基于CRT组合运算故障的RSA故障分析研究

陈财森 王 韬 寇应展 张金中
(军械工程学院计算机工程系 石家庄 050003)

摘 要 原有的基于模幂运算故障的RSA-CRT故障攻击算法,因添加了错误检验操作而失效。为寻找新的故障攻击方法,以Shamir防御算法为攻击分析对象,对CRT组合运算步骤产生故障的情况进行分析,建立了基于CRT组合运算故障的攻击模型,提出了能够完整推算出RSA密钥的故障攻击算法。进行了推导论证和实验仿真,结果表明原有防御措施并不能有效地抵御故障攻击,新的攻击算法具有良好的可行性,在算法复杂度上,对固定故障值仅需2个注入故障,对随机故障给出优化的密钥空间搜索方案。最后分析了原有防御算法的问题,同时给出相应的防御建议。

关键词 旁路攻击,故障分析,中国剩余定理,RSA密码算法
中图法分类号 TP393.08 **文献标识码** A

Research on Fault Analysis against RSA Based on Fault in CRT Combination Operation

CHEN Cai-sen WANG Tao KOU Ying-zhan ZHANG Jin-zhong
(Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract The former fault analysis can not attack on RSA-CRT with corresponding countermeasure. In order to find the new vulnerability to fault analysis, this paper took Shamir countermeasure as the analyzed object. An attack model based on fault in CRT combination operation was advanced, and gave a differential fault analysis algorithm that can completely recover the RSA key. The fact that the previous countermeasures can not effectively resist the differential fault analysis was demonstrated, and the complexity of our attack was estimated both by a theoretical analysis and software simulations. Experiment results show that the new fault analysis algorithm has well feasibility; it only requires two fault injections for permanent fault, and an improved scheme of key searching for random fault is advanced. Finally, a corresponding advice on countermeasure to differential fault analysis was given by analyzing the problem of previous countermeasures.

Keywords Side channel attack, Fault model, Differential fault analysis, Error checking, Chinese remainder theorem, RSA

1 引言

随着微型集成电路芯片在各个领域的广泛应用,为保证芯片内部存储信息的安全性,防止非授权访问,采用了密码技术,但密码算法只能够从密码算法角度分析保证其安全性,未必能保证其实现的安全性。由于算法在设备的实现过程中可导致旁路信息泄露,攻击者利用采集、分析这些泄露信息可推导出密钥。这种专门针对密码算法实现的攻击,称为旁路攻击[1]。故障攻击是旁路攻击的一种,指攻击者使用物理手段,如电磁干扰、激光、时钟频率突变等,来干扰密码芯片正常工作,使其执行某些错误操作,最后依据错误信息推算出密钥的一种攻击方式[2]。

RSA作为目前最为广泛使用的公钥密码算法,为提高其执行速度,在实现时采用中国剩余定理(CRT),但CRT算法在执行过程中却存在遭遇故障攻击的隐患。1996年D. Boneh等人首次提出故障分析[2],采用该方法成功地攻破RSA签名密钥,而后Arjen Lenstra扩展了原有故障分析算法,提出了一种只需要一个错误的签名消息就可以成功破解密钥的攻击方案[3]。C. Aumuller等人给出RSA-CRT算法的攻击手段及相应的对策[4]。奥地利的IAIK实验室也专门针对故障攻击的故障注入方式进行了研究[5]。2009年,Emmanuelle等人提出了针对RSA算法的二阶故障分析,对算法执行过程中出现两次故障的情况进行分析[6]。在防御故障攻击研究方面以Shamir[7]和Johannes Blomer等人[8]提出的算法为典型,分别简称Shamir算法和BOS算法,其基本思想均是在最后输出结果时进行正确性检验,结果表明采用了他们

到稿日期:2010-12-04 返修日期:2011-04-17 本文受国家自然科学基金(60772082),河北省自然科学基金(08M010)资助。
陈财森(1983-),男,博士生,主要研究方向为信息安全和公钥密码旁路分析,E-mail:caisenchen@163.com;王 韬(1964-),男,教授,博士生导师,主要研究方向为网络安全与对抗;寇应展(1962-),男,教授,硕士生导师,主要研究方向为计算机系统安全;张金中(1985-),男,硕士生,主要研究方向为信息安全与网络对抗。

的防御算法能够有效地抵御原有故障分析算法的攻击。

本文结合原有故障分析算法，以 Shamir 防御算法为分析对象，根据分析发现，防御算法没有考虑到 CRT 组合运算步骤产生故障的情况，并建立基于组合运算故障的攻击模型，给出密钥分析算法，证明采用 Shamir 防御算法的 RSA 仍存在遭受故障攻击的隐患。通过算法分析和推理论证，并对故障出现的不同情况进行分析讨论，给出相应的攻击算法，并进行仿真实验，对攻击算法的可行性及复杂度进行分析，最后提出防御措施建议，以提高 RSA 算法的安全性。

2 RSA 算法与故障攻击模型

2.1 RSA-CRT 算法

RSA 密码算法的安全性是基于大整数因式分解的困难性问题。因式分解 $n=pq$，等价于计算私钥 $d\in\varphi(n)$，$\varphi(*)$ 表示欧拉函数。相应于私钥 d 的公钥为：$e=d^{-1}\bmod\varphi(n)$。RSA 签名消息 m 的计算结果为 $S=m^d \bmod n$，通过比较计算 $\tilde{m}=S^e \bmod n$ 和 m 进行验证。为提高 RSA 执行效率，采用中国剩余定理实现的 RSA-CRT 算法，描述如算法 1 所示。

算法 1 RSA-CRT 算法

输入：消息 m，参量 p,q,d,n；

输出：S

1. $S_p=m^{d \bmod (p-1)} \bmod p$
2. $S_q=m^{d \bmod (q-1)} \bmod q$
3. $c_p=q(q^{-1} \bmod p)$，$c_p=q(q^{-1} \bmod p)$
4. $S=CRT_{(p,q)\to n}(S_p,S_q)=c_p.S_p+c_q.S_q \bmod n$

由算法 1 可知，通过 $m^d=CRT_{(p,q)\to n}(S_p,S_q)$ 可以快速实现 RSA 签名，其中 $S_p=M^{d \bmod (p-1)} \bmod p$ 等价于 $M^d \bmod p$（对于 S_q 情况也一样）。该算法比一般平方乘实现算法的执行效率大约提高了 4 倍。

2.2 RSA-CRT 故障分析模型

RSA-CRT 算法主要分为模幂运算和组合运算两个阶段，基于模幂运算产生故障的攻击模型如图 1 所示，虚线框中给出模幂运算可能发生故障的位置，原有的攻击算法大都是基于该模型的[2,3]，而本文主要关注组合运算阶段注入故障的情况。

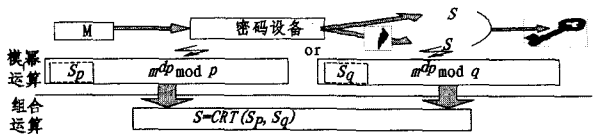

图 1 基于模幂运算的 RSA-CRT 故障攻击模型

Dan Boneh 等人[2]指出 RSA-CRT 算法只要其中的一个幂运算（S_p 或 S_q）发生错误，则模数 N 就可能通过故障签名 $\hat{S}$ 分解。假设计算 S_p 时发生错误，$\hat{S}_p$ 为故障中间值，那么可以利用错误签名 $\hat{S}=\mathrm{CRT}(\hat{S}_p,S_p)$ 和正确的签名 S 通过计算 $\mathrm{GCD}(S-\hat{S},N)$ 获取素数 q。

Arjen Lenstra 扩展了上面的攻击算法，提出只需要一个故障的签名即可通过 $\mathrm{GCD}(\hat{S}^e-m,N)$ 获取素数 q，成功因式分解模数 n，最终推算出密钥 d[3]。

3 基于 CRT 组合运算故障的攻击模型

3.1 Shamir 的故障攻击防御算法

简单的故障分析的防御算法：一种是在最后通过对同一消息进行两次签名，对比两次签名的结果是否一致，再决定是否输出结果；另一种是最后通过采用公钥对签名进行验证，再决定是否输出结果，由于公钥验证过程往往比签名过程快得多，因此其带来的额外时间开销比第一种小得多。但仅仅进行运算检验并不足以防止故障。Shamir 针对 RSA 算法的故障攻击方式，给出了另外一种防御算法[7]。算法的主要思想是：利用 S_{pr} 和 S_{qr} 的额外计算在 CRT 组合运算执行前进行故障检测，选择一个小的与 n 互质的大约 32 位随机素数 r，通过计算验证其正确性来决定是否执行 CRT 组合运算，从而避免错误签名的输出，算法描述如下。

算法 2 Shamir 防御算法

输入：消息 m，参量 p,q,d,r；

输出：S

1. $S_{rp}=m^d \bmod rp$；
2. $S_{rq}=m^d \bmod rq$；
3. if($S_{pr} \bmod r\neq S_{qr} \bmod r$) Return Error；
4. Else
5. Return $S=CRT_{(p,q)\to n}(S_{rp} \bmod p,S_{rq} \bmod q)$

可以发现 Shamir 算法仅仅对 S_p 和 S_q 运算进行检查，只能够保护两个素数 p 或 q 进行模运算时出故障的情况，而没有考虑 CRT 组合运算步骤产生故障的情况，因此该防御算法仍然存在遭受故障分析的可能。

3.2 故障分析模型

由于 Shamir 防御算法已经对 S_p 和 S_q 的运算进行了错误检测，避免了错误签名的输出，使得原有的攻击算法失效。但该算法由于在执行过程中直接采用指数 d，仍然存在组合运算步骤注入故障执行攻击的可能性。

本文在图 1 攻击模型的基础上，建立基于 CRT 组合运算步骤注入故障的分析模型，如图 2 所示。

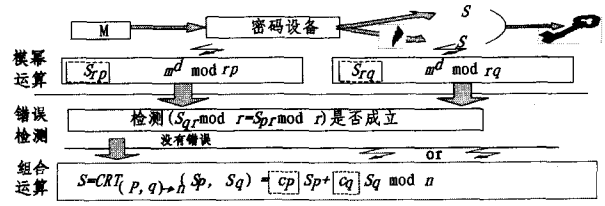

图 2 基于组合运算故障的 RSA 故障分析模型

防御算法主要由 3 个部分组成，分别是模幂运算、错误检验和组合运算。图 2 虚线框中给出 CRT 组合运算阶段可能发生故障的 2 个位置，本文考虑其中一个参数发生故障的情况，将注入故障类型按影响程度分为永久故障、瞬时故障两种[5]，下面依据分析模型，对攻击算法做进一步分析。

3.3 故障注入

通过对算法 2 的分析发现，Shamir 的防御算法仅对 S_p 和 S_q 发生错误的情况进行检测，而当组合运算发生故障时，防御算法则检测不到，导致故障签名输出。排除 S_p 和 S_q 发生故障的情况，容易受故障影响的变量是组合运算步骤 $S=$

$CRT_{(p,q)\to n}(S_p,S_q)=c_p\cdot S_p+c_q\cdot S_q \bmod n$ 中的变量 c_p 和 c_q。假设设 $m\neq 0$ 为签名消息，$S=m^d \bmod n$ 为 RSA 签名，$\tilde{S}$为引入故障 δ 的错误签名结果，$\delta<n$，假设 c_q 为受故障影响的变量(相对 c_p 注入故障的分析算法是一样的)。设 k_q,k_p 为满足 $0\leqslant m^d-k_qq<q$ 和 $0\leqslant m^d-k_pp<p$ 的自然数，那么错误的签名 $\tilde{S}$ 可以表示为：

$$\begin{aligned}\tilde{S}&=((c_q+\delta)(m^d \bmod q)+c_p(m^d \bmod p)) \bmod n\\&=((c_q+\delta)(m^d-k_qq)+c_p(m^d-k_pp)) \bmod n\\&=(c_q(m^d-k_qq)+c_p(m^d-k_pp))+\delta(m^d-k_qq) \bmod n\\&=(1+\delta)m^d-\delta k_qq \bmod n\\&=((1+\delta)m^d-q(\delta k_q)) \bmod n\end{aligned}\tag{1}$$

即 $\tilde{S}=((1+\delta)S-q(\delta k_q)) \bmod n$。

3.4 密钥分析算法

3.4.1 针对已知特定故障的攻击算法

假设攻击者能够控制故障的情况，包括时机、位置和大小等参数，使得 $\delta=0 \bmod p$，即故障 $\delta=kp$，由于 $n=pq$，那么 $q(\delta k_q)) \bmod n=0$，$\tilde{S}-S=(\delta S-q(\delta k_q)) \bmod n$，通过计算 $gcd(\tilde{S}-S,n)=p$ 因式分解模数 n，再利用 $d=e-1 \bmod (p-1)(q-1)$计算出私钥 d。

如果故障 δ 已知，当 $\delta\neq 0 \bmod p$，$k_q\neq 0 \bmod p$ 且 $m\neq 1 \bmod n$ 时，可通过计算 $gcd(\tilde{S}-(1+\delta)S,n)=q$ 因式分解模数 n，直接计算出私钥 d。

3.4.2 针对未知随机故障的攻击算法

由于实际执行攻击过程中攻击者受条件和能力的限制，大多数情况下不可能准确地控制故障产生的位置、时机和大小，因此故障绝大多数是随机的。下面对 δ 值未知的情况进行讨论，并给出相应的攻击算法。

(1) δ 为永久故障的情况

对于 δ 未知时，给出扩展攻击算法，将式(1)重写为：

$$\frac{\tilde{S}}{S}=(1+\delta)-\frac{q(\delta k_q)}{S} \bmod n\tag{2}$$

假设 S_1 和 S_2 表示为两个正确的签名，$\tilde{S}_1$ 和 $\tilde{S}_2$ 表示对应的两个错误签名，两次注入的故障 δ 不变，相当于永久故障，那么由式(2)可得：

$$\frac{\tilde{S}_1}{S_1}-\frac{\tilde{S}_2}{S_2}=-q(\frac{\delta k_{q1}}{S_1}+\frac{\delta k_{q2}}{S_2}) \bmod n\tag{3}$$

那么当 $\frac{\delta k_{q1}}{S_1}+\frac{\delta k_{q2}}{S_2}\neq 0 \bmod p$ 时，可以直接通过计算公式 $gcd((\frac{\tilde{S}_1}{S_1}-\frac{\tilde{S}_2}{S_2}),n)=q$ 获取 q，因式分解模数 n。

而当 $\frac{\delta k_{q1}}{S_1}+\frac{\delta k_{q2}}{S_2}=0 \bmod p$ 时，则不能通过该算法分解模数。该情况下，由于 δ 与 P 互质，则有 $\frac{k_{q1}S_2}{S_1}=k_{q2} \bmod p$，等式中的所有变量依赖于 d，p 和 q，因此它们对攻击者来说是未知的，对选定的 m_1、k_{q1}、m_2，都存在一个 $k_{q2}\in Z_p$ 满足上面的等式。但 p 是一个 512 位的大素数时，要寻找满足该等式的变量的概率几乎为零，因此需要尝试寻找另外一种攻击算法。

以 Arjen Lenstra 扩展的故障分析算法思想[3]为依据，将式(1)重写为：

$$\tilde{S}^e=((1+\delta)S-q(\delta k_q))^e \bmod n\tag{4}$$

设两个不同的签名消息为 m_1、m_2 以及对应的错误签名结果 $\tilde{S}_1$ 和 $\tilde{S}_2$，和整数 $\Delta\in Z_p$。由于 $S^e \bmod n=m$，因此有：

$$\tilde{S}_i^e=(1+\delta)^em_i-q(\Delta) \bmod n\tag{5}$$

式中，$i\in\{1,2\}$，$q(\Delta)$表示 $\tilde{S}^e$ 中除$(1+\delta)^em_i$ 之外其他含 q 的项。

所以有 $\tilde{S}_1^em_2-\tilde{S}_2^em_1=q(\Delta)(m_1-m_2) \bmod n$，假如 $\Delta\neq 0 \bmod p$，则可以通过计算 $gcd(\tilde{S}_1^em_2-\tilde{S}_2^em_1,n)=q$ 因式分解模数 n。

(2) δ 为随机故障的情况

当每次注入故障值 δ 是随机时，需将式(3)变化为：

$$\frac{\tilde{S}_1}{S_1}-\frac{\tilde{S}_2}{S_2}-(\delta_1-\delta_2)=-q(\frac{\delta_1k_{q1}}{S_1}+\frac{\delta_2k_{q2}}{S_2}) \bmod n\tag{6}$$

由于 δ_1 与 δ_2 都为随机值，因此 $\frac{\delta_1k_{q1}}{S_1}+\frac{\delta_2k_{q2}}{S_2}=0 \bmod p$ 的概率很小，不考虑该情况的分析。利用式(6)攻击者通过穷举遍历 $\delta_1-\delta_2$ 的值 $\Delta\delta$，计算 $gcd((\frac{\tilde{S}_1}{S_1}-\frac{\tilde{S}_2}{S_2}-\Delta\delta),n)=q$ 因式分解模数 n。实际执行攻击时可对同一个消息 m 执行签名，再将两次不同故障对应的错误签名 $\tilde{S}_1$ 和 $\tilde{S}_2$ 代入 $gcd((\frac{\tilde{S}_1-\tilde{S}_2}{S}-\Delta\delta),n)$计算是否存在最大公约数。

假设对 C_q 注入的故障为随机字节故障，那么设 R^8 表示随机字节，l 表示故障位置，则有 $R^8\in[1;2^8-1]$，$l\in[0;\frac{bits(q)}{8}-1]$，那么 $\delta=R^8\cdot 2^{8l}$ 表示故障值的大小。可知故障 $\delta_1-\delta_2$ 的值 $\Delta\delta$ 具有 $2^{bits(q)}$ 的搜索空间，导致攻击的复杂度极高，因此，在实际执行攻击过程中，攻击者可尝试寻找现在同一个字节位置的两个故障，使之只有一个字节的差异，从而可以缩小 $\Delta\delta$ 的搜索空间为 R^8，攻击流程如图 3 所示。

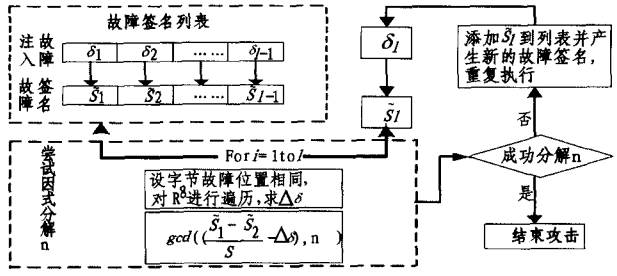

图 3 故障攻击算法执行流程图

每次新增一个故障签名 $\tilde{S}_l$，假设其故障位置为某个字节 i，与现有的故障列表中的签名尝试因式分解计算，如果都没满足，再尝试其它字节位置，如果都没用满足则产生新的故障签名，重复以上执行过程，直到因式分解 n。

4 实验结果及讨论分析

4.1 仿真实验结果及算法复杂度分析

表 1 故障分析仿真结果

故障类型	故障条件	故障数目	样本空间
永久故障	$\delta=0 \bmod p$	1	1
	$\delta\neq 0 \bmod p$	1	1
	$\frac{\delta k_{q1}}{S_1}+\frac{\delta k_{q2}}{S_2}\neq 0 \bmod p$	2	1
	$\frac{\delta k_{q1}}{S_1}+\frac{\delta k_{q2}}{S_2}=0 \bmod p$	2	1
随机故障	$\frac{\delta_1 k_{q1}}{S_1}+\frac{\delta_2 k_{q2}}{S_2}\neq 0 \bmod p$	2～64	$1\sim 2^{512}$

根据本文前面的故障分析算法，在 Windows xp 环境下，在普通 PC 机(CPU：AMD Athlon 64 3000+，内存：1G)上使用 VC++ 6.0 调用 OpenSSL 密码库的库函数实现 Shamir 防御算法，并设计仿真实现本文提出的故障攻击算法。利用软件模拟故障诱导过程，通过在 CRT 组合运算步骤对系数 C_q(或 C_p)注入永久故障或随机字节故障，获取对应的故障签名 $\tilde{S}$，依据攻击算法尝试因式分解模数 n，进而计算推导私钥 d。假设 RSA 密钥长度为 1024 位，表 1 给出注入不同故障对应的实验仿真结果。

对于永久故障攻击，可以在注入故障之前实现获取正确的签名，用于后面的分析。

4.2 故障分析算法的进一步讨论分析

新的攻击算法都是在假设的条件下论证得到的。在正确的位置和时机注入故障，目前主要采用的注入故障技术有剥脱技术(如用探针直接破坏电路)、腐蚀技术、激光注入故障技术、电磁场引导故障技术等，由于篇幅限制，本文不做详细论述，具体可参见文献[5]。针对 Shamir 防御算法的攻击算法，本文仅针对在组合运算过程中对 c_p 和 c_q 中的一个变量发生故障的情况进行分析，而当两个变量都发生故障时，即二阶故障的情况，则本文的攻击算法也会失效。假设此时设 c_q 引入的暂时故障为 δ_1，c_p 引入的暂时故障为 δ_2，那么式(1)可重写为：

$$\begin{aligned}\tilde{S} &= ((c_q+\delta_1)(m^d \bmod q)+(c_p+\delta_2)(m^d \bmod p)) \bmod n\\ &= ((c_q+\delta_1)(m^d-k_qq)+(c_p+\delta_2)(m^d-k_pp)) \bmod n\\ &= (c_q(m^d-k_qq)+c_p(m^d-k_pp))+\delta_1(m^d-k_qq)+\delta_2(m^d-k_pp) \bmod n\\ &= (1+\delta_1+\delta_2)S-\delta_1k_qq-\delta_2k_pp \bmod n \qquad (7)\end{aligned}$$

此时只有满足 $\delta_1=0 \bmod p$，$\delta_2\neq 0 \bmod p$(两种故障互为等效)且都已知时，才可以通过计算 $gcd(\tilde{S}-(1+\delta_1+\delta_2)S,n)=p$ 因式分解 n。对于其他情况，由于采取前面其他的算法计算都无法消除 p 和 q 其中的一个，导致原有故障攻击失效。

4.3 防御建议

先前抵御 RSA-CRT 故障攻击的防御算法大多可以分为 3 个部分：S_p 和 S_q 两个幂运算、计算签名 S 的组合运算和故障检验的步骤，后者又分为两种情况，一种是采用条件检验是否有故障发生，如 Shamir 的防御算法，假如发生故障则不执行组合运算，不输出签名；另一种产生随机数，假如有故障产生则用随机数替代签名，否则输出正确的签名。假如攻击者能够在前面的操作步骤注入故障，且跳过检验步骤或检验步骤出错，则仍然可以实施故障攻击。本文对在组合运算步骤出故障的情况进行了分析，证明了 Shamir 算法并不能有效地抵御故障分析，主要原因是原有的防御措施只是在模幂运算之后进行错误检验，而没有对组合运算之后的签名结果进行检测，另外私钥 d 直接参与计算也是其容易遭受攻击的因素之一。密钥分析算法一般都是通过注入故障和计算 $f()$得到 $f(\tilde{S})=\delta+q\Delta \bmod n$ 或 $f(\tilde{S})=\delta+p\Delta \bmod n$，再通过其它的组合计算 $F()$获得 $gcd(F(f(\tilde{S})),n)=p$ 或 q，从而因式分解 n。

结合攻击算法分析思路，本文提出一种可行的防御措施建议：核心算法是让最终签名结果的计算在故障检验步骤之后进行。具体为：首先在组合步骤之前产生随机数，计算出一个中间结果值 S^*，这使和 p 或 q 有关的因素都被盲化，然后执行故障检验步骤，假如有故障，则不输出结果，否则再通过计算消除随机数，并输出正确的签名 S。

结束语 本文介绍了基于模幂运算故障的 RSA-CRT 攻击算法以及 Shamir 防御算法，通过分析发现 Shamir 防御算法仍然存在遭受故障分析的可能性，建立了新的基于组合运算故障的 RSA 故障攻击模型。对注入故障出现的不同情况给出相应的分析算法，并进行理论论证和实验仿真，结果表明，给出的攻击算法除随机故障的情况外，最多只需要两个故障即可破解密钥；同时给出针对随机故障情况降低密钥搜索空间的方案。为了更全面地讨论攻击算法，针对可能同时出现两个故障的情况作进一步分析，最后总结故障攻击的核心思想，分析了原有防御算法的不足，提出防御建议。

基于组合运算故障是针对 RSA-CRT 故障分析的一种新的思路，但同时由于实际执行攻击时，故障出现的情况是多变的，攻击者很难精确控制故障的时机、位置及大小等参数，因此关于如何在适当的位置和时机注入故障等方面还有待进一步研究。为抵御新的防御算法设计一种有效的防御算法也是将来值得研究和关注的。

参 考 文 献

[1] Kocher P. Timing attack on Implementations of Diffie-Hellman, RSA, DSS, and Other systems[C]// Proceedings of Advances in Cryptology-CRYPTO'96. Berlin: Springer-verlag, 1996: 104-113

[2] Boneh D, DeMillo R, Lipton R. On the Importance of Checking Cryptographic Protocols for Faults[J]. Advances in Cryptology—Eurocrypt'97, 1997, 1233(12): 37-51

[3] Lenstra A K. Memo on RSA Signature Generation in the Presence of Faults[EB/OL]. September 1996. http://cm.bell-labs.com/who/akl/

[4] Aumuller C, Bier P, Fischer W, et al. Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures[C]// CHES 2002. Berlin: Springer, 2003: 260-275

[5] Schmidt J-M. Differential Fault Analysis[R]. Austria: IAIK Lab in Austria, 2008

[6] Dottax E, Giraud C, Rivain M, et al. On Second-order Fault Analysis Resistance for RSA-CRT Implementations: Information Security Theory and Practice[C]// LNCS. Berlin Heidelberg, 2009: 68-85

[7] Shamir A. How to Check Modular Exponentiation [J]. EUROCRYPT' 97. Springer-Verlag, 1997

[8] Blömer J, Otto M, Seifert J-P. A New RSA-CRT Algorithm Secure against Bellcore Attacks[C]// Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003. Washington, DC, USA: ACM, 2003: 311-320