doi: 10.3969/j.issn.1006-1576.2011.06.028

一种针对采用 SPA-FA 防御措施的 RSA 故障分析算法

范黎恒[1]，陈财森[2]，曾剑隽[3]
（1. 总装重庆军代局，重庆 400060；2. 军械工程学院 计算机工程系，石家庄 050003；
3. 北京交通大学 电子信息工程学院，北京 100080）

摘要：在对 RSA 密码算法和安全错误攻击原理进行研究的基础上，以硬件模乘法器实现的模幂算法为分析对象，提出一种针对采用抗 SPA-FA 防御措施的 RSA 故障分析算法。攻击者利用在模幂运算过程中对其中的乘数寄存器注入故障，再通过判断最后输出结果的正确性来判断相应密钥位的值，并从 3 个方面给出算法的可行性分析。结果证明了 RSA 算法在采用硬件乘法器情况下的安全漏洞，能为有效防护算法的安全和研究提供参考。

关键词：旁路攻击；RSA 算法；安全错误攻击；故障分析；SPA-FA 防御措施
中图分类号：TP301.6　　**文献标志码：**A

A RSA Fault Analysis Algorithm of SPA-FA Resistant Measures

Fan Liheng[1], Chen Caisen[2], Zeng Jianjun[3]
(1. PLA Representation Bureau of General Equipment Department in Chongqing, Chongqing 400060, China;
2. Dept. of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China;
3. School of Electronic & Information Engineering, Beijing Jiaotong University, Beijing 100080, China)

Abstract: To research the principle of safe-error attack and RSA algorithm, the modular measures based on the hardware modular multiplier is taken as the analyzed target. Propose a fault analysis on RSA using SPA-FA resistant. The attacker inject fault into one of the multipliers during the modular exponentiation, and then it can use the output to deduce the corresponding bits of key. The vulnerability of RSA using SPA-FA resistant based on hardware modular multiplier is proved in theory, the practicability of the attack is analyzed in three aspects. The result shows that RSA algorithm can provide effective resistant measures safety and research with reference when there is safety leak in hardware multiplication situation.

Keywords: side channel attack; RSA algorithm; safe-error attack; fault analysis; SPA-FA resistant measures

0　引言

随着大规模集成电路的快速发展，电子硬件设备在安全领域的应用越来越广泛，智能卡、单片机等设备的安全性也得到越来越多的关注。同时，随着现代技术手段的不断提高，单纯从密码算法的数学结构来分析密码算法的安全性已远远不够，还需要从密码算法实现的角度来考虑安全问题。在实际应用中，密码算法通常在各种芯片上实现，如智能卡、加密存储卡、加密机芯片等，密码算法在这些芯片中产生的故障信息，可能被攻击者用来破解密码。故障攻击是指通过辐射、X 光、微探测或改变电压等方法在芯片中注入故障，依据从芯片泄漏的信息推算出密钥的一种攻击方式。

RSA 算法作为目前最流行的公钥密码算法，被广泛应用于智能卡中。其加解密运算的主要部分是模幂运算，因此大多数智能卡为加速其运算速度采用硬件模乘法器。但自从 1996 年 D. Boneh 等[1]人首次提出故障分析并成功破解 RSA 签名算法以来，RSA 算法在硬件实现上遭受旁路攻击和故障分析的威胁日趋严重。2005 年，Giraud 提出一种抵御简单功耗分析（SPA）和故障分析（FA）的模幂算法，后来 Boscher 等人又提出另一种抵御 SPA-FA 的算法[2]。他们声称这 2 种算法能够同时抵御针对 RSA 软件实现的计算安全错误攻击（C safe-error attack）和内存安全错误攻击（M safe-error attack）。此后人们又提出相应的防护措施[3-5]，但后来又接着被破解[6-7]。笔者基于硬件模乘法器实现的基础，针对已经采取抗 SPA-FA 防御措施的 RSA 算法进行研究分析，建立攻击模型，在执行模幂运算过程中，利用向模乘法器注入故障，修改寄存器或内存内容的手段，依据输出结果逐步恢复密钥位，同时对攻击算法的可行性进行分析。

1　RSA 算法及安全错误攻击

1.1　RSA 算法

RSA 是使用公钥（N,e）加密和使用私钥 d 解

收稿日期：2011-03-04；修回日期：2011-04-02
基金项目：国家自然科学基金（60772082）；河北省自然科学基金（08M010）资助
作者简介：范黎恒（1984—），男，山东人，工学硕士，助理工程师，从事信息安全研究。

密的公钥加密系统。它使用的模数 N 是 2 个大素数的乘积（$N=p\times q$），指数 e 和 d 必须满足 $e\times d=1\bmod(p-1)(q-1)$ 的条件。RSA 密钥是由公钥（N,e）和私钥 d 组成的。加密运算执行 $C=M^e \bmod N$，解密运算执行 $M=C^d \bmod N$，可以看出其核心是模幂运算 $M=C^d \bmod N$，这里 $N=p\times q$ 是 RSA 的模数，p 和 q 是素数因子，d 是解密私钥，C 为密文。硬件设备采用的 RSA 算法一般主要实现方式为平方乘算法。

1.2 安全错误攻击

安全错误攻击一般以模幂运算为分析对象，并假设攻击者的目的是要找到运算中隐藏着的幂指数 d。依据错误的方式可分为计算安全错误攻击（C safe-error attack）[5]和内存安全错误攻击（M safe-error attack）[8]。

计算安全错误攻击指的是攻击者向算术逻辑部件运算器（ALU）注入任何暂时性随机故障，例如以算法 1 为分析对象。由于该算法运行时消耗的时间与幂指数 d_i 无关，其消耗时间为恒定时间，因此攻击者能轻易获取每个循环中第 2 个乘法运算 $S[b]=S[b]\cdot M \bmod N$ 的时机。此外，当前指数位 d_i 为 0 时，从算法 1 中可以看出第 2 个乘法操作为多余操作，其运算出错不会影响最后的输出结果。因此，如果攻击者能够在第 i 个循环中执行 $S[b]=S[b]\cdot M \bmod N$ 运算时向 ALU 注入一个计算故障，然后通过判断输出结果的准确性，则可以推算出 d_i 为 1 还是 0。同理，内存安全错误攻击则需要在算法运行过程中向计算器或内存注入一个暂行性内存故障。与计算安全错误攻击相比较，它需要更为严格的假设，例如要求攻击者更好地攻击故障的位置和时机。针对模幂运算中的乘法运算 $A\cdot B \bmod N$，例如采用蒙哥马利乘法运算时，B_j 中一些数值可能在运算过程中被修改，但是 B 的值能够在计算 $B=A\cdot B \bmod N$ 后被纠正。由于被乘数 B 在每次循环运算过程中总被调用，因此任何对 B 注入的错误都可能对输出结果产生影响。

算法 1 抵御 SPA 的模幂运算算法

输入：$M,d=(d_{n-1},d_{n-2},\cdots,d_0)_2,N$

输出：$M\text{^}d \bmod N$

1) $S[0]=1$
2) For i from n−1 to 0 do
3) $b=-d_i$
4) $S[0]=S[0]\text{^}2 \bmod N$
5) $S[b]=S[b]\cdot M \bmod N$
6) Return $S[0]$

2 故障分析模型

执行一次故障攻击，首先需要对算法和故障影响进行分析研究。除了考虑如何注入故障外，还要知道如何利用故障的结果获取有用信息，因此很有必要建立一个注入故障与故障对算法产生影响之间关系的模型。文中的攻击模型与 M safe-error 攻击类似，由于大多数智能卡已经采用硬件乘法器加速密钥算法运算性能，因此以该类设备为攻击分析对象，而不是软件实现的 RSA 算法，建立的模型如图 1。M safe-error 攻击假设被乘数 A 和乘数 B_i 都被发送到 MUL 处理单元，当一个硬件乘法器被使用时，该假设可能失效。循环中 A 值的每次载入显得繁琐并不能有效利用乘法器。这意味着每次计算 $A\bullet B_j$ 都需要从内存中通过总线将 A 载入。因此模乘法器从乘法运算开始就一次载入 A，然后再不断地载入 B_i 的值执行计算，提高运算效率[9]。攻击利用在第 i 个循环运算执行过程中对 B_j 注入故障，分析最后输出结果的正确性来判断密钥 d_i 的值，最后通过重复执行攻击恢复完整密钥。

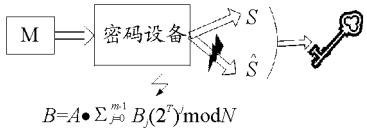

图 1 故障分析模型

因此文中的攻击模型假设：1) 攻击者向寄存器或内存注入一个暂行性内存故障；2) 攻击对象使用硬件模乘法器。

3 针对用 SPA-FA 防御措施的故障分析算法

3.1 典型的 SPA-FA 防御算法

现有的抗 SPA-FA 防御算法以 Boscher 等人提出的 right-to-left 二进制平方乘算法[2]和 Giraud 提出的蒙哥马利阶梯算法[10]为代表。这 2 种算法都利用了每次循环运算中临时变量之间的某种关系，通过在最后输出计算结果之前对存在的关系进行验证来防止错误结果的输出。此处笔者仅以 Boscher 提出的算法为分析对象，描述如算法 2。

算法 2 抗 SPA-FA 的 Right-to-Left 模幂算法

输入：$M,d=(d_{n-1},d_{n-2},\cdots,d_0)_2,N$

输出：$M\text{^}d \bmod N$ or “Error”

1) $S[0]=1$
2) $S[1]=1$

3)　$A=M$
4)　for i from 0 to $n-1$ do
5)　{
6)　　$S[\overline{d_i}]=S[\overline{d_i}]\cdot A \bmod N$
7)　　$A=A^2 \bmod N$
8)　}
9)　if ($M\cdot S[0]\cdot S[1]=A \bmod N$) and ($A\neq 0$) then
10)　　return $S[0]$
11) Else
12) return Error

从算法 2 可以看出，无论密钥位 d_i 为 1 还是为 0，都执行乘法操作，因此可能有效地抵御简单功耗分析，另外由于利用每次循环运算过程中 $M\cdot S[0]\cdot S[1]=A \bmod N$ 这一恒等式的关系，在最后输出结果之前检验这一关系是否成立，从而判断计算过程中是否有错误发生，从而抵御故障分析。

3.2　密钥分析算法

首先，从算法 2 的循环计算步骤 4 到步骤 7 中，可得出如下的计算：

当 d_i=0 时，执行计算：

$S[1]=S[1]\cdot A \bmod N$ 和 $A=A^2 \bmod N$

当 d_i=1 时，执行计算：

$S[0]=S[0]\cdot A \bmod N$ 和 $A=A^2 \bmod N$

那么假设攻击对象采用硬件模乘法器，参数 S[i]和 A 将被注入指定的运算器，然后将输出结果重新存储在 S[i]中。如果在 S[i]的值被载入之后，S[i]的寄存器被注入一个错误，那么这个错误是否被检测则取决于指数 d_i 的值。

例如假设 d_i 为 1 时，攻击者在 S[0]载入之后对其注入一个临时故障。由于计算输出结果重新存储在 S[0]，那么注入的错误没有对最后的结果产生影响。而当 d_i 为 0 时，攻击者在 S[1]被载入之后对 S[0]注入一个临时故障，由于输出结果仍然存储在 S[1]，S[0]的值没有被纠正，因此其错误将在后面的计算中传播，从而导致最终计算错误。对 S[1]注入故障的分析算法与 S[0]注入故障的情况一样。因此可以将密钥分析算法描述如下：

1) for i from 0 to $n-1$ do；

2) 在运算算法 2 步骤 6 过程中，对寄存器 S[0]或者 S[1]注入一个安全错误；

3) 如果最后输出“Error”，则推断 d_i 为 0，否则为 1。（相对 S[1]注入故障的情况刚好相反）；

4) 最后进行密钥验证。

3.3　算法的可行性分析

主要从以下 3 个方面进行算法的可行性分析：

1) 攻击者对故障注入时机的控制。由于算法在循环过程中重复执行平方和乘法操作，能很容易地利用设备的功耗属性和计时分析找出执行乘法操作的时机。Aumuller 等人利用功耗属性控制其攻击的时机[12]。另外，由于硬件模乘法器通常比 ALU 消耗更多的功耗，能很容易地找出模幂运算的间隔。另外由于步骤 6 是整个运算操作过程中消耗时间最多的部分，因此并不难控制故障注入的时机。不需要任何特定的故障值，只需要令目标寄存器更改为任意值即可，这也降低了攻击的难度。

2) 修改寄存器值的可能性。寄存器值的修改高度依赖于设备的结构以及攻击方法，还有和 ICs 之类的设备采取的防止故障分析的措施有关。目前已经有针对未采取保护措施的设备执行的故障分析结果。文献[11]已经成功地利用更改 IC 智能卡的供给电压 Vcc 的值来产生故障。Bar-EL 等人实现了类似的实验。Skorobogatov 和 Anderson 也在文献[12]中表明 SRAM 的值是可能被更改的。

此外还有很多基于修改寄存器值的文献已经公开发表。Boneh 等人假设存储在寄存器的值能够被更改，给出 RSA 和 Rabin 签名方案的攻击方法[1]。Seifert 模型是假设攻击者能够注入随机寄存器故障，破解 RSA 认证[13]。文献[14]中也给出类似的故障模型。Joy 和 Yen 利用 M safe-error 故障模型执行攻击[12]。

3) 如何检验错误是否发生。当错误发生时，算法 2 输出“Error”，使我们能够很容易判断是否有错误发生。另外假如设备发现输出错误时，重新执行运算直到正确的结果输出，对于这种情况，攻击者可以通过观测设备执行时间来判断错误是否发生，发生一次错误的执行时间大约会是正常执行的 2 倍。

综上所述，基于更改寄存器值的随机暂行故障模型具有一定可行性。

4　结论

笔者从理论上给出针对已经采取抗 SPA-FA 防护措施的 RSA 故障分析算法，证明 RSA 算法在采用硬件乘法器情况下的安全漏洞。由于目前大多数智能卡都采用硬件模乘法器，因此可以认为该漏洞是普遍存在的。该故障模型为暂时性行随机内存故障，现已经被许多公开发表的相关文献所接受，但至今仍然没有关于安全错误攻击的相关实验结果，

（下转第 96 页）