现代密码学
Modern Cryptography

张方国

中山大学计算机学院

Office: Room A305, IM School Building

E-mail: isszhfg@mail.sysu.edu.cn

HomePage: http://cse.sysu.edu.cn/content/2460

第十五讲 RSA(二)

- RSA攻击：分解因子算法
- 对RSA的其他攻击

5.6 分解因子算法

Pollard ρ-1算法

算法 5.8 Pollard $p-1$ Factoring Algorithm(n, B)

$a \leftarrow 2$

for $j \leftarrow 2$ **to** B

  do $a \leftarrow a^j \bmod n$

$d \leftarrow \gcd(a-1, n)$

if $1 < d < n$

  then return (d)

  else return("failure")

Pollard ρ算法

算法 5.9 Pollard ρ Factoring Algorithm(n, x_1)

external f

$x \leftarrow x_1$

$x' \leftarrow f(x) \bmod n$

$p \leftarrow \gcd(x - x', n)$

while $p = 1$

do {
- **comment**：在第 i 次反复中，$x = x_i$ 且 $x' = x_{2i}$
- $x \leftarrow f(x) \bmod n$
- $x' \leftarrow f(x') \bmod n$
- $x' \leftarrow f(x') \bmod n$
- $p \leftarrow \gcd(x - x', n)$

if $p = n$

  then return ("failure")

  else return (p)

Dixon的随机平方算法

许多分解因子算法的理论基于下列的简单事实。假定我们可以找到 $x \not\equiv \pm y \pmod n$ 使得 $x^2 \equiv y^2 \pmod n$，那么

$$n \mid (x-y)(x+y)$$

但 $x+y$ 或者 $x-y$ 都不能被 n 整除。因此 $\gcd(x+y, n)$ 是 n 的一个非平凡因子(类似地，$\gcd(x-y, n)$ 也是 n 的一个非平凡因子)。

作为一个例子，容易验证 $10^2 \equiv 32^2 \pmod{77}$。通过计算 $\gcd(10+32, 77)=7$，我们发现了 77 的一个因子 7。

随机平方算法使用一个因子基(factor base)，$\mathcal{B}$ 因子基是 b 个最小素数的集合(适当选取 b)。我们首先得到几个整数 z，使得 $z^2 \bmod n$ 的所有素因子都在因子基 $\mathcal{B}$ 中(如何做到这一点将在稍后讨论)。然后，将某些 z 相乘使得每一个在因子基中的素数出现偶数次。这样我们就建立起了一个所期望的类型的同余方程 $x^2 \equiv y^2 \pmod n$，该方程可能导出 n 的一个分解。

实际中的因子分解

RSA-129： Rivest等最初悬赏$100的RSA-129，已由包括五大洲43个国家600多人参加，用1600台机子同时产生820条指令数据，通过Internet网，耗时8个月，于1994年4月2日利用二次筛法分解出为64位和65位的两个因子，原来估计要用4亿亿年。所给密文的译文为“这些魔文是容易受惊的鱼鹰”。

- RSA-130于1996年4月10日利用数域筛法分解出来。
- RSA-140（465-bit）于1999年2月分解出来（185 networked computers）。
- RSA-154(512bit)于1999年8月分解出来。
- RSA-160, 2003.01 (http://www.loria.fr/~zimmerma/records/rsa160)
- RSA-174(576bit), 2003. 12
- RSA-194(640-Bit), 2005. 11

RSA-768bits： RSA-768 has 768 bits (232 decimal digits), and was factored on December 12, 2009 by Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Pierrick Gaudry, Alexander Kruppa, Peter Montgomery, Joppe W. Bos, Dag Arne Osvik, Herman te Riele, Andrey Timofeev, and Paul Zimmermann

- http://eprint.iacr.org/2010/006

RSA-768 =
1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413

RSA-768 =
33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489
×
36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917

5.7 对RSA的其他攻击

分解n是攻击RSA的一种方法，该方法是不是唯一的？

常见的三种：

（1）计算$\phi(n)$;

（2）解密指数；

（3）$Wiener$的低解密指数攻击

计算$\phi(n)$并不比计算n容易。

因为知道$\phi(n)$就能计算出n的两个素因子p和q

例5.13: 假定$n=84773093$，且敌手得到$\phi(n)=84754668$，则通过解方程得到两个素因子9539和8887

解密指数

如果解密指数a已知，那么n可以通过一个随机算法在多项式时间内分解。因此可以说计算a并不比分解n容易。

这个结论告诉我们，一旦泄露a，重新选择加密指数是不够的，必须选择一个新的n。

解密指数

先描述一个$Las\quad Vegas$型的随机算法。该算法具有最坏情形成功的概率至少为$1-\varepsilon$。故算法没有答案的概率至多为ε。

如果有这样一个$Las\quad Vegas$算法，那么只需多次运行，只至找到一个答案为止。算法运行m次没有得到答案的概率为ε^m。为了得到答案的平均运算次数为$1/(1-\varepsilon)$

解密指数

描述一个对于给定a,b和n作为输入，以至少$1/2$的概率分解n的$Las\quad Vegas$算法。故如果算法运行m次，那么n被分解的概率至少为$1-1/2^m$

算法基于当$n = pq$是两个素数的乘积时，$x^2 \equiv 1(\bmod n)$有4个平方根，其中两个平凡根，两个非平凡根(模n的互为相反数)

解密指数

例5.14：假定$n = 403 = 13 \times 31$。1模403的四个平方根为1, 92, 311, 402。

假定x是1模n的非平凡平方根。那么

$x^2 \equiv 1^2 \pmod{n}$但$x \neq \pm 1 \pmod{n}$

那么在随机平方分解算法中，我们可以通过计算$\gcd(x+1, n)$和$\gcd(x-1, n)$来分解

解密指数

算法5.10通过寻找1模n的非平凡平方根来分解n。先看一个例子

假定$n = 89855713, b = 34986517,$且$a = 82330933,$

取随机数$w = 5,$ 我们有

$$ab - 1 = 2^3 \times 3600590733787795(\text{即}r)$$

则$w^r \bmod n = 85877701$和$85877701^2 \equiv 1 (\bmod n)$

因此算法将返回值$x = \gcd(85877702, n) = 9103$

这是n的一个因子，另一个因子为$n / 9103 = 9871$

解密指数

算法5.10 $RSA-FACTOR(n,a,b)$

$Comment:$假定$ab\equiv 1(\bmod \phi(n))$

记$ab-1=2^s r$，，为奇数

随机选择w使得$1\le w\le n-1$

$x\leftarrow \gcd(w,n)$

$if \quad 1<x<n$

$\quad then \quad return(x)$

$Comment:x$是n的一个因子

$v\leftarrow w^r \bmod n$

$if \quad v\equiv 1(\bmod n)$

$\quad then \quad return("failure")$

$while \quad v\not\equiv 1(\bmod n)$

$$do\begin{cases} v_0\leftarrow v \\ v\leftarrow v_0^2 \bmod n \end{cases}$$

$if \quad v_0\equiv -1(\bmod n)$

$\quad then \quad return("failure")$

$$else\begin{cases} x\leftarrow \gcd(v_0+1,n) \\ return(x) \end{cases}$$

$Comment:x$是n的一个因子

解密指数

在该算法中，如果w为p或q的倍数，那么可以直接分解n。如果$\gcd(w,n)=1$，那么可通过连续平方计算$w^r, w^{2r}, w^{4r}, \ldots,$直到对某个$t$，有$w^{2^t r} \equiv 1 \pmod n$(其中$ab-1=2^s r \equiv 0$)

解密指数

因为$w^{2^s r}\equiv 1\pmod{n}$.因此,$while$循环至多运行$s$次就会终止.在$while$循环结束时,我们找到一个值$v_0$,使得$(v_0)^2\equiv 1\pmod{n}$,但是$v_0\not\equiv 1\pmod{n}$.如果$v_0\equiv -1\pmod{n}$,那么算法失败;否则,$v_0$是1模$n$的一个非平方根,我们能分解$n$.

该算法的成功概率至少为$1/2$

$Wiener$的低解密指数攻击

$Wiener$提出的一种攻击：

如果解密指数a满足：$3a < n^{1/4}$且$q < p < 2q$

那么可以成功地计算a。

其他攻击方法：

习题5.14,5.17的方法。

以下情况的RSA是不安全的

- 模数n的两个素因子相差太大或太小
- N=pq, p-1或q-1没有大素因子
- 低解密指数和低加密指数
- p-1和q-1有大公因子