DOI:10.15943/j.cnki.fdxb-jns.2013.03.003

第 52 卷 第 3 期 复 旦 学 报（自然科学版） Vol. 52 No. 3
2013 年 6 月 Journal of Fudan University (Natural Science) Jun. 2013

文章编号：0427-7104(2013)03-0297-06

对称加密算法 AES 和 DES 的差分错误分析

孙维东，俞 军，沈 磊

（复旦大学 微电子研究院，专用集成电路与系统国家重点实验室，上海 200433）

摘 要：分析对称加密算法 AES 和 DES 对差分错误分析的安全性. 描述了一种针对 AES 加密算法的差分错误分析方法，通过软件模拟成功恢复根密钥. 实验结果表明，只需 20 次左右的错误注入就能实现 AES 差分错误分析. 提出一种新的 DES 差分错误分析方法，该方法与 AES 差分错误分析在原理上类似，软件模拟结果表明破解 DES 需要的错误注入次数更多. 因此无任何防护手段的 AES 和 DES 加密算法很容易受到差分错误分析的攻击. 最后提出引入错误侦测机制能有效抵御此类攻击.

关键词：高级加密标准；数据加密标准；差分错误分析；智能卡

中图分类号：TP 309.7 **文献标志码**：A

错误攻击最早是由 Boneh，Demillo 和 Lipton 在 1997 年提出的[1]，这种攻击方法通过人为地对加密算法注入错误，从而产生错误加密结果，然后分析正确以及错误的加密结果来破解加密算法的密钥. Biham 和 Shamir 在 1997 年提出对 DES 加密算法实行错误攻击[2]，并称之为差分错误分析. 2003 年，Dusart 和 Giraud 分别提出两种不同的 AES 差分错误分析方法，前者错误注入在中间状态[3]，而后者错误注入在密钥[4]. Skorobogatov 和 Anderson 在 2003 年首次通过光照方法实现了 AES 的差分错误分析[5].

AES 的数据分组长度为 128 比特，密钥长度可以为 128、192 以及 256 比特，不同的密钥长度分别对应不同的加密轮数：10、12、14 轮. 数据以 4×4 状态矩阵的形式分为 16 字节进行循环加密，每 1 轮加密包含 4 个运算：SubBytes、ShiftRows、MixColumns 以及 AddRoundkeys. 每轮加密所需的轮密钥提前由根密钥经过密钥扩展运算生成. 图 1 是 AES 加密流程图.

DES 将长度为 64 比特的输入数据通过 16 轮的迭代运算，得到长度同样位 64 比特的输出数据，每轮迭代使用的 64 比特子密钥中的 48 比特是由原始 56 比特根密钥产生的，因此 DES 算法也有密钥扩展过程，每轮子密钥中有 8 比特奇偶校验位以及 8 比特随机位. DES 算法系统分为 4 个部分：子密钥生成、初始位变换、F 变换以及逆初始位变换. 其中 F 变换又包含以

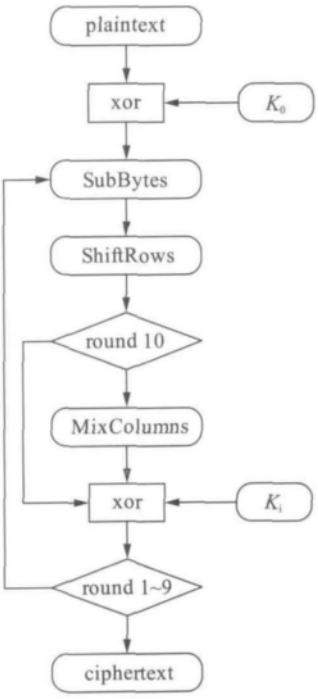

图 1 AES 加密流程图

Fig. 1 Encryption process of AES

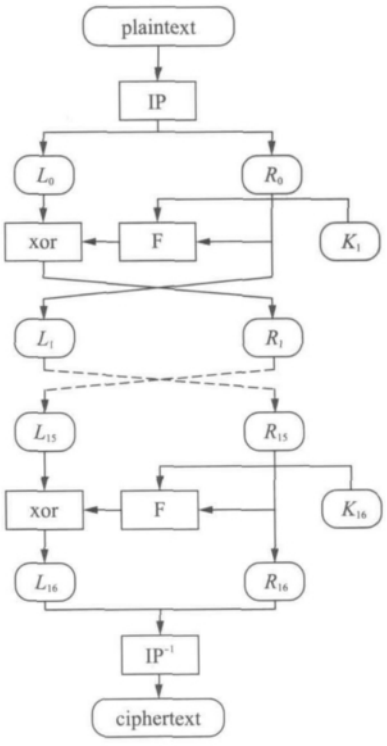

图 2 DES 加密流程图

Fig. 2 Encryption process of DES

收稿日期：2012-05-14

作者简介：孙维东(1986—)，男，硕士研究生；沈 磊，男，高级工程师，通讯联系人，E-mail：shenlei@fmsh.com.cn.

下运算：E 变换、S 盒变换以及 P 变换. 图 2 是 DES 的加密流程图.

1 差分错误分析原理

上世纪九十年代，一种称为光攻击的物理攻击方法被提出并得到了广泛的发展. 这种攻击方法是指用光照射正在工作的芯片表面，由于光的入侵，智能卡芯片的硅片内部会产生电压和电流，从而导致失效行为. 采用这种技术的攻击者可以绕过加密芯片的密码或 PIN 码得到私密数据，或者对加密操作进行攻击从而产生出密钥数据，严重影响了加密芯片的安全性能.

差分错误分析正是在光攻击基础上发展起来的通过人为地对加密算法注入错误，从而产生错误加密结果，然后分析正确以及错误的加密结果来破解加密算法的密钥. 对于注入的错误来说，可以是单比特数据的错误，也可以是多比特数据的错误，实际应用过程当中显然后者的情况更加普遍. 如果能将错误注入控制在同个字节中，也就实现了对单字节数据的错误注入.

2 AES 差分错误分析过程

假设采用最简单的 AES 加密方式，密钥长度 $N_k=4$，分组大小 $N_b=4$，加密轮数 $N_r=10$. 那么错误注入在第 9 轮 ShiftRows 之后，任意选取该中间状态中的某个字节异或一个非 0 的错误 ε，经过随后的加密过程得到错误的密文输出. 该错误密文与正确密文将会有 4 个字节不相同，图 3 是错误传播的过程，图中每 1 格代表中间状态(state)的一个字节(byte)，16 个字节的数据构成了上述 AES 加密方式的中间状态，阴影格子代表该字节注入了错误. 图中可以看出，在第 9 轮 ShiftRows 之后的中间状态上，错误注入在第一个字节 B_0，然后经过第 9 轮的 MixColumns 变换，错误扩散至 4 个字节 B_0、B_1、B_2、B_3，随后经过第 9 轮 AddRoundkey 以及第 10 轮 SubBytes 变换，错误仍处于该 4 个字节，而第 10 轮的 ShiftRows 变换之后，错误传递至 B_0、B_7、B_{10}、B_{13} 这 4 个字节，随后的变换过程该 4 个错误字节的位置将不再发生变化，也就是说密文同样是 B_0、B_7、B_{10}、B_{13} 这 4 个字节将含有错误.

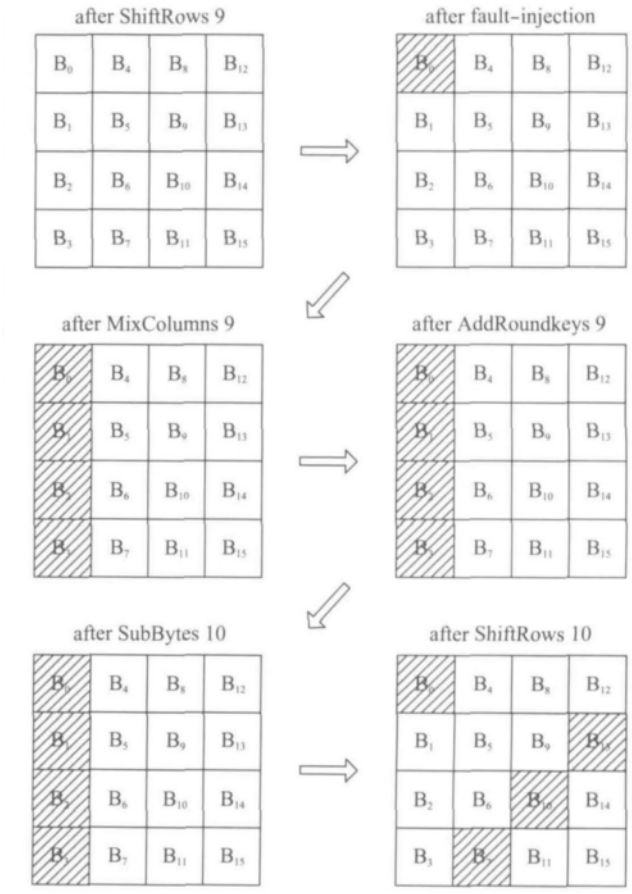

图 3 错误传播过程

Fig. 3 Fault propagation

假设用等式(1)来表示第 9 轮 ShiftRows 之后错误的中间状态，等式左边是错误的中间状态(F)，等式右边是正确的中间状态(S)异或错误(ε). 那么错误传递过程中错误的中间状态就如以下等式所示.

第 9 轮 ShiftRows 之后

$$F_9=S_9\oplus\begin{pmatrix}\varepsilon&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix},\qquad(1)$$

第 9 轮 MixColumns 之后

$$F_9=S_9\oplus\begin{pmatrix}2\varepsilon&0&0&0\\\varepsilon&0&0&0\\\varepsilon&0&0&0\\3\varepsilon&0&0&0\end{pmatrix},$$

第 9 轮 AddRoundkey 之后

$$F_9 = S_9 \oplus \begin{pmatrix} 2\varepsilon & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 \\ 3\varepsilon & 0 & 0 & 0 \end{pmatrix},$$

第 10 轮 SubBytes 之后

$$F_{10} = S_{10} \oplus \begin{pmatrix} \varepsilon_0 & 0 & 0 & 0 \\ \varepsilon_1 & 0 & 0 & 0 \\ \varepsilon_2 & 0 & 0 & 0 \\ \varepsilon_3 & 0 & 0 & 0 \end{pmatrix},$$

第 10 轮 ShiftRows 之后

$$F_{10} = S_{10} \oplus \begin{pmatrix} \varepsilon_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_1 \\ 0 & 0 & \varepsilon_2 & 0 \\ 0 & \varepsilon_3 & 0 & 0 \end{pmatrix},$$

加密结束之后

$$Cipher' = Cipher \oplus \begin{pmatrix} \varepsilon_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_1 \\ 0 & 0 & \varepsilon_2 & 0 \\ 0 & \varepsilon_3 & 0 & 0 \end{pmatrix}. \tag{2}$$

最后错误的密文输出可以用等式(2)表示，很显然，ε_0、ε_1、ε_2、ε_3 **的值可以通过异或正确和错误密文得到. 观察第 10 轮 Subbytes 变换前后，原本未知的错误值** ε **经过 Subbytes 变换得到了已知的** ε_0、ε_1、ε_2、ε_3，**由于 Subbytes 变换可以看成非线性的 S 盒变换，因此可以通过遍历的方式求解，用 S 表示该 S 盒变换，得到等式(3)所示的方程组. 式中** X_0、X_1、X_2、X_3 **分别表示第 10 轮 Subbytes 之前中间状态对应 4 个字节的值**

$$\begin{cases} S(X_0 + 2\varepsilon) = S(X_0) + \varepsilon_0, \\ S(X_1 + \varepsilon) = S(X_1) + \varepsilon_1, \\ S(X_2 + \varepsilon) = S(X_2) + \varepsilon_2, \\ S(X_3 + 3\varepsilon) = S(X_3) + \varepsilon_3. \end{cases} \tag{3}$$

实际情况下 ε **的值是未知的，对于上述方程组中的第一个方程，必须同时遍历** ε **和** X_0**. 由于两者都是字节类型数据，因此每个方程遍历** 256×256 **次，然后得到满足该方程的**(ε, X_0)**可能值组合构成的集合. 同样的，对于其余的 3 个方程也可以得到**(ε, X_1)、(ε, X_2)、(ε, X_3)**可能值组合构成的集合. 而对于同一次错误注入，**ε **的值是唯一的，那么通过这 4 个集合中** ε **值的交集可以进一步缩减 4 个集合元素个数. 另外，既然得到了中间状态** X_0、X_1、X_2、X_3 **的可能值，则根据等式(4)**

$$S(X_0) \oplus K_{10}(0) = C(0), \tag{4}$$

第 10 轮密钥的对应字节可以很简单地通过中间状态异或密文得到. 也就是说，通过 1 次错误注入得到了第 10 轮密钥中 4 个字节的所有可能值，而 ε **的值具体是多少这并不重要.**

重复进行错误注入和上述整个过程，直到筛选出最终正确的末轮密钥的 4 个字节. 随后在该中间状态的不同字节上注入错误时，又能得到末轮密钥的其余字节的值，最后得到完整的末轮密钥.

总结上述密钥传播的规律可以发现，在 B_0 **注入错误时，错误传递至** B_0、B_7、B_{10}、B_{13} **四字节，最后恢复了密钥的第 1、8、11、14 字节；假如在** B_4 **注入错误时，错误传递至** B_1、B_4、B_{11}、B_{14} **四字节，可以恢复密钥的第 2、5、**

12、15 字节;假如在 B_8 注入错误时,错误传递至 B_2、B_5、B_8、B_{15} 四字节,可以恢复密钥的第 3、6、9、16 字节;假如在 B_{12} 注入错误时,错误传递至 B_3、B_6、B_9、B_{12} 四字节,可以恢复密钥的第 4、7、10、13 字节,如图 4 所示.

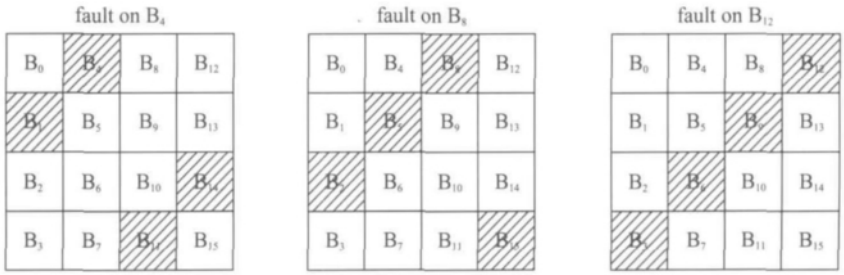

图 4 不同位置注入后错误传播情况

Fig. 4 Fault propagation after fault-injection on different positions

3 DES 差分错误分析过程

DES 的差分错误分析方法与 AES 类似,其错误注入点也位于倒数第 2 轮的中间状态.但与 AES 多采用字节数组处理数据不同,DES 算法大量地应用了比特位作为单位来处理数据,如位置换等等,单字节的错误很容易扩散至多字节,并且由于数据经过 DES 的 S 盒长度会缩减,这些都给对错误攻击的分析带来很大困难.因此,很难通过假设注入单字节错误来进行攻击.

虽然实际操作中可能较难实现,假设对于 DES 加密算法可以注入单比特的错误,那么将一个随机错误 ε 注入到 DES 加密过程中倒数第 2 轮(即 15 轮)迭代结束之后的中间状态上,导致该中间状态的某个比特发生错误,该错误的具体数值未知并随着之后的加密过程扩散到其他比特.由于注入比特位位置的不同,最后的密文将产生多个比特的错误数据.具体的扩散过程会在下个章节中详细介绍.

假设对于 DES 加密方式,错误注入在第 15 轮迭代之后,任意选取该中间状态中 R_{15} 的某个比特并取反.因为 $R_{16}=R_{15}$,所以 R_{16} 的同 1 比特也发生错误,而 L_{16} 是 R_{15} 经过 F 变换得到,F 变换中的 E 变换有可能将单比特错误扩散为 2 比特,然后经过非线性的 S 盒变换,此时就可能有 1 个或者 2 个 S 盒的输出产生错误,随后的逆 IP 置换只会改变比特的位置而不会增加错误的比特数.因此,对错误密文与正确密文都进行 IP 变换恢复出各自对应的 L_{16} 和 R_{16} 之后,再将 L_{16} 进行逆 P 变换,将很容易观察到预想的 1 个或者 2 个 S 盒的错误.图 5 是错误传播的过程,图中是 DES 加密的第 16 轮迭代过程,阴影表示该部分数据注入了错误.图 6 是 F 变换的错误传递过程.

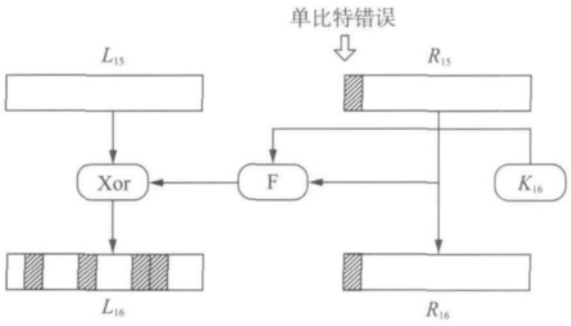

图 5 DES 错误传递

Fig. 5 Fault propagation of DES

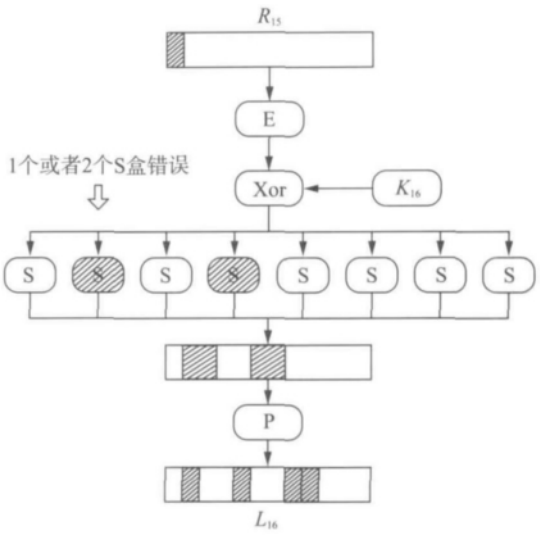

图 6 F 变换错误传递

Fig. 6 Fault propagation in function F

通过上述分析，存在以下等式：

$$F(R_{15})\oplus F(R_{15}')=R_{16}\oplus R_{16}', \tag{5}$$

$$S(R_{15}\oplus K_{16})\oplus S(R_{15}'\oplus K_{16})=R_{16}\oplus R_{16}', \tag{6}$$

$$S(L_{16}\oplus K_{16})\oplus S(L_{16}'\oplus K_{16})=R_{16}\oplus R_{16}'. \tag{7}$$

对于等式(7)，除了 K_{16} 外其余均为已知量，因此同 AES 类似，可以通过遍历对应 S 盒可能情况的方法来求解满足等式 7 的所有 K_{16} 可能值. 那么经过多次的错误注入，就能筛选出 K_{16} 的正确值. 值得注意的是每个 S 盒对应可以求解 K_{16} 的 6 比特值，如果在 R_{15} 的其余适当的比特位注入错误，则最后可以得到完整的 K_{16}. 同样的，可以通过 DES 的根密钥反推算法求得 DES 算法的根密钥.

4　软件模拟

使用软件模拟 AES 差分错误分析方法，编写 AES 加密算法的 C 程序，并用软件模拟在第 9 轮 ShiftRows 之后的中间状态注入单字节错误，假设采用 AES-128 加密方式，明文和密钥如下：

明文为 00112233445566778899aabbccddeeff，根密钥为 000102030405060708090a0b0c0d0e0f.

根据明文和根密钥计算出正确密文以及各轮子密钥，其中需要末轮密钥来对比破解结果：

正确密文为 69c4e0d86a7b0430d8cdb78070b4c55a，末轮密钥为 13111d7fe3944a17f307a78b4d2b30c5.

首先在 B_0 进行第 1 次错误注入，错误值为 2e，错误密文为 ebc4e0d86a7b0493d8cda7807022c55a，随后用差分错误分析计算出末轮密钥中的第 1 个字节 K_0 可能值如下：

{3b,b9,59,db,c2,40,68,ea,13,91,70,f2,12,90,c3,41,a0,22,89,0b,71,f3,3a,b8,eb,69,58,da,23,a1}；

随后在 B_0 进行第 2 次错误注入，错误值为 8f，错误密文为 7fc4e0d86a7b04bed8cdaa8070d3c55a，用该错误密文计算出 K_0 可能值，经过筛选，K_0 可能值的范围缩减为{f2,12,ea,eb,13,0b,f3}；

最后进行第 3 次错误注入，错误值为 ee 错误密文结果为 c9c4e0d86a7b04edd8cde88070e6c55a，K_0 可能值{13}，该值正好是末轮密钥第 1 个字节的正确值. 依此类推，末轮密钥的其余字节也可以用同样的方法恢复，得到完整的末轮密钥之后，再用根密钥反推算法得到正确的根密钥，该算法可以参考文献[3]附录，至此完成了 1 次完整的 AES 差分错误分析.

前文提到在 B_0 注入错误时，可以恢复密钥的第 1、8、11、14 字节，而软件模拟证明只需要 3 次错误注入在 B_0，就能恢复密钥的第 1 字节，同时也能恢复第 8、11、14 字节，经过完整的差分错误分析发现，平均只需要 3～5 次错误注入就能恢复末轮密钥的 4 个字节，那么恢复完整的末轮密钥需要的错误密文数大约为 15～20 条.

DES 差分错误分析的软件模拟方式与 AES 类似，在此不作赘述，试验结果表明，DES 需要比 AES 更多的错误注入来恢复根密钥.

5　差分错误分析防护

光攻击主要利用光注入产生的热、电等效应导致电路损坏或者出错，而其物理级防护实际上为检验或者增加噪声容限，大部分能够从根本上防护光攻击的方案均需从版图层次出发.

对电源和输入信号的过滤作为第一道屏障：快速反应稳压器可以防止电压的瞬变，也可防止时钟供应的反常. 传感器被作为第二道防线. 例如，一个安全控制器被很高的电压攻击，如果传感器监测到临界值，则智能卡会触发一个警报进入安全状态. 电压传感器检查电源供应，时钟传感器监测频率异常，温度和光传感器检查光攻击和热攻击. 因为光攻击也能够通过芯片背面进行，所以光传感器的布置就不是仅限于前面的照射. 第三道屏障是在设计安全控制器内核时建立的. 硬件和软件相结合的策略被用作有效的第三道屏障. 既然纯软件对策在某些情况下能够成为失效分析的目标，则硬件和软件的结合就成为必不可少的.

上述方法都是从物理角度来抵御差分错误分析，在系统角度上也存在某些方法，主要方案分为两类：其一为冗余校验，采用两种或者更多的方案来计算某个关键地方的值，然后校验这些方案的运算结果来达到错误侦测的目的，该防护的缺陷是额外的面积开销；其二为回旋校验，在加密的同时做一次解密运算，然

后对比加密结果同明文是否相同,该防护利用对称加密算法加解密结构几乎相同的特性节省了额外的面积开销,但是缺陷是芯片加密的时间加倍和速度变慢;其三为数据位校验,比如用在信号传输过程中的奇偶校验,而对于处理器来说,在运算过程中的校验也有 F 校验、9 校验等,该防护的错误侦测覆盖率高,但是也有额外的面积开销.

从安全性角度来看,对芯片加入防护手段是必须的,但是实际芯片生产过程中则还需要兼顾芯片面积和计算时间等方面.因此,实际设计过程中我们必须综合多方面的考虑,最后采取一个兼顾芯片主要性能以及安全性能的折中方案,从而达到最好的实际效果.

差分错误分析越来越成为影响智能卡加密算法安全性能的重要因素,本文总结出一种 AES 加密算法的差分错误分析方法,简化了其破解算法,并用软件模拟整个攻击过程,最后成功恢复了根密钥,实验结果证明仅需约 20 条左右的错误密文就能成功破解 AES 加密算法.另外,本文提出了一种新的 DES 差分错误分析方法,通过在末轮注入错误来恢复 DES 算法密钥,同样通过软件模拟实现了该差分错误分析.上述破解实例表明在智能卡芯片加密模块的设计过程中需要加入差分错误分析的防护手段来提升安全性能,可以引入错误侦测机制来抵御差分错误分析.本文总结了多种可以有效抵御差分错误分析的防护方法,并从面积、速度功耗等方面分析了实际芯片设计过程中这些防护方法的优缺点,对实际智能卡加密芯片设计的安全性能提高有着重要意义.

参考文献:

[1] Boneh D, DeMillo R, Lipton R. On the importance of checking cryptographic protocols for faults [C]// Lecture notes in computer science, Proceedings of EUROCRYPT'97. Berlin, Heidelberg: Springer-Verlag, 1997: 37-51.

[2] Biham E, Shamir A. Differential fault analysis of secret key cryptosystems [C]// Lecture notes in computer science, Proceedings of CRYPTO'97. Berlin, Heidelberg: Springer-Verlag, 1997: 512-525.

[3] Dusart P, Letourneux G, Vivolo O. Differential fault analysis on AES [C]// Lecture notes in computer science, ACNS. Berlin, Heidelberg: Springer-Verlag, 2003: 293-306.

[4] Giraud C. DFA on AES [C]// Lecture notes in computer science, AES. Berlin, Heidelberg: Springer-Verlag, 2003: 571.

[5] Skorobogatov S, Anderson R. Optical fault induction attacks [C]// Cryptographic Hardware and Embedded Systems, CHES 2002, Lecture notes in computer science. Berlin, Heidelberg: Springer-Verlag, 2003: 31-48.

Differential Fault Analysis on AES and DES

SUN Wei-dong, YU Jun, SHEN Lei

(*School of Microelectronics, State Key Laboratory of ASIC and System, Fudan University, Shanghai* 200433, *China*)

Abstract: The ability of symmetric encryption algorithm AES and DES against the differential fault analysis is examined. Two kinds of differential fault analysis on AES are described, and the initial key is recovered by software emulation. The result shows that the attack on AES can be realized by 20 times of fault-inductions. Then a new differential fault analysis on DES is described. The principle of this attack is analogous to differential fault analysis on AES. Result shows that attack on DES needs more fault-inductions. Therefore symmetric encryption algorithm AES and DES without any protection are vulnerable to differential fault analysis. It is presented that such attacks can be resisted by fault-detection.

Keywords: AES; DES; differential fault analysis; smart card