修回日期**:2016-03-04 **网络出版时间**:2016-08-01
基金项目:国家自然科学基金资助项目(11501302)
作者简介:陈春玲(1961-),男,教授,硕士,研究生导师,研究方向为软件工程、分布式组件技术、网络信息安全及其应用;齐年强(1991-),男,硕士研究生,研究方向为网络安全。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20160801.0904.026.html

$$n = p \times q \tag{1}$$

得到欧拉函数:

$$\varphi(n) = (p-1) \times (q-1) \tag{2}$$

选择随机整数 e,使得 $\mathrm{GCD}(e,\varphi(n)) = 1$ 且 $1 < e < \varphi(n)$。

计算:

$$d = e^{-1} \mathrm{Mod} \varphi(n) \tag{3}$$

其中,d 为 e 的关于模 $\varphi(n)$ 的乘法逆元,满足 $e \times d = 1 \mathrm{Mod} \varphi(n)$。

公钥为(e , n),私钥为(d , n)。

步骤 2:加密信息。

发送者通过式(4)来加密信息 M。

$$C = M^e \mathrm{Mod}(n) \tag{4}$$

其中,C 是加密后产生的密文。

步骤 3:解密信息。

接收者通过式(5)来解密密文 C。

$$M = C^d \mathrm{Mod}(n) \tag{5}$$

其中,M 是加密前的信息。

传统 RSA 算法密钥生成过程时间主要是求取最大公约数的时间及计算公钥和私钥模乘法逆元的时间[5-6]。求取最大公约数采用欧几里得算法,其时间复杂度为 $O(\varphi(n)/2)$,模乘法逆元的计算方法采用穷举法,时间复杂度为 $O(\varphi(n))$,因此密钥生成的时间复杂度为 $O(3\varphi(n)/2)$。加密解密计算的时间主要是模幂运算的时间,即形式为 $X^n \mathrm{Mod}(m)$ 的函数运算时间,其采用递归法实现,时间复杂度为 $O(n)$。所以传统 RSA 算法密钥生成的时间复杂度为 $O(\varphi(n)/2)$,加密算法时间复杂度为 $O(e)$,解密算法时间复杂度为 $O(d)$。

在传统的 RSA 公钥密码体制中一共出现了六个变量:p、q、n、$\varphi(n)$、e、d。但是在加密端只能知道 n 和 e,在解密端可以知道 p、q、n 和 d。其中可以将密文解密成明文的密钥是 d,即解密密钥。如果 d 被泄露了,那么加密是无效的。

在加密端,在已知 n 和 e 的情况下如何推导出解密密钥 d。

由式(3)知,推导 d 需要知道公钥 e 和 $\varphi(n)$。

由式(2)知,只有知道 p 和 q,才能算出 $\varphi(n)$。

由式(1)知,要想知道 p 和 q,必须对 n 进行大整数的因子分解。

所以在 RSA 密码体制中,如果模 n 被因子分解,那么其私钥 d 也会被泄露出去[7-9]。虽然至今还未能在理论和事件中证明有分解大整数的有效办法,但大数因子分解未被证明是 NP 问题,且随着计算机计算能力的提高,原来被认为不可能分解的某些大整数可能会被成功分解,这对 RSA 密码体制的安全性构成了潜在的威胁[10]。

为了确保 RSA 加密算法的安全性,只有不断地增加密钥长度。目前,一般要求密钥长度在 2 014 bit 位到 2 048 bit 位之间,甚至更高。

2　RSA 算法的改进

改进的 RSA 算法引进了对三个素数因子的 RSA 加密算法[11-12]和两个以上的步骤来消除密钥中的 n。引进的三个素数因子虽然增加了素数因子的个数,但是减少了素数因子的位数,消除密钥中的 n 可以使攻击者无法通过因式分解 n 分解得到因子 p 和 q。所以,一定程度上使得 RSA 算法更加安全。算法包括三个部分:产生内部密钥(n 已经被消除)、加密信息和解密信息。

步骤 1:产生密钥。

选择大素数 a_1、a_2、a_3,$a_1 \neq a_2 \neq a_3$。计算乘积:

$$n = a_1 \times a_2 \times a_3 \tag{6}$$

得到欧拉函数:

$$\varphi(n) = (a_1-1) \times (a_2-1) \times (a_3-1) \tag{7}$$

选择随机整数 k_1,使得 $\mathrm{GCD}(k_1,\varphi(n)) = 1$ 且 $\lfloor \sqrt{n} \rfloor \leqslant k_1 \leqslant \varphi(n)$。

根据 a_1、a_2、a_3 的大小关系,计算 X 来替代 n:

如果 $a_1 > a_2 > a_3$ 或者 $a_1 > a_3 > a_2$,解出 X 得:

$$\mathrm{GCD}(X,n) = 1, n - a_1 < X < n \tag{8}$$

如果 $a_2 > a_1 > a_3$ 或者 $a_2 > a_3 > a_1$,解出 X 得:

$$\mathrm{GCD}(X,n) = 1, n - a_2 < X < n \tag{9}$$

如果 $a_3 > a_1 > a_2$ 或者 $a_3 > a_2 > a_1$,解出 X 得:

$$\mathrm{GCD}(X,n) = 1, n - a_3 < X < n \tag{10}$$

将求得的 X 代入式(11),求解出 k_2:

$$k_2 = k_1^{-1} \mathrm{Mod}(X) \tag{11}$$

现在,公钥是(k_1, X),私钥是(k_2, X)。

步骤 2:加密信息。

发送者通过公钥(k_1, X)来加密信息 M,公式为:

$$C = M^{k_1} \mathrm{Mod}(X) \tag{12}$$

其中,C 是加密后产生的密文。

步骤 3:解密信息。

接收者通过私钥(k_2, X)来解密密文 C,公式为:

$$M = \sqrt{C^{k_2} \mathrm{Mod}(X)} \tag{13}$$

其中,M 是加密前的信息。

根据上述步骤,可以设计出改进 RSA 算法密钥生成的流程,见图 1。

通过用 X 替换 n 成功消除了 n。这让算法变得相对安全,因为攻击者无法通过 X 分解到 a_1、a_2 和 a_3。

RSA 算法的约束是,加密时所取的信息值必须小于大数的值,即小于 n 的值[13]。

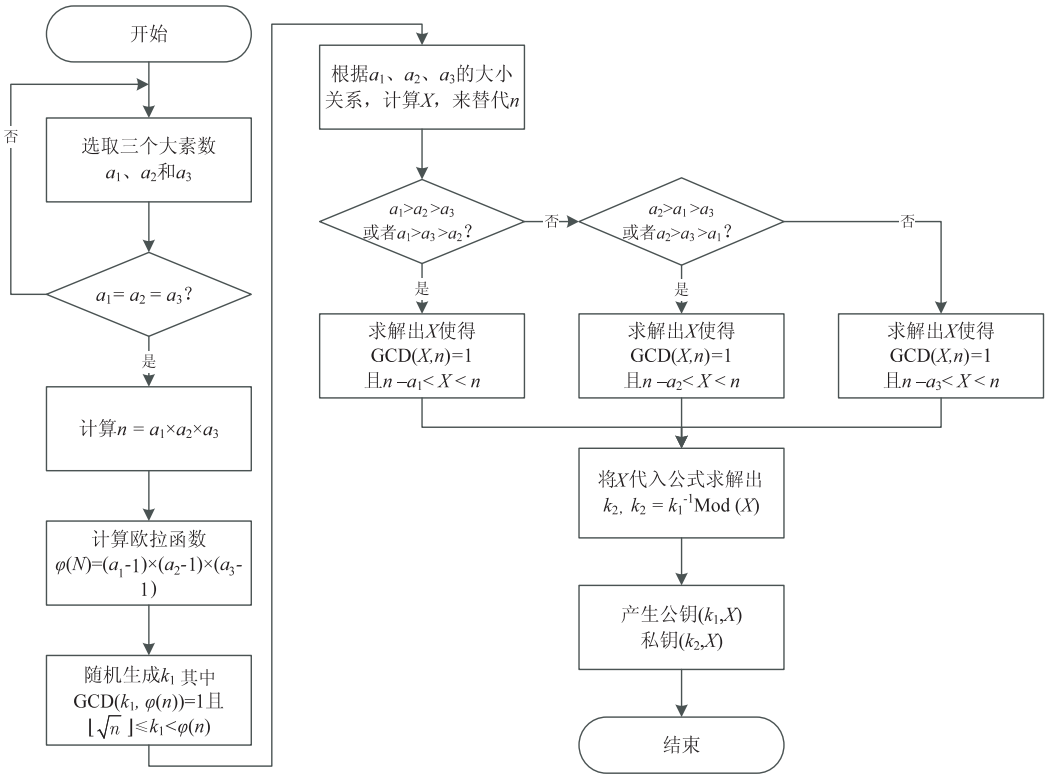

图 1 改进 RSA 密钥生成算法流程图

再者,能够通过另外一层的加密扩展该算法,但是这种大于一层的加密只能运用于二因子的 RSA 算法,并不适用于三因子的 RSA 算法。这个可以在计算完 X 通过重复改进算法中的式(8)~(10)并且得到 k_2 来实现。

但是这一次需要采用获取答案的四次方根。这个方法可以应用更多次,只要取得的值满足所有的约束。

改进的 RSA 算法密钥生成以及加解密算法所采用的方法是一样的,但是步骤上有所不同。密钥生成算法增加了根据 a_1、a_2、a_3 的大小替换 n 的 X 的计算,时间复杂度为 $O(2\varphi(n))$,加密算法的时间复杂度为 $O(k_1)$;解密算法在模幂运算结果的基础上增加了平方根运算,因为平方根运算的时间复杂度为常数阶,因此解密算法的时间复杂度为 $O(k_2)$。

3 实验结果与分析

以计算 $\varphi(n)$ 的攻击方法为例,如果在传统的 RSA 加密算法中,攻击者可以计算出来 $\varphi(n)$,就能够分解出 n 的素数因子。这是由于 $\varphi(n)=(p-1)(q-1)=pq-(p+q)+1=n-(p+q)+1\rightarrow p+q=n+1-\varphi(n)$ 且 $pq=n$,那么可以构造一元二次方程 $x^2-(n+1-\varphi(n))x+n=0$,方程的根就是 p 和 q[14-15]。

例如:如果 $n=221$, $\varphi(n)=192$,那么,一元二次方程 $x^2-30x+221=0$ 的两个根为 $x_1=13$, $x_2=17$。因此, $n=13\times17=221$。

改进的 RSA 算法中,在已知公钥的前提下,无法知道公钥中的 n,因此无法通过方程分解出 n 的因子;再者,采用三个素数因子的 RSA 算法,就算攻击者知道了 $\varphi(n)$,但是对于分解 n 的三个素数因子没有一个具体的公式去求解,所以分解的困难度会大大增加,提高了算法的安全性。

表 1 和表 2 分别为 RSA 算法和改进 RSA 算法在时间上的差异。以滴答(1 ms 等于 10 000 滴答)为单位,并只用单层加密,即使用 X 作为系数。因为此算法专注于 RSA 算法的安全方面,密钥的生成以及加解密方面增加了计算量,所以时间会有一些增加。

表 1 RSA 算法的时间

p	q	信息	密钥的产生时间	加密解密时间	总时间
5	7	2	17 782	23 279	41 061
7	11	3	19 460	24 702	44 162
11	13	5	16 923	21 914	38 837

表 2 改进 RSA 算法的时间

a_1	a_2	a_3	信息	密钥的产生时间	加密解密时间	总时间
2	5	7	2	20 735	26 894	47 599
5	7	11	3	19 847	25 784	45 631
7	11	13	5	24 732	29 284	54 016

图 2 是 RSA 算法和改进 RSA 算法之间产生密钥的时间对比,以滴答为单位。

密钥的产生过程在时间上有很多的变化,因为多余的一些加密步骤或消除 n 的分布消耗了时间。

图 3 为 RSA 算法和改进 RSA 算法之间加密和解

密的时间对比,以滴答为单位。

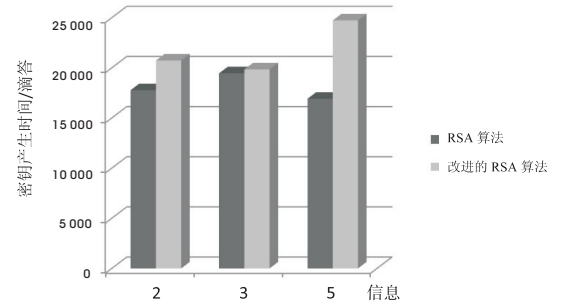

图 2　RSA 算法和改进 RSA 算法之间密钥产生的时间对比

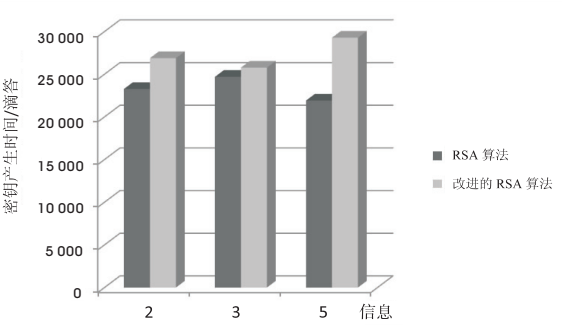

图 3　RSA 算法和改进 RSA 算法之间加密和解密的时间对比

信息的加密和解密存在时间上的差异,因为解密算法发生了改变。后者更加复杂并会消耗更多时间。

图 4 为 RSA 算法和改进 RSA 算法之间的总体时间对比,以滴答为单位。

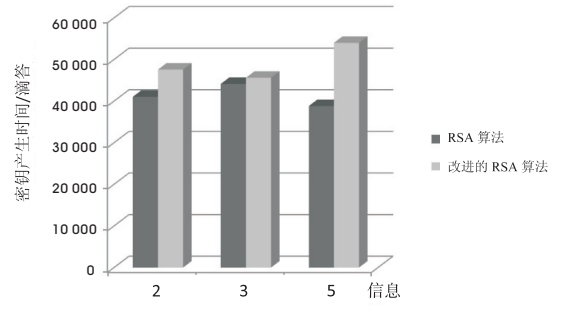

图 4　RSA 算法和改进 RSA 算法之间的总时间对比

根据上述对传统的 RSA 算法和改进算法的时间复杂度的分析,两种算法的时间复杂度级别相同,又因为增加的时间并不影响密钥生成以及加解密的过程,消耗的时间并不是很高,因此可以忍受。

4　结束语

由于改进 RSA 算法克服了传统 RSA 算法因素数选择问题位数要求越来越多使得素数产生效率下降、n 被因子分解密码体制安全性下降等缺点[16],所以,改进 RSA 算法具有独特的优势,实现了对信息的安全加密。由于在最初的 RSA 算法中加入三个素数因子并且消除 n,用来替代 n 最新生成的数不仅可以在公钥和私钥中使用,而且还能成功避免受到因子分解的攻击,使得 RSA 算法更加安全,但是在时间上会有一些增加,因此还需要对密钥生成、信息的加解密速度和时间进行深入研究。

参考文献:

[1] Diffie W, Hellman M E. New directions in cryptography[J]. IEEE Transactions on Information Theory, 1976, 22(6): 644-654.

[2] 胡　云. RSA 算法研究与实现[D]. 北京:北京邮电大学, 2010.

[3] Aboud S J. An efficient method for attack RSA scheme[C]// Proc of second international conference on applications of digital information and web technologies. [s. l.]: IEEE, 2009: 587-591.

[4] Mehrotra V. An effective method for attack RSA strategy[J]. International Journal on Advanced Networking and Applications, 2012, 3(5): 1362-1366.

[5] RSA Laboratory. RSA algorithm time complexity[EB/OL]. 2009. http://www.rsa.com/rsalabs/node.aspid=2215.

[6] 靳丽君. 非对称加密体制中 RSA 算法的研究[J]. 电子设计工程, 2011, 19(11): 29-30.

[7] 李　青, 李雄伟, 金　涛. RSA 算法的研究与简单实现[J]. 网络安全技术与应用, 2007(6): 88-91.

[8] Jonsson J, Kaliski B. The public-key cryptography standards [J]. RSA Data Security, 1993, 8(5): 33-35.

[9] Ambedkar B R, Gupta A, Gautam P, et al. An efficient method to factorize the RSA public key encryption[C]//Proc of international conference on communication systems and network technologies. [s. l.]: [s. n.], 2011.

[10] 鄢喜爱, 杨金民, 田　华. RSA 公钥密码算法的分析[J]. 长春工业大学学报:自然科学版, 2006, 27(2): 142-144.

[11] 安吉旺, 徐凯宏. 基于 RSA 和密钥的二维码加密编码的研究[J]. 森林工程, 2014, 30(2): 125-129.

[12] 谢仁康. 非对称加密二维码防伪系统的设计[D]. 成都:电子科技大学, 2013.

[13] Milanov E. The RSA algorithm, University of Washington: Department of Mathematics[EB/OL]. 2013-10-08. http://www.math.washington.edu/~morrow/336_09/papers/Yevgen y.pdf.

[14] Dhakar R S, Gupta A K, Sharma P. Modified RSA Encryption Algorithm (MREA)[C]//Proc of ACCT. [s. l.]: [s. n.], 2012.

[15] Sarkar S, Maitra S. Cryptanalysis of RSA with two decryption exponents[J]. Information Processing Letters, 2010, 110(5): 178-181.

[16] 刘项洋. 基于 RSA 的随机密钥交换系统的研究与设计[D]. 合肥:合肥工业大学, 2004.