文章编号：2096-1618(2016)04-0348-05

针对 Montgomery 模幂算法的选择明文 SPA 攻击

万武南[1]， 陈 俊[2]

(1. 成都信息工程大学信息安全工程学院，四川 成都 610225；2. 成都信息工程大学计算机学院，四川 成都 610225)

摘要：大整数模幂运算的核心是大整数模乘运算，一般采用 Montgomery 模乘算法实现。针对 Montgomery 模乘算法实现方式中大整数拆分成多个字节或字相乘存在功耗泄露问题，提出一种选择能产生 Montgomery 模乘算法的某操作数由多个字节或字为零组成的大整数的特定明文，简单功耗分析(simple power analysis，SPA)的方法。通过输入特定明文，一条功耗曲线就能将模幂算法中平方和乘运算位置区分出来，私钥攻击难度下降。在搭建真实的8051智能卡芯片攻击环境下，输入特定明文进行SPA攻击，1024比特私幂指数私钥攻击准确率可达99 %。实验结果表明可选特定明文数量多，用单一的屏蔽特殊明文的方法无法有效防范文中提出的SPA攻击，最后给出防范此选择明文SPA攻击的建议。

关 键 词：侧信道攻击；简单功耗分析；Montgomery 模乘算法；选择明文；模幂算法

中图分类号：TN918.4 **文献标志码**：A

0 引言

自 Kocher 等[1]提出差分功耗分析(differential power analysis，DPA)方法以来，观测密码算法运行过程中芯片的功耗变化从而获取其密钥的方法得到深入研究和广泛发展，简单功耗分析(simple power analysis，SPA)[2]，DPA[2-3]，相关功耗分析(correlation power analysis，CPA)[4-5]等方法先后被提出。目前，RSA、ECC 等公钥密码算法被广泛应用到各种不同嵌入式硬件设备中，公钥密码算法易受侧信道攻击方法的威胁。

大整数模幂算法是 RSA、ElGamal 等公钥密码算法最为核心和重要的计算，目前一般采用 Montgomery 模乘算法实现，因此提出许多针对 Montgomery 模幂算法的侧信道攻击方法[6-12]。Messerges 等[3]提出能够有效攻击 RSA 算法的 DPA 方法，例如 SEMD，MESD 和 ZEMD。也提出由于信号受噪声干扰，根据硬件密码设备运行中的采集功耗曲线直接估计密钥信息的 SPA 攻击难以提取私幂指数私钥。但文献[6-7]提出 SPA 改进算法，采用特殊明文对可以得到功耗消耗差异比较大的功耗曲线，以分析相应的私幂指数私钥。随后，Naofumi Homma 等[8]针对模幂算法实现对 FPGA 硬件设备的选择明文 SPA 攻击，通过使用两个特殊明文进行碰撞攻击。但没有做智能卡上的 SPA 攻击，并且至少需要两条功耗曲线。文献[9]提出通过选择单一明文 N-1 的 SPA 攻击方法，但此 SPA 攻击易防御，只需模幂算法实现中对 N-1 明文屏蔽就可防范。

针对硬件设备实现大整数 Montgomery 模乘算法，需要将大整数乘分解成若干短整数字节或字相乘存在乘法功耗泄露问题，提出一种选择明文的 SPA 攻击，通过采集特定明文的模幂功耗曲线，提取私幂指数。在搭建真实的 8051 芯片攻击环境下，单条曲线恢复 1024 比特幂指数准确率可达99 %，且满足 SPA 攻击的特定明文的曲线条数数量大，用单一的屏蔽特殊明文的方法无法防范，需对模幂底数进行随机掩码防御。

1 Montgomery 模乘算法实现

1.1 Montgomery 模乘基本算法

1985 年，Montgomery[12]提出 Montgomery 算法，设计思想借助一个新的完全剩余系，采用模加右移法，避免模运算中复杂且耗时的除法运算。

Montgomery 模乘算法基本思想为：计算 $a\times b=a\cdot b\cdot r^{-1}\bmod n$，符号“×”为 Montgomery 乘积，符号“·”为普通乘法。其中 n 为 k 比特的整数，$2^{k-1}<n<2^k$，$a<n$，$b<n$；$gcd(n,r)=1$，通常 $r=2^k$，则 r^{-1} 是 r 模 n 的逆，即 $r\cdot r^{-1}\equiv 1(\bmod n)$，$r\cdot r^{-1}-n\cdot n'=1$，$n'=-n^{-1}\bmod r$。

Montgomery 乘积算法记为：$MonProc(a,b,r,n)=$算 $a\times b=a\cdot b\cdot r^{-1}\bmod n$，算法具体过程如下

算法 1 Montgomery 模乘

收稿日期：2016-07-01

基金项目：国家自然科学基金面上资助项目(61572086)；四川省大数据与智慧城市创新开放基金资助项目(RWS-CYHKF-01-20150003)；四川省教育厅重点资助项目(16ZA0212)

输入　整数 a,b,r,n

n 为 k 比特的整数,即 $2^{k-1}<n<2^k a<n, b<n; gcd(n, r)=1$

$r=2^k$

输出　$a \cdot b \cdot r^{-1} \bmod n$

$t=a \cdot b$

$m=t \cdot n \bmod r$

$u=(t+m \cdot n)/r$

若 $u \geqslant n$, 则输出 $u-n$,否则输出 u

1.2　Montgomery 模乘实现方式

算法 1 给出了 Montgomery 模乘算法的基本形式,为加快 Montgomery 模乘的计算,1990 年 Duccse 采用 $n'_0=-n_0^{-1} \bmod 2^w$ 代替 n'对算法做改进[13],其中 w 为字长,通常 $w=32$,算法中大整数的进制为 $W=2^w$。1996 年,Koc 等[14]对 Montgomery 模乘进行分析,提出 SOS、CIOS、FIOS、FIPS 和 CIHS 5 种实现方法,每一种算法都将大整数乘法分解为若干次小整数的乘积,适合在各种 CPU 软件实现和各种硬件芯片上实现。下面给出 FIOS 实现方式。

$MonProc(a,b,r,n)=a \times b=a \cdot b \cdot r^{-1} \bmod n$,算法中大整数采用的进制为 $W=2^w$,存储空间以 w 比特的存储单元进行存储。设 n 为 k 比特的整数,$2^{(e-1)w}<n<2^{ew}, r=2^k=2^{ew}, a<n, b<n; gcd(n,r)=1, n'_0=-n_0^{-1} \bmod W$。FIOS 实现算法如算法 2 所示。

算法 2　Montgomery 模乘算法 FIOS 实现

输入　整数 $n=(n_{e-1} \cdots n_1 n_0)_W, a=(a_{e-1} \cdots a_1 a_0)_w$

$b=(b_{e-1} \cdots s_1 s_0)_W, r=2^k=2^{sw}$

输出　$t=a \cdot b \cdot r^{-1} \bmod n$,则 $t=(t_{e+1} \cdots t_1 t_0)_W$

for $i=0$ to $e-1$

　　$(C,S)=t_0+a_0 \cdot b_i$

　　$u=S \cdot n_0^{'} (\bmod W)$

　　$(C,S)=(C,S)+u \cdot n_0$

　　$(C,S)>>w$

for $j=1$ to $e-1$

　　$(C,S)=(C,S)+a_j \cdot b_i+t_j+n_i \cdot u$

　　$n_{j-1}=S$

　　$(C,S)>>w$

　　$(C,S)=(C,S)+C$

　　$t_{e-1}=S$

　　$t_e=t_{e+1}+C$

　　$t_{e+1}=0$

if $t>n$ 则 $t=t-n$

输出 $(t_{e-1} \cdots t_1 t_0)_W$

算法 2 中 C 表示 w 位的进位存储器,S 表示 w 位的求和存储器,u 表示 w 位临时存储单元。在 FIOS 实现算法中包含 $5e^2+3e+2$ 个加操作、$7e^2+5e+2$ 个读操作、$3e^2+4e+1$ 个写操作和 $2e^2+e$ 乘法操作。

1.3　Montgomery 模幂算法

公钥密码算法中大整数模幂运算 $m^d \bmod n$,通常采用一种称为二元表示法(binary representations, BR)的迭代算法实现,将指数 d 二进制化实现,即将指数 d 表示成 $d=(d_{l-1} \cdots d_1 d_0)_2, d_i \in \{0,1\}$,RSA 算法 l 通常为 1024 或 2048。模幂运算有 L-R 和 R-L 两种算法。选取 L-R 算法,Montgomery 模幂 L-R 算法如下[8]。

算法 3　Montgomery 模幂 L-R 算法

输入　整数 $n=(n_{e-1} \cdots n_1 n_0)_W$, n 为 k 比特的整数,

即 $2^{k-1}<n<2^k, m=(m_{e-1} \cdots m_1 m_0)_W$,

$d=(d_{l-1} \cdots d_1 d_0)_2, d_{l-1}=1, r=2^k=2^{ew}, W=2^w$

输出　$R=m^d \bmod n, 0 \leqslant R<n$

步骤 1　$A=r^2 (\bmod n)$

步骤 2　$m'=MonProc(m,A,r,n)$

步骤 3　$R=MonProc(A,1,r,n)$

步骤 4　for $i=l-1$ to 0

　　(i) $R=MonProc(R,R,r,n)$　//平方运算

　　(ii) if $(d_i==1)$ then $R=MonProc(R,m',r,n)$ //乘运算

步骤 5　$R=MonProc(R,1,r,n)$

步骤 6　输出 R

2　Montgomery 模幂算法选择明文 SPA 攻击

2.1　Montgomery 模幂功耗分析攻击模型

在真实环境下,密码算法运行的功耗曲线采集要受到设备、环境等多方面的影响,算法产生具体的功耗的组成如式(1)。

$$P_{total}=P_{op}+P_{data}+P_{el.\ noise}+P_{const} \tag{1}$$

式中,P_{total}为总功耗;P_{op}为操作依赖分量;P_{data}为数据依赖分量;$P_{el.\ noise}$为电子噪声;P_{const}为恒定分量。

根据算法 2 可知,FIOS 实现操作主要包括乘法、加法、移位和数据读取操作,每种操作都存在功耗泄露。相对于加法、移位和读取操作,功耗消耗较大。乘法操作有与位跳变相关的功耗消耗,根据汉明重量功

耗模型可知,理论上来说乘法运算中两操作数的汉明重量不同,其功耗也有高低之分,通常汉明重量大,功耗消耗大,汉明重量小,功耗消耗小。对于大整数 Montgomery 模乘算法被拆分成多个短整数字节或字 $a_j \cdot b_i$,则 $a_j=0, b_i\neq 0$ 产生功耗要比 $a_j\neq 0, b_i\neq 0$ 的功耗要低,比 $a_j=0, b_i=0$ 产生功耗要高。因此,把字节或字乘法的功耗消耗记为 P,根据操作数零和非零分为 4 种,如表 1 和图 1 所示。

表 1 $a_j \cdot b_i$ 运算时的功耗情况

a_j	b_i	对应功耗 P
$a_j\neq 0$	$b_i\neq 0$	P_{High}
$a_j=0$	$b_i\neq 0$	P_{Medium}
$a_j\neq 0$	$b_j=0$	P_{Medium}
$a_j=0$	$b_j=0$	P_{Low}

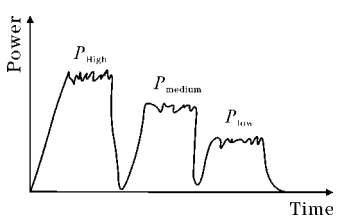

图 1 乘法运算中功耗大小的 3 种曲线图

据功耗组成公式(1)可知,在同样的硬件设备环境下,对于不同的输入,FIOS 算法的操作完全一样,理论上一次 $MonProc(a,b,r,n)=a\times b=a\cdot b\cdot r^{-1}\bmod n$ 产生功耗,其 P_{op} 和 P_{const} 是一样。功耗 P_{data} 跟 Montgomery 模乘算法输入 a 和 b 两操作数直接相关,由于乘法操作功耗消耗较大,而其他操作功耗较少,因此只考虑操作数对乘法操作功耗影响。在 FIOS 中 e^2 个乘法运算跟输入 a 和 b 两数直接相关,即大整数相乘的主要功耗消耗 $P_{LIM}(a,b)$ 可由字节或字相乘组成,如公式(2)所示。

$$P_{LIM}(a,b)=\sum_{j=0}^{e-1}\sum_{i=0}^{e-1}p(a_j,b_i) \tag{2}$$

其中 $p(a_j,b_i)$ 是每次字节或字相乘消耗功耗。

根据图 1 乘法运行功耗原理可知,字节或字乘法操作数会影响功耗的高低变化。因此在 FIOS 实现中,若输入 $a=(a_{e-1}\cdots a_1a_0)_W$ 和 $b=(b_{e-1}\cdots s_1s_0)_W$ 两大整数操作数中,大整数 a 中 $a_j=0$ 较多,而 $a_j\neq 0$ 较少(即大整数 a 中 0 比特较多,1 比特较少),则字节或字乘法功耗 $p(a_j,b_i)$ 的值有较多 P_{Medium},因此与操作 a 和 b 随机情况,功耗 $P_{LIM}(a,b)$ 呈现明显下降趋势。

2.2 Montgomery 模幂算法选择明文 SPA 攻击

在 RSA 算法中,幂指数 d、模数 n 的长度通常为 1024 或 2048 比特。并且由于硬件设备处理一般为 8、16 或 32 比特。硬件设备的字长 $w=32$,因此算法 3 中参数 $k=1024, e=32, l=1024, r=2^{1024}$。

根据算法 3 可知,模幂运算中与幂指数相关的功耗主要由步骤 4 产出。若指数 $d_i=0$,则只进行(i)步操作,若 $d_i=1$,则需要进行(i)和(ii)步。具体操作如表 2 所示。

表 2 与幂指数相关功耗操作

d_i	对应功耗消耗操作
$d_i=0$	(i) $MonProc(R,R,r,n)$ 平方(S)
$d_i=1$	(i) $MonProc(R,R,r,n)$ 平方(S)
	(ii) $MonProc(R,m',r,n)$ 乘(M)

因此若(i)步和(ii)步 Montgomery 模乘运算产生功耗存在如图 2 所示高低变化,(i)步功耗高,(ii)步功耗低。图 2 中横轴为时间,纵轴为功耗电压值,白色底为(i)步平方运算(S),灰色底为(ii)步乘运算。则通过寻找低功耗位置,能分析出幂指数 $d_i=1$ 的位置,并且可知 $d_i=1$ 的功耗曲线为高、低两个模乘功耗组成,而 $d_i=0$ 则由单个高模乘功耗组成,因此通过单条功耗曲线可以快速分析出幂指数 d,图中幂指数为 110010。

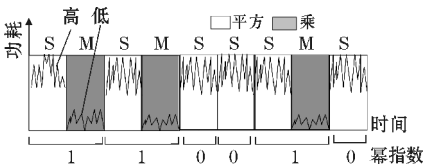

图 2 算法 3 模乘功耗示意图

(i)步和(ii)步都是进行 Montgomery 模乘操作,由 1.1 节分析可知,两步的功耗高低变化与输入两操作数直接相关。而(i)步中 $MonProc(R,R,r,n)$ 中,输入的两操作数为 R,根据算法可知每次循环迭代输入都是动态变化,而(ii)步中 $MonProc(R,m',r,n)$ 中,输入的两操作数为 R 和 m',$m'=MonProc(m,r^2,r,n)=mr(\bmod n)$ 在一次大整数模幂运算是固定不变。

由 $m'=mr(\bmod n)$ 可知 m' 的值由 m,r,n 值计算而来,在 RSA 算法中 n 和 r 值都是公开的,因此对于给定的 r 和 n,可以选定特定明文 $m=m'r^{-1}(\bmod n)$,使 m' 刚好由多个为零的字节或字组成的大整数,使(ii)步中功耗要明显低于(i)步中功耗。

如若 $m'=(m'_{e-1}\cdots m'_1 m'_0)_W=2^{1023}$,即 $m'_i=0, i=0,1,2,\cdots,e-2$,则选择的特定明文(底数) $m=2^{1023}r^{-1}\pmod n$,如图 3 所示,在(ii)步中功耗要明显小于(i)步功耗。因此,给定 n 和 r,通过选定为零的字节或字组成的大整数 m',计算出选择底数 m,则通过分析一条功耗曲线高低区分出模乘的功耗运行的是(i)步和(ii)步操作,从而区分出是幂指数 0 和 1 的。

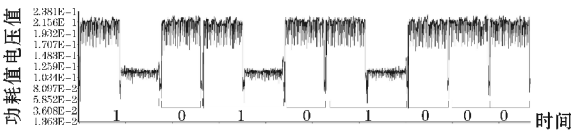

图 3　明文为 $2^{1023}r^{-1}\pmod n$ 时的功耗曲线

3　实验结果和性能分析

3.1　功耗测试环境

实验是在自主开发的功耗采集和分析平台上对 8051 芯片进行测试,整个功耗分析攻击平台的硬件配置主要有:工作站、数字采样示波器(Tektronix PPO4032)、SASEBO 功耗采集板。

实验 PC 工作站与示波器通过网线相连,示波器与 SASEBO 相连,SASEBO 通过 Usb 串口通信与 PC 机相连。示波器主要是接收指令采集功耗曲线和触发信号。SASEBO 采集板集成了智能卡读卡器、FPGA 开发板多种功能。插入具有算法 3 的 8051 智能卡芯片,PC 机通过 Usb 串口通信发送相关命令给 SASEBO,密码芯片工作和示波器触发,示波器采集功耗,通过以太网把功耗数据传送到 PC 机进行功耗分析。

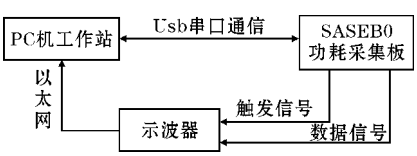

图 4　功耗采集实验硬件环境

3.2　实验数据处理和分析

在如图 4 的实验环境下,随机生成长度 1024 比特的 1000 个不同幂指数 d 和 n,产生特定明文 m,分成采用示波器采样频率1 MHz,2.5 MHz,每次采集 1 条功耗曲线,对私幂指数分析实验结果见图 5。

图 5 中横坐标表示 l 的值,(即 m' 中为零的字节或字的个数,$m'_i=0, i=0,1,\cdots,l$ 的情况),纵坐标表示 SPA 攻击提取 1024 比特幂指数的准确率。

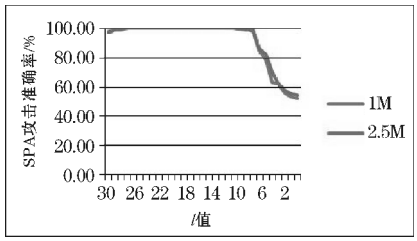

图 5　不同特定明文的 SPA 攻击准确率

从图 5 实验结果可以看出,m' 中为零的字节或字的个数超过 25,即 1024 中 800 多比特为零,幂指数攻击准确率为99 % 以上,m' 中为零的字节或字的个数小于 7 时(即 1024 中少于 224 多比特为零),幂指数攻击准确率急剧下降。示波器采用频率为2.5 M比采样频率为1 M时,幂指数攻击准确率略有提高,但滤波器采样频率的增加,对单条功耗曲线的选择明文 SPA 攻击准确率提高影响不大。

4　结束语

针对 Montgomery 模幂算法,提出一种选择明文 SPA 攻击,通过输入特殊特性的明文,可以通过一条功耗曲线,提取 1024 比特模幂指数准确率达到了99 %。此方法可以用来检测硬件设备模幂算法对明文底数是否做了防御。并且针对选择明文 SPA 模幂底数防御措施设计中,不能只单采用屏蔽特殊明文的方法无法防范,需对模幂底数进行随机掩码或者采用 RSA-CRT 方式实现,能有效防范文中提出的选择明文 SPA 攻击。

致谢:感谢中央高校基本科研业务费项目"智能卡边信道安全检测平台"对本文的支持!

参考文献:

[1] Kocher P, Jaffe J, Jun B. Differential power analysis[C]. Advances in Cryptology-CRYPTO´99, California, USA: Springer, 1999: 789-789.

[2] S M Yen, W C Lien, S J Moon, et al. Power Analysis by Exploiting Chosen Message and Internal Collisions-Vulnerability of Checking Mechanism for RSA-Decryption[C]. Proc. Mycrypt '05 ,2005: 183-195.

[3] Messerges T S, Dabbish E A, Sloan R H. Investigations of power analysis attacks on smartcards[C]. Proc USENIX Workshop Smartcard Technology, Chicago, Illinois, USA: IEEE Press, 1999:151-161.

[4] M F Witteman, Jasper G J van Woudenberg, et al.

Defeating RSAMultiply-Always and Message Blinding Countermeasures [C]. The Cryptographers´ Track at the RSA Conference 2011, San Francisco, CA, USA,2011:14-18.

[5] E AkalpKuzu, A Tangel. A new style CPA attack on the ML implementation of RSA[C]. Computer Science and Engineering Conference (ICSEC), 2014.

[6] A P Fouque, F Valette. The Doubling Attack-WhyUpwards is Better Than Down wards, Proc. Int'l WorkshopCryptographic Hardware and Embedded Systems (CHES '03),2003: 269-280.

[7] S M Yen, W C Lien, S J Moon,et al. Power Analysis by Exploiting Chosen Message and Internal Collisions-Vulnerabilityof Checking Mechanism for RSA-Decryption[C], Proc. Mycrypt '05,2005: 183-195.

[8] Naofumi Homma, Atsushi Miyamoto,TakafumiAoki,et al. Comparative power analysis of modular exponentiation algorithms[J]. IEEE Transations on computer,2010,59(6): 795-807.

[9] 曹娜娜. 针对8051芯片RSA算法的选择明文SPA攻击[D]. 成都:成都信息工程学院,2012,14-38.

[10] J Heyszl, A Ibing, S Mangard,et al. Clustering Algorithms for Non-profiled Single-Execution Attacks on Exponentiations[C]. Smart Card Research and Advanced Applications. Volume 8419 of the series Lecture Notes in Computer Science, 2014:79-93.

[11] C Clavier, B Feix, G Gagnerot,et al. Horizontal Correlation Analysis on Exponentiation[C]. Proc. ICICS, ser. Lecture Notes in Computer-Science,2010,6476:46-61.

[12] Montgomery,P I. Modular Multiplication Without Trial Division[J]. Mathematics of Computation, 1985,44(170):519-521.

[13] Dusse S R, KaliskiJr B S. A Cryptographic Library for the Motorola DSP56000. Advances in Cryptology-FUROCRYPT90,1990.

[14] Knuth D E, The Art of Comparing Montgomery Multiplication Algorithms[J]. IEEE Micro,1996, 16(3):26-33.

A Simple Power Analysis Attack on the Montgomery Modular Exponentiation Algorithms

WAN Wu-nan[1], CHEN Jun[2]

(1. College of Information Security Engineering, Chengdu University of Information Technology, Chengdu 610225, China;2. College of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Large module multiplication which is normally implemented by the Montgomery multiplication algorithms is the kernel of large module power multiplication. Accord to the problem that power leakages are mainly caused by multiplication operations of many bytes or words which the large integers are divided intoin the Montgomery multiplication algorithms, a chosen particular message simple power analysis attack is proposed in this paper, when an operationwhich is composed of many zero bytes or wards is generated by the particular message. Squaring operations and multiplication operations canbe? distinguished by using one power trace which is generated by the specific message inputin modular exponentiation algorithms, and thereby make secret exponent analysis less difficult. The correctness rate of private exponents with 1024 bits could reach 99% when the particular message is inputted during SPA attack in smart card 8051 chip. The experimental results indicate that there are many optional specific plaintexts, thus the method of the single shielding could not protect effectively against the proposed SPA attack. Finally, we will give advice on countermeasures to such enhanced chosen message SPA techniques.

Key words: side channel attack; simple power analysis; Montgomerymultiplication algorithm; chosen plaintext; module exponentiation algorithm