第 35 卷 第 2 期 计 算 机 工 程 2009 年 1 月
Vol.35 No.2 Computer Engineering January 2009
·安全技术· 文章编号：1000—3428(2009)02—0123—03 文献标识码：A 中图分类号：TN918

RSA 公钥密码算法的计时攻击与防御

陈财森，王 韬，郑媛媛，赵新杰
(军械工程学院计算机工程系，石家庄 050003)

摘 要：计时攻击根据密码算法在密码设备中运行时的执行时间差异，分析和判断密码算法的各种有效信息，是最具威胁的旁路攻击方式之一。该文研究 RSA 加密算法和计时攻击的原理，分析 RSA 解密过程，阐述针对基于模幂算法的 RSA 计时攻击的原理，讨论如何抵御该计时攻击。

关键词：RSA 公钥密码算法；计时攻击；模幂运算；RSA 隐蔽

Timing Attacks and Defenses on RSA Public-key Algorithms

CHEN Cai-sen, WANG Tao, ZHENG Yuan-yuan, ZHAO Xin-jie
(Dept. of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003)

【Abstract】 Timing attacks are used to collect and analyze the valuable information of algorithms from the different amounts of time, which are taken when cryptographic devices are working, and they are the most effective side channel attacks. This paper analyzes the process of the RSA decryption algorithm on the research in RSA encryption algorithm and timing attacks, illustrates the theory of timing attack to RSA bases on the modular exponentiation and the square and multiply algorithm, and discusses how to defense timing attack on RSA.

【Key words】 RSA public-key algorithms; timing attacks; modular exponentiation; RSA blinding

1 概述

公开密钥密码是现代密码学中最重要的研究内容之一。在迄今所有的公钥密码体制中，RSA算法是第 1 个能同时用于加密和数字签名的算法，并广泛应用于保护数据传送的公钥算法。在硬件上，安全电话、以太网，尤其是智能IC卡，都采用了RSA技术；在软件上，Microsoft 和 Netscape的浏览器、提供Internet安全秘密保障的安全套接层(Secure Sockets Layer, SSL)[1]、S/WAN和电子邮件安全协议PGP (Pretty Good Privacy)等也都引入了RSA加密算法。

RSA的公钥中包含一个数N，由 2 个大素数p, q相乘而得。RSA的强度取决于分解一个大数的难度。例如，在一次RSA算法挑战中(1999 年 8 月)，一个 512 bit的RSA密钥N通过 292 台高速主机进行因式分解，整个过程花了 35.7 个CUP年，即以每秒运行百万条指令的速度需要耗时 80 000 MIPS年(MIPS年是指一台每秒执行百万条指令的处理器运行一年，即执行约 3×10^{13}条指令)，大约需要花费 3.7 个月[2]。或许可以假设RSA在密钥N为 1 024 bit时对于因式分解攻击是安全的。RSA可以通过增加密钥的长度至 2 048 bit或更多来提高它的安全度。

传统的攻击思想一直认为RSA拥有强大而精确的加密强度，攻击者仅能获取输入信息和输出信息，而其他有关密钥的信息是无法轻易获取的，但是Kocher提出了旁路攻击的概念[2]，即密码设备在运算和工作时会通过各种隐通道泄露时间消耗、声波、电磁辐射、功率消耗等信息，旁路攻击就是将这些信息进行收集、分析，并从中萃取出与密码操作相关的信息，进一步实现密码解密。计时攻击归类于旁路攻击，由于它不需要像其他旁路攻击方式需要特殊的设备和物理访问机器，因此以操作简便、排除候选密钥精确有效而著称。本文主要研究计时攻击如何获取RSA的密钥，从而实现对RSA加密系统的攻击，并讨论了RSA加密算法应该如何抵御计时攻击。

2 计时攻击

计时攻击[3]是一种利用密码设备工作运行时的时间特征推导出私钥的攻击方法。文献[3-4]证明，攻击者可以通过记录计算机解密消息所用的时间确定私钥。计时攻击不仅可用于攻击RSA，也可以用于攻击其他公钥密码系统。由于这种攻击的完全不可预知性以及它仅依赖于密文，因此具有很大的威胁。

计时攻击类似于窃贼通过观察他人保险柜拨号盘的时间长短来猜测密码。它的攻击原理如图 1 所示。

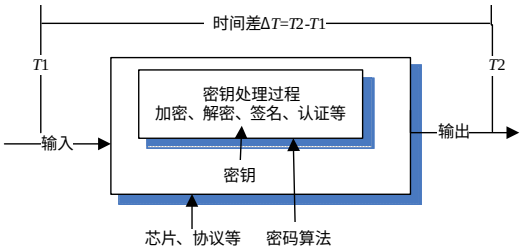

图 1 计时攻击原理

由于密码算法的执行时间会因为输入参数的不同而不同，因此通过精确记录密码设备密钥操作时间，分析其执行一组操作所花费的时间ΔT，再应用统计学的方法推导出加密

基金项目：国家自然科学基金资助项目“基于 Cache 的远程计时攻击研究”(60772082)；军械工程学院科学研究基金资助项目
作者简介：陈财森(1983－)，男，硕士研究生，主研方向：信息安全，网络对抗；王 韬，教授、博士生导师；郑媛媛，博士研究生；赵新杰，硕士研究生
收稿日期：2008-07-10 **E-mail：**caisenchen@163.com

系统在计算中涉及的密钥。

3 针对 RSA 的计时攻击

3.1 RSA 加密算法

RSA是一个使用公钥(N, e)加密、使用一个私有的指数d解密的公钥加密系统。系数N是 2 个大素数p, q的乘积；指数e和d必须满足e^d=1mod (p-1)(q-1)。RSA 密钥是由公钥(N,e)和私钥d组成的。例如：选择 2 个素数p=11, q=3，那么N=p×q=33。计算(p-1)(q-1)=10×2=20，再选择一个相对于 20 的素数 3。通过ed=1 mod 20，由扩展的欧几里德算法得出d的一个可能值为 7，因为 3×7=21=1 mod 20。因此，得到公钥(N=33, e=3)及相应的私钥d=7。丢弃原始的p和q。因式分解破解RSA时，攻击者需要把N分解成为p和q，再利用公钥e，根据ed=1 mod (p-1)(q-1)，可以很容易地找到私钥d。

为了加密一段明文M，计算$C=M^e \bmod N$，其中，C是密文。计算$M=C^d \bmod N$可以破解密文C，由此产生原始的信息M。继续使用之前的例子，公钥为(33, 3)，私钥d=7。假设要发送信息M=19。加密产生一个编码信息$C=M^e \bmod N$ = 193 mod 33=28。发送者发送密文C=28。为破解密文C，接收者使用私钥d并计算$M=C^d \bmod N$=287 mod 33=19，这就是原始的信息 19。这个密码系统由于Euler Theorem的数字理论而得到推广[5]。RSA的另一个应用是数字签名，一种用来证明信息来源的方法。签名利用私钥d以同样的解密方法进行解密，接收者利用公钥e执行加密操作以验证签名。

3.2 RSA 计时攻击的实现原理

RSA中的运算操作包括模幂运算$M=C^d \bmod N$，其中，N是RSA的一个系数；C是密文或者是签名；d是私钥。攻击者的目标就是获取d。对于计时攻击，攻击者需要目标系统多次计算精心选择的C的$C^d \bmod N$。通过测量请求和分析的时间变量总数，攻击者可以一次恢复私钥d的一个位直至整个私钥都被发现。这样的攻击本质上是一个信号探测问题，该信号由目标指数位产生的时间变量组成[3]。因为私钥d的位数是有限的，所以计时攻击从计算上是可行的。

RSA的解密过程就是求出$M=C^d \bmod N$的结果，在二进制中，$d=d_0d_1\cdots d_n$, d_0=1。这里可以将C^d化为$x\times y$，对于正整数x, y和n，要计算$xy \bmod n$，采用传统的方法是先计算xy，再将结果用n去除，求出非负整数$r<n$，满足$xy=nq+r$。这种算法很费时且没必要，因为计算出xy不仅要浪费时间和存储空间，而且结果中只有r有价值，商q并无用处，反而大大增加了无用的开销，所以现在大多采用更为快捷的模幂算法，算法如下：

输入 正整数n，小于n的整数C，整数$d=(d_{t-1}d_{t-2}\cdots d_1d_0)2$

输出 $C^d \bmod N$

```
s = C,K=0
for i = t-1 to 0
{K=2*K;
S = mod(S^2, N)
}
if dj == 1 then
{K=K+1;
 S = mod(SC, N)
}
end if
      next i
     return s
```

计算流程如图 3 所示。

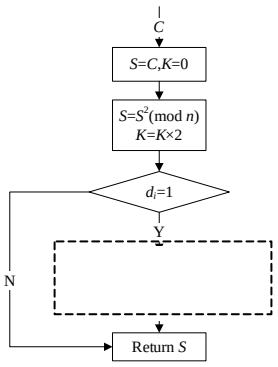

图 3 模幂算法流程

由图 3 可明显看出，当指数d的位为 1 时，整个运算过程多了一个乘法操作，即先平方后乘法操作；而当d的位为 0 时，仅有平方操作。又因为在硬件操作时，乘法操作需要附加的寄存器参与，所以比平方操作耗时长[6]。如计算$M=C^d \bmod N$，C=5, d=18, n=33，d转换为二进制为 10010。这里可以把 18 从左至右逐位排列为(0, 1, 10, 100, 1001, 10010) = (0, 1, 2, 4, 9, 18)，即 18 可以由一系列的数构成，对每个数进行左移运算，如果下一个二进制位是 1，则再加 1。建立 18 的步骤如下：

1 = 0×2 + 1
2 = 1×2
4 = 2×2
9 = 4×2+1
18 = 9×2

现在假设要计算 $5^{18} \bmod 33$，一个有效的方法是通过对数字有效的乘方运算再乘以对数字加 1 的数，这就是模乘算法的原理，操作步骤如下：

$5^1 = (5^0)^2\times5^1 = 5 \bmod 33$
$5^2 = (5^1)^2 = 5^2 = 25 \bmod 33$
$5^4 = (5^2)^2 = 25^2 = 625 = 31 \bmod 33$
$5^9 = (5^4)^2\times5^1 = 31^2\times5^1 = 4\ 805 = 20 \bmod 33$
$5^{18} = (5^9)^2 = 20^2 = 4 \bmod 33$

计算过程数据如下，其中，只是为了便于解释算法，引入变量 K，K 的终值等于幂指数的值：

i	d	K	S
4	1	1	5
3	0	2	25
2	0	4	31
1	1	9	20
0	0	18	4

根据上面的分析，需要在解密运行过程中判断执行时间的差异来确定d_i为 1 或者 0，所以，还需要用到统计学的方法，这样在计时攻击执行的过程，就需要大量的样本数据。研究人员能够在 2 h之内从一个RSA加密服务程序中解析出 1 024 bit的RSA私钥。攻击需要大约 350 000 个样本[7]。在攻击过程中，可以选择一系列的密文C，执行攻击并记录下时间的变化，这里需要一个e值(e的值是根据经验确定的)，通过样本分析得出，如果记录的时间超过e，表示d_i的值为 1，否则为 0，然后通过求d_i的期望值来统计分析d_i是 1 还是 0，这样通过大量的样本分析，最终逐位地获取整个私钥d。

4 计时攻击的防范

目前有几种针对计时攻击的防范措施，其中应用最广泛的是RSA隐蔽。RSA隐蔽是在RSA计算中采用随机数使得计时信息变得不可能。在解密密文C之前，首先计算$C'=r^eC \bmod$

N，其中，r是一个随机产生的数；e是公开指数。与通常的RSA运算一样，先计算$C'^d \bmod N = r^{ed}\, C^d \bmod N = rC^d \bmod N$，再使用欧拉定理[8]。然后把获得的值乘以$r^{-1}$，计算$r^{-1}\, r\, C^d \bmod N$以获取需要的明文。具体步骤如下：

(1)产生 0~n-1 之间的秘密的随机数 r。

(2)计算$C'= r^eC \bmod n$，其中，e是公开的指数。

(3)像通常的RSA运算一样，计算$M'=(C')^d \bmod n$。

(4)计算$M= M'r^{-1} \bmod n$，其中，r^{-1}是r的模n的乘法逆元，即$r^{-1}r=1 \bmod n$。

根据$r^{ed} \bmod n = r \bmod n$，可以证明结论是正确的。

因为不同的r作用于不同的信息，所以原始信息在求幂运算之前随随机数的改变而变化。这样，隐蔽能够预防攻击者通过请求执行解密操作，利用计时信息的结果获取密钥，但是隐蔽方法导致运算的性能降低了 2%~10%[9]。

另一种防范措施是使所有的私钥运算都不依赖于输入，保证所有的幂运算在返回结果前执行的时间都相同。比如在Montgomery算法中，即使中间结果没有用超过N的值，也总是执行额外的缩减运算。这种改进相对容易实现，但会降低算法的性能[10]。

还可以将所有的 RSA 计算量化，比如，使它们总是用若干预定的时间完成预运算。这种方法的最大缺点是所有运算时间必须取运算消耗时间最长的，无法使执行性能得到优化，攻击者还可以通过收集额外的观察数据抵消随机延时，仍然可能攻击成功。

5 结束语

基于模幂运算的 RSA 公钥密码系统对于计时攻击是脆弱的，如果包括密钥的幂运算操作能够被攻击者准确地计时，密钥就能够通过选择输入与密钥成比例的数并运用统计学的方法恢复。目前使用最广泛的针对计时攻击的抵御方法是 RSA 隐蔽方法，它使攻击者无法获取有差异的计时信息。计时攻击突破了传统的攻击方式，在信息安全中具有重要的意义，目前国内对计时攻击的研究还处于初步阶段，需要大量的实验和研究以寻找更优化的攻击方法和抵御措施。

参考文献

[1] 周玉洁, 冯登国. 公开密钥密码算法及其快速实现[M]. 北京: 国防工业出版社, 2002: 18-65.

[2] RSA Laboratories[Z]. [2007-12-11]. http://www.rsasecurity.com/rsalabs/node.asp?id=2098.

[3] Kocher P. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems[C]//Proc. of CRYPTOLOGY'96. Berlin, Germany: Springer-Verlag, 1996.

[4] Kaliski B. Timing Attacks on Cryptosystems[EB/OL]. RSA Laboratories. (1996-01-20). http://www.rsasecurity.com/rsa-labs.

[5] Burton D M. Elementary Number Theory[M]. 2nd ed. [S. l.]: Brown Publishers, 1989.

[6] Kaihara M E, Naofumi T. A Hardware Algorithm for Modular Multiplication/Division Based on the Extended Euclidean Algorithm[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2005, E88-A(12): 3610-3617.

[7] Dhem J F, Koeune F. A Practical Implementation of the Timing Attack[C]//Proc. of CARDIS'98. [S. l.]: Springer,1998.

[8] Brumley D, Bonoeh D. Remote Timing Attacks Are Practical[DB/OL]. [2008-05-10]. http://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf.

[9] Stallings W. 密码学编码学与网络安全——原理与实践[M]. 4 版. 孟庆树, 王丽娜, 傅建明, 等, 译. 北京: 电子工业出版社, 2006.

[10] Wing Wong. Timing Attacks on RSA: Revealing Your Secrets Through the Fourth Dimension[DB/OL]. [2008-04-10]. http://www.cs.sjsu.edu/faculty/stamp/students/article.html.

(上接第 117 页)

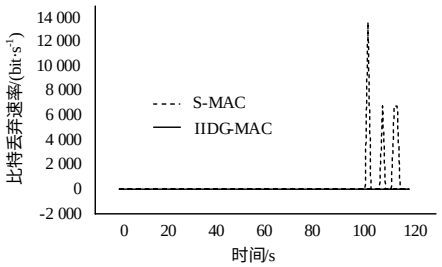

图 5 饱和系统比特丢弃速率

在饱和状态时，S-MAC性能较差的原因是：(1)在使用S-MAC时，节点每次成功发送后立即将其当前竞争窗口(CW_{curr})值降为CW_{min}，由式(7)可知CW_{min}并不适用于当前博弈状态(n)的均衡策略初始竞争窗口值；(2)节点每次竞争信道失败后，CW_{curr}以二进制指数退避机制增加，没有根据当前博弈状态(n)调整均衡策略。综上，在信道竞争较为激烈时，IIDG-MAC能快速将其初始竞争窗口调整为均衡策略，从而提高了数据帧的发送成功率，避免了过多碰撞，提升了各项性能指标。

6 结束语

本文综合考虑了博弈论和无线传感器网络的特点，针对S-MAC 中数据帧重传消耗的能耗较高等不足，提出基于纳什均衡的 IIDG-MAC 协议。仿真结果表明，与 S-MAC 相比，IIDG-MAC 能够将其每次竞争过程开始时的初始竞争窗口值调整到最佳，从而提高了数据帧的发送成功率，避免过多碰撞的发生，实现能量的节省和其他各项性能指标的提升。

参考文献

[1] Ye Wei, Heidemann J, Estrin D. An Energy-efficient MAC Protocol for Wireless Sensor Networks[C]//Proc. of INFOCOM'02. New York, USA: [s. n], 2002: 1567-1576.

[2] Van Dam T, Langondoen K. An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks[C]//Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. Los Angeles, USA: [s. n.], 2003.

[3] Bianchi G. Performance Analysis of the IEEE 802.11 Distributed Coordination Function[J]. IEEE Journal of SAC, 2000, 18(3): 535-547.

[4] Yong Kang, Xiu Ming, Yong Ren. Game Theory Models for IEEE 802.11 DCF in Wireless Ad Hoc Networks[J]. IEEE Communications Magazine, 2005, 43(3): 22-26.

[5] A Brief Tutorial on the PHY and MAC Layers of the IEEE 802.11b Standard[Z]. (2000-02-18). http://tinyurl.com/2d6f48.