RSA公钥密码计时攻击研究及仿真

田军舰,寇应展,陈财森
(军械工程学院 计算机工程系,河北 石家庄 050003)

摘 要:密码设备在执行加解密运算时泄露的时间信息能够被攻击者捕获,进而推算出密钥,破解密码系统。该文研究了RSA公钥密码算法和计时攻击的原理,分析了RSA加解密的模幂运算过程,阐述了基于模幂运算的RSA计时攻击原理,同时进行了仿真实现。仿真结果证明了RSA密码算法在计时攻击中存在安全缺陷,也说明了计时攻击与其他传统攻击相比更能准确快速地获得密钥。针对RSA公钥密码算法在计时攻击中存在的缺陷以及面临的安全威胁,讨论了抵御计时攻击的措施。

关键词:计时攻击;RSA;模幂运算;公钥密码
中图分类号:TP309.7 **文献标识码**:A **文章编号**:1673-629X(2010)08-0150-04

Research and Simulation of Timing Attacks on RSA

TIAN Jun-jian, KOU Ying-zhan, CHEN Cai-sen
(Dept. of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: Time information from cipher device to perform encryption/decryption could be captured by attacker. It is easy to conjecture the secret key and then break cryptosystem. Gives a research on RSA public-key algorithm and timing attacks against RSA, analyzes the process of modular exponentiation, and illustrates the theory of timing attack on modular exponentiation. At the same time, it is simulated and the result shows that RSA algorithm against timing attacks is vulnerable. Also it proposed that it can get the secret key faster and preciser in timing attack than other traditional attacks. In the end, discuss how to defense the timing attack using countermeasures.

Key words: timing attack; RSA; modular exponentiation; public cryptography

0 引 言

RSA是由三位MIT的学者Rivest、Shamir与Adleman于1978年提出的公钥密码算法。它是目前使用最为广泛的公钥密码算法之一,已成为公钥密码的国际标准。该算法的数学基础是数论中的Euler定理,其安全性建立在大整数因子分解的困难性之上。

RSA密码算法不但可以应用于加解密,而且还可以应用在数字签名、密钥交换等领域。对于使用正确、密钥强健的RSA来说,用传统的数学分析和穷举攻击进行破解几乎是不可能的。然而,近年来诞生了一类新的解密技术,称为旁路攻击[1](Side Channel Attacks)。旁路攻击主要针对硬件设备在进行密码运算时通过各种渠道(如声波、电磁辐射、功耗、时序等)泄漏的敏感信息,进行收集分析,并从中提取和密码操作相关的信息,推算出加密系统在计算中涉及到的秘密参量。计时攻击[2](Timing Attack)作为旁路攻击中的一种重要攻击方式,是根据密码算法在执行时表现出的时间差异,推断出相关的密钥信息。因为计时攻击不需要特殊的设备,实施简单,并且攻击成功率高,所以是最具威胁的一种攻击方式。

文中主要研究了针对RSA密码算法的计时攻击原理,设计了攻击过程,并且通过仿真实现成功获取了RSA的512位密钥,最后讨论了抵御这种攻击的措施。

1 计时攻击原理

计时攻击作为旁路攻击的一种,它试图攻击的是密码算法的实现而不是算法本身,只要密码系统在对不同密文执行解密时存在时间差异,其密钥就可能被计时攻击破解。

加密系统处理不同的输入信息所消耗的时间会有微小的差别,其中的原因包括[3]:对不必要的操作进行了性能优化、分支和条件语句、RAM Cache命中、运行

收稿日期:2009-12-10;**修回日期**:2010-02-09
基金项目:国家自然科学基金资助项目(60772082);军械工程学院科学研究基金(YJJXM07033)
作者简介:田军舰(1985-),男,河南南阳人,硕士研究生,研究方向为网络安全;寇应展,教授,硕士生导师,研究方向为网络安全。

于不固定时间的处理器指令(例如乘法和除法),以及其他种种原因。计时攻击正是通过分析密码设备已经执行的一组操作和这些操作所花费的时间,推导出秘密参量。它的攻击原理如图 1 所示。

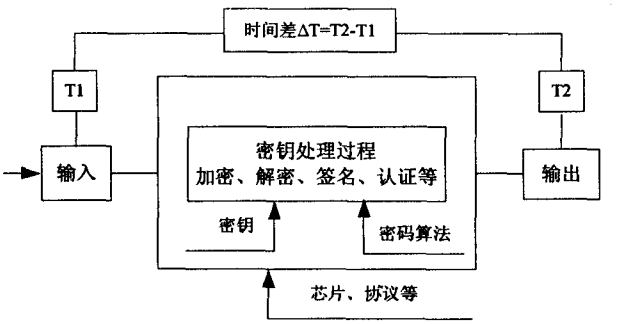

图 1　计时攻击原理

由于密码算法的执行时间会因为输入参数的不同而不同,因此可以通过精确记录密码设备与密钥操作有关的时间,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系统在计算中涉及的密钥。

2　针对 RSA 模幂运算的计时攻击

2.1　RSA 算法简介

RSA[4]算法为非对称密码算法,由于其加密和解密密钥是不同的,因此解决了对称密码体制所不能解决的密钥管理困难、陌生人之间通信保密和数字签名等问题。

RSA 算法的实现过程为:令 p 和 q 是随机选取的两个大素数(大约为十进制 100 位或者更大),$n = p * q$,n 是公开的,而 p 和 q 保密。随机选取一个数 e,e 为小于 $\varphi(n)$ 且与它互素的正整数,即满足 $\gcd(e, \varphi(n)) = 1$,其中 $\varphi(n) = (p-1)(q-1)$。利用辗转相除法得到整数 d 和 r,使得满足式(1):

$$ed + r\varphi(n) = 1, \text{即 } ed = 1(\mathrm{mod}\ \varphi(n)) \qquad (1)$$

这里 n、e 和 d 分别称为模数、加密密钥和解密密钥。$< n, e >$ 组成公钥,$< n, d >$ 组成私钥。例如:选择两个素数 $p = 13$,$q = 17$,那么 $n = p * q = 221$。计算 $(p-1)(q-1) = 12 * 16 = 192$。选择一个相对于 192 的素数 71 为 e 的值,通过 $ed = 1 \bmod 192$,可以得到 d 的一个可能值为 119。于是模数 n 为 221,公钥 e 为 71,私钥 d 为 119。

在对明文 M 进行加密时,首先将它分成比 n 小的数据分组(采用二进制,选取小于 n 的 2 的最大次幂),也就是说,如果 p 和 q 为 300 位的素数,那么 n 将有 600 位,每个消息分组 M_i 应小于 600 位长。加密后的密文 C,将有相同长度的分组 C_i 组成。加密公式表示为:

$$C = M^e \bmod n$$

解密为加密的逆过程,即

$$M = C^d \bmod n$$

由加解密过程可以看出,其中最主要的运算是模幂运算。表 1 总结了 RSA 加解密算法。

表 1　RSA 加密/解密算法

项目	满足条件
公钥	n:两个素数 p 和 q 的乘积(p,q 保密); e:与 $(p-1)(q-1)$ 互素
私钥	$d = e^{-1}(\mathrm{mod}(p-1)(q-1))$
加密	$C = M^e(\mathrm{mod}\ n)$
解密	$M = C^d(\mathrm{mod}\ n)$

2.2　针对 RSA 模幂运算的计时攻击原理

RSA 算法实现中,大数模幂运算是密码算法的核心运算。正是模幂运算泄露的时间信息给计时攻击提供了实际的可操作性。

RSA 的解密过程就是求出 $M = C^d \bmod N$ 的结果,而计时攻击的目标就是获取私钥。在二进制表示中,$d = d_0 d_1 \cdots d_n$,$d_0 = 1$。在实际的计算中,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 的积,再将结果除以 n,得到余数 $r(r < n)$,即 x、y、n、r 满足式子 $xy = nq + r$。显然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。所以在实际操作中采用更为快捷的"平方 - 乘法"算法,如算法 1 所示:

算法 1　"平方 - 乘法"算法

输入:C(密文对应的大数),N(模数),整数 $d = (d_{t-1}d_{t-2}\cdots d_1 d_0)_2$ 输出:$C^d \bmod N$
(1)$A \leftarrow C$; (2)对于 i 从 $t-2$ 递减到 0,执行: 1) $A \leftarrow A \cdot A \bmod N$; 2)若 $d_i = 1$,则 $A \leftarrow A \cdot C \bmod N$; (3)返回(A)

计算流程如图 2 所示。

由图 2 可明显看出,当指数 d 的二进制位为 1 时,整个运算过程多了一个乘法操作,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。根据以上分析,需要在密码算法解密执行过程中监测执行时间的差异来确定 d_i 为 1 或者 0。正是利用了 RSA 密码算法中模幂运算泄漏的时间信息,可以利用计时攻击来破解 RSA 的密钥。

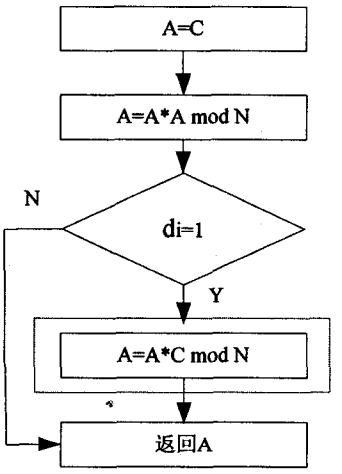

图 2 “平方－乘法”计算流程

3 RSA 计时攻击的仿真实现

3.1 仿真实现的关键技术

以上分析了针对 RSA 公钥密码算法中模幂运算计时攻击的原理，在我们的仿真实现中，仍然有以下几个关键技术问题要解决：

(1)高精度精确计时。计时攻击主要通过对执行时间的监测来推算密钥，由于差异值一般都很小，因此需要高精度精确计时技术。因此，利用 Pentium CPU 内部时间戳进行计时，它以 64 位无符号整型数的格式，记录了自 CPU 上电以来所经过的时钟周期数。由于目前的 CPU 主频都非常高，因此这个部件可以达到纳秒级的计时精度。在 Pentium 以上的 CPU 中，提供了一条机器指令 RDTSC[5] (Read Time Stamp Counter)，该指令将处理器的时间戳计数器当前值装入 EDX:EAX 寄存器对。由于 EDX:EAX 寄存器对恰好是 C 语言保存函数返回值的寄存器，所以编程实现中可以把这条指令看成是一个普通的函数调用。这种计时方法得到的数据抖动比较厉害，其实对任何计量手段而言，精度和稳定性永远是一对矛盾。如果用低精度的 C 语言内置函数(例如 timeGetTime)来计时，基本上每次计时的结果都是相同的；而 RDTSC 指令每次结果都不一样，经常有几百甚至上千的差距，但是可以通过增大样本量测量，降低数据的抖动性，提高测量精度。

(2) 大规模计时样本的捕获。为了有效降低计时数据的抖动，提高计算解密时间的精度，需要多次对解密操作进行计时，但一般来说要获取解密器的控制权并进行大量的解密操作不太可能，因此计时攻击通常只适用于解密器自动应答的情况，如 SSL 协议。

(3) 进程的干扰以及网络噪声。通常除了理想的解密算法本身产生的计时差别外，在系统中还会存在其他运行的进程和密码进程抢占 CPU 资源，另外比如 Cache 命中率、分时操作系统的时间片轮转、网络传输时延的不确定性等等，都会给解密计时精度产生影响。尽管通过大量取样平均化处理可能减少这些误差影响，但是这使得攻击所需要的样本量大大增加。在我们的仿真实现中，尽可能地减少其他进程，以便减少需要的样本。

(4)先验值的确定。不同的执行环境和设置参数，再加上其它噪声，都会影响每次实验的结果，而且先验值一旦前面的确定错误，会直接对后面的位信息的判断产生影响。在我们的仿真中，为了有效地适应攻击执行环境的变化，在攻击之前进行大量的实验，学习各种环境下的执行情况，作为下一步实验的依据。

3.2 仿真结果的分析

仿真实验环境为：Intel Core Duo CPU 2.4GHz，内存 1G，编程环境为 VC++ 6.0。文中进行了四次实验分别得到了密钥为不同长度时的执行时间的先验值，如表 2 所示。其中先验值的单位为时钟周期。

表 2 密钥为不同长度时的先验值

密钥位数	先验值
128	1500
256	3000
512	6000

在仿真平台上对使用 512 位密钥的 RSA 算法进行了计时攻击，计时攻击流程如图 3 所示。流程共分为四部分：先验值学习、产生密钥、加解密处理、采集时间数据进行密钥推算和验证。

(1)先验值学习。

在已知密钥位信息的情况下，执行解密和计时操作，记录密钥位分别为 1 和 0 时对应的时间范围 f_1 和 f_0，作为下一步推断密钥位的判断依据。对每种攻击对象的情况只需要进行一次先验值学习即可。

(2) 密钥的产生。

分为素数 p 和 q、模数 R 和 r、公钥 PK 和私钥 SK 的生成。

(3) 加解密处理。

运用 RSA 算法的公式 $E(M,e) = M^e \bmod n = C$，$D(C,d) = C^d \bmod n = M$，对输入明文 M 进行加密，对输出的密文 C 进行解密。

(4) 采集时间数据进行密钥推算和验证。

应用高精度精确计时技术，对 RSA 模幂运算采用从左到右二进制平方乘法运算步骤进行计时，并将计时信息保存；执行完毕后对保存的计时信息进行分析，依据计时数据的大小，设置判断值大小，一位一位地判

断私钥 d 的位,从而推算出整个私钥,最后进行密钥的验证,如果正确则攻击成功,否则攻击失败,记录破解密钥位的正确率,并调整判断值大小或者更改输入明文。

在实际的计时攻击中,测量解密过程中模幂运算的时间,并进行分析。文中针对采用 512 位密钥的 RSA 密码算法,对解密过程中与密钥有关的操作步骤执行时间数据进行采集,分析结果如图 4 所示。在图中纵轴时间线大于先验值(6000 个时钟周期)的对应密钥位为 1,时间线小于先验值(6000 个时钟周期)的对应密钥位为 0。图 4 中下面显示的"01"字符串是真实的密钥,与攻击所得时间线对应的密钥值完全一致,证明了仿真实验恢复的密钥是正确的。

计时攻击针对的是 RSA 密码算法的实现,绕开了分解大数问题的困难性,因此对 RSA 公钥密码算法的安全性有很大威胁。目前,有几种防范计时攻击的措施,其中应用最广泛的是盲化技术[6,7]。盲化技术使 RSA 在解密之前要计算 $x = r^e g \bmod N$,其中 r 是随机的,e 是解密指数,g 是要解密的密文,然后对 x 进行解密,最后把获得的值乘以 r^{-1},即计算 $r^{-1} r\, g^d \bmod N$ 获得明文。因为 r 是随机的,因此盲化可以有效抵御攻击者利用计时信息的结果获取密钥。另一种防范措施是使私钥参与的运算都不依赖于输入,保证在所有的幂运算返回结果前执行的时间相同。比如在采用蒙哥马利算法[8]的快速 RSA 实现中,即使中间结果不大于模数,也执行额外约简,这样就消除了因执行额外约简[9]而引起的时间差异,从而保护了私钥,这种方法容易实现,但是会造成算法效率的下降。

4　结束语

基于模幂运算的 RSA 公钥密码算法对于计时攻击是脆弱的,如果与密钥相关的幂运算操作能够被攻击者准确地计时,那么 RSA 算法的安全性将不复存在,目前最有效的防御方法是采用盲化技术。文中研究了针对 RSA 公钥密码的计时攻击原理,并进行了仿真。仿真结果表明,使用模幂运算的 RSA 算法给计时攻击提供了条件,可以在极短的时间内攻破 RSA 公钥密码算法的 512 位密钥。目前国内对公钥密码算法的计时攻击还处于初级阶段[10~12],因此尚有大量研究工作要做。

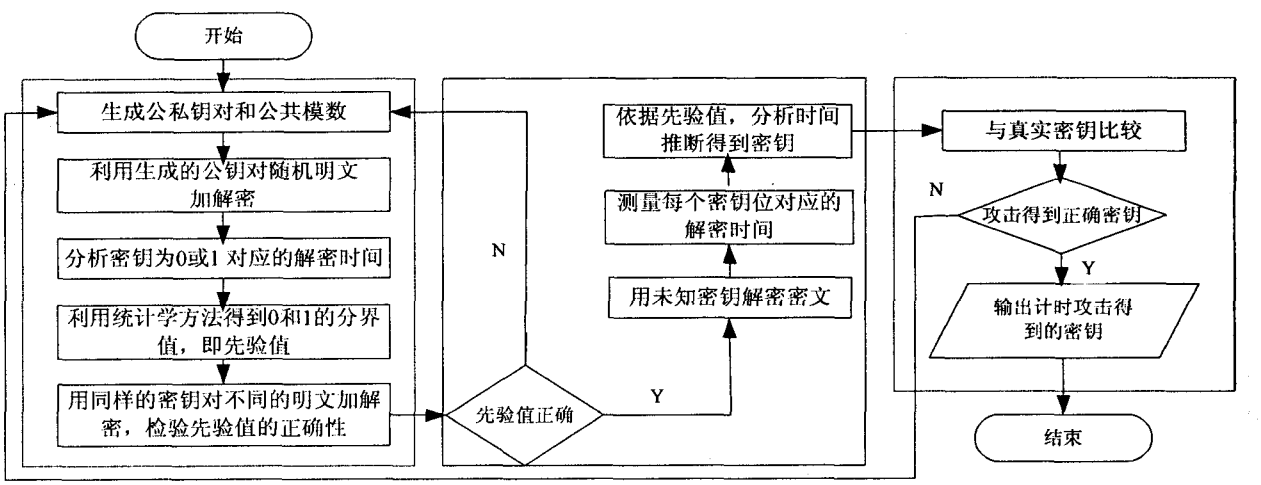

图 3　计时攻击流程

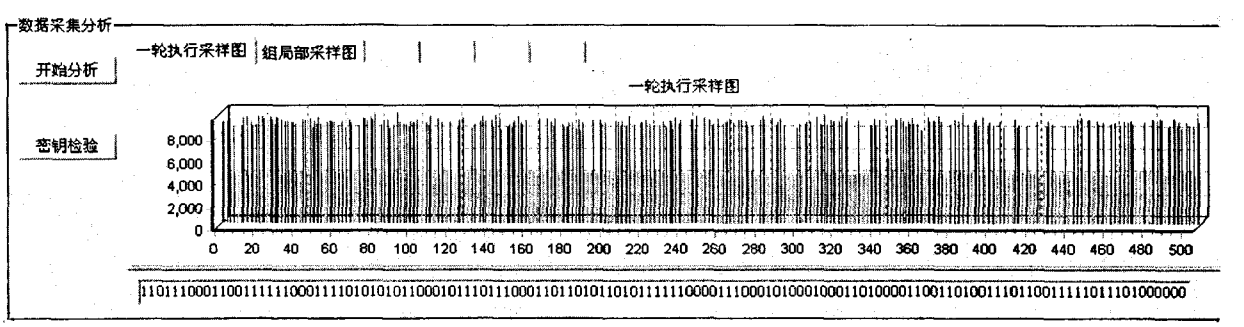

图 4　仿真平台下得到的各密钥位对应执行时间

参考文献:

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

(下转第 158 页)

改进后的 IKEv2 初始交换协议由四条消息组成，当检测到 DoS 攻击时，采用 cookie 机制增加两条消息。公钥由用户的身份信息 ID 确定，私钥由 PKG 生成。密钥衍生方法不变，认证载荷的计算基于 Weil 对的数字签名算法。

消息 1′：发起方 I 发起建立连接的请求，向响应方 R 发送安全关联 SA 提议、DH 交换值、保证密钥新鲜性和防重放攻击的随机值 N_i。

消息 2′：响应方 R 选择一个 SA 提议，并将自己的 DH 交换值、随机值 N_r 以及加密后的身份信息 ID_r、认证载荷 AUTH 一起发送给发起方 I。

消息 3′：在验证了响应方的认证载荷之后，I 向 R 发送用于 IPSec SA 的提议，同时包括自己的身份信息和证明、对方的身份信息以及流量选择符。

消息 4′：R 选择一个提议，并响应 I 的认证消息，完成初始交互。

基于 Weil 对改进的 IKEv2 协议具有以下优点：

(1)精简了系统架构，不必部署 PKI 就可实现对双方身份的认证，降低了系统的成本和开销；

(2)新的数字签名认证不需要传递、验证和存储数字证书，因此降低了对网络带宽和通信实体的要求。

文中采用同样的方法对改进后协议的安全性进行了形式化分析，结果表明改进后的协议满足认证性和私密性，能够对发起方身份提供主动保护，对响应方身份提供被动保护。

4 结束语

针对 Spi 演算不能形式化定义 D－H 密钥交换、IKEv2 密钥生成等问题，对其语义进行了应用扩展，扩展后的 Spi 演算能够分析 IKEv2 协议的安全性。基于扩展 Spi 演算的 IKEv2 协议形式化分析表明，IKEv2 协议满足认证性和私密性，但不能对发起方身份提供主动保护。文中对此提出了一种改进方案，改进后的协议解决了发起方身份保护的问题。

参考文献：

[1] Kaufman C. RFC4306 - Internet Key Exchange (IKEv2) protocol[S]. 2005.

[2] Harkins D, Carrel D. RFC 2409 - The Internet Key Exchange (IKE)[S]. 1998.

[3] Abadi M, Gordon A D. A calculus for cryptographic protocols: the Spi calculus[J]. Information and Computation, 1999, 148: 1 - 70.

[4] 季庆光，冯登国. 对几类重要网络安全协议形式模型的分析[J]. 计算机学报，2005，28(7)：1071 - 1083.

[5] Abadi M, Blanchet B. Just Fast Keying in the Pi calculus[J]. Journal of the ACM, 2007, 10(3): 1 - 54.

[6] 赵 宇，王亚弟，韩继红. 基于 Spi 演算的 SSL3.0 协议安全性分析[J]. 计算机应用，2005，25(11)：2515 - 2520.

[7] 顾永跟. 基于进程演算的安全协议形式化研究[D]. 上海：上海交通大学，2005.

[8] 张朝东，徐明伟. 密钥交换协议 IKEv2 的分析与改进[J]. 清华大学学报：自然科学版，2006，46(7)：1274 - 1277.

[9] 曹春杰，张帆，马建峰. 可证安全的 Internet 密钥交换协议[J]. 武汉大学学报：理学版，2006，52(5)：545 - 549.

[10] 韩旭东，汤 隽，郭玉东. 新一代 IPSec 密钥交换规范 IKEv2 的研究[J]. 计算机工程与设计，2007，28(11)：2549 - 2552.

[11] 邱司川，潘 进，刘丽明. IKEv2 协议的分析与改进[J]. 计算机工程，2009，35(15)：126 - 128.

[12] Shamir A. Identity - Base Cryptosystems and Signature Schemes[C]//Advances in Cryptology - Crytpo'84. LNCS 196. [s.l.]: Springer - Verlag, 1984: 47 - 53.

[13] Boneh D, Franklin M. Identity Based Encryption from the Weil Pairings[C]//Advances in Cryptology - Crypto 2001. LNCS 2139. [s.l.]: Springer - Verlag, 2001: 213 - 229.

(上接第 153 页)

[2] 吴文玲，冯等国. 分组密码的设计与分析[M]. 第 2 版. 北京：清华大学出版社，2009：169 - 171.

[3] 杨 玺. 计时攻击及其防范[J]. 通信技术，2008，41(7)：185 - 188.

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

[5] Intel. Using the RDSTC instruction for performance monitoring[R]. America: Intel, 1997.

[6] 钟 楼. 并行窗口算法在防御 RSA 计时攻击中的研究[J]. 计算机工程与应用，2006，43(4)：145 - 147.

[7] 陈财森，王 韬，郑媛媛. RSA 公钥密码算法的计时攻击和防御研究[J]. 计算机工程，2009，35(3)：123 - 125.

[8] Montgomery, Peter. Modular multiplication without trial division[J]. Mathematics of Computation, 1985, 44(170): 519 - 521.

[9] 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, 88(12): 3610 - 3617.

[10] 晏 楠，谷大武，丁 宁. RSA 体制下使用随机算法防御时间攻击的方法[J]. 计算机工程，2006，32(11)：174 - 176.

[11] 陈财森，王 韬，郑媛媛. 针对 OpenSSL 的 RSA 实现算法的计时攻击[J]. 军械工程学院学报，2009，21(2)：71 - 74.

[12] 陈财森，王 韬. 基于 Cache Missing 的 RSA 计时攻击[J]. 微电子学与计算机，2009，26(5)：180 - 182.