1 密码电路故障攻击理论与技术

1.1 故障攻击的理论基础：包括攻击原理、基本假设、模型、方法和评价标准。此外，还将阐述一些常用的DFA和其他一些safe error的故障分析方法

1.1.1 故障分析的攻击原理和基本攻击方法，包括DFA和IFA，FSA等

电子设备在运行时会受到各种各样自然的或是人为的干扰，在某些极端条件下设备停止工作或是输出不正确的结果。传统密码分析方法利用密码算法结构和一定明密文对来研究密码安全性，不同与此，故障攻击利用了设备在故障时的不正确结果中隐含的信息来辅助分析。故障攻击主要分为故障注入和故障分析两部分。故障注入主要研究密码设备在受到电压时钟毛刺，激光等干扰时的发生故障的模式；故障分析利用特定的故障输出作为辅助信息来完成密码分析，推导秘密信息。

最早的故障是偶然发生的，放射性元素衰变产生的带电粒子使得芯片出现了故障。之后为了提高芯片的稳定性和可靠性特别当芯片被用于外太空等极端环境下时，研究人员模拟和实验了大量不同的故障环境和对芯片的影响，这阶段研究的主要目的是研制高可靠性设备。

首个利用故障输出来推导密码算法的工作是由*Bellcore*等人在1997年提出的。该方法成功利用一对正确和错误密文对得出使用中国剩余定理实现的RSA算法的私钥。紧随其后故障分析的思想被Biham和Shamir等人用于分组密码DES的分析并提出了对几乎所有分组密码都有效的差分故障攻击。之后研究人员提出了大量的针对各种不同密码算法的故障分析和防护方案，使得故障分析成为一个极为活跃的研究领域。由于故障攻击可以在实际时间内成功攻击密码设备这也导致了对各种专用的故障注入设备的实验和开发。

在现行的软硬件架构和实现中故障注入可以影响程序的执行指令同时也可以影响程序的运算的中间值，对密码算法的故障分析主要集中于影响密码运算中间值的故障类型的分析。影响程序执行指令故障注入可以被用于跳过认证，修改程序流程等多种目的，但是在本章节中我们不考虑这种情况。

1.1.2 故障模型分类和介绍

故障模型是对设备在故障注入时产生的影响的抽象描述。故障模型要具有通用性，这样不同的密码算法不同的实现方式都可以通过故障模型加以描述同时这也提供一种便利的叙述背景和比较基准；故障模型也要具有可实现性，故障模型必须是实际的故障注入实验中可以重现的并且这种实现可以基于不同的软硬件设备和不同故障注入手段。

现在的分组密码大部分是迭代式的结构，算法包括多轮大致相同的轮函

数，轮函数由几个操作构成，故障注入一般影响某个特定轮的特定操作的中间值或是某几个特定轮中的某个中间值，该中间值的长度一般为分组密码明文长度。在公钥等密码算法中，受影响的中间值一般可以限定为在执行某个操作或是某一组操作时。

一些为大部分研究者所使用和认可的故障模型有单比特故障模型，单字节故障模型。单字节模型，密码算法的中间值的某一比特位的值发生改变，受影响的比特的位置在目标中间值中是随机且均匀分布的，该比特位在受到故障注入影响后随机的变为0或1。单字节故障模型，目标中间值的某一个字节发生改变，受影响的字节的位置在中间值中是随机且均匀分布的，该字节的故障值是随机并且均匀分布的，即故障值为0 255中的任何一个的概率为1/256。

在不同的分析场景下还有一些不同的故障模型，如在safe error的分析中，一般会假设受影响的比特、字节等的故障值为某一特定值。在轻量级分组密码的分析中，由于它们一般采用比较小（如3X3，4X4）的S盒，研究人员也会采用随机单S盒的故障模型。

随着故障攻击研究的深入，研究人员也在探讨各种新的故障模型，如故障位置或是故障值非均匀分布的故障，多字节的故障模型和更为通用的有偏差非特定型故障模型。

1.1.3 故障攻击的评价指标，包括所用故障模型，故障数，攻击轮数，攻击的时间复杂度等。

1.2 先进分析方法：包括一些当前研究热点，如故障和功耗攻击结合的分析方法

1.2.1 故障和传统密码学分析方法的结合

1.2.2 故障分析和功耗等旁路分析方法的结合

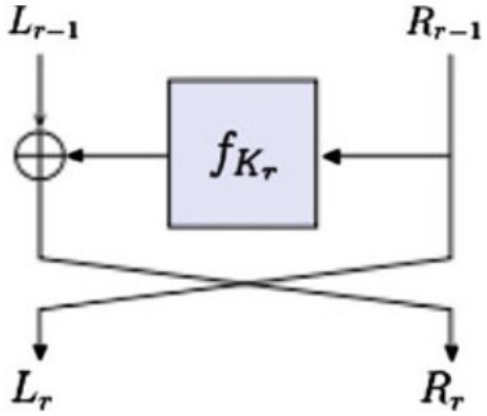
1.2.3 故障攻击在未知但可区分的故障模型下的应用

1.3 各类密码结构/算法的故障分析方法：针对常用的对称密码和非对称密码，给出其结构弱点和故障攻击分析方式

1.4 DES故障攻击

1996年，Biham和Shamir第一次提出了对DES的差分故障攻击（DFA），之后又有一些文章对此进行了扩展和改进，本节首先简要介绍DES算法，然后引入针对DES的后两轮原始故障攻击、中间轮故障攻击及基于内部碰撞的前几轮故障攻击。

Figure 1: Des feistel结构



1.4.1 DES算法

DES为64位的16轮Feistel结构(见图1)，每一轮的变换为$F_{K_r(L,R)=(R,L\oplus f_{K_r}(R))}$,其中L，R分别表示数据的左右部分，f是以48位密钥K_r为参数的32位输入到32位输出的映射函数（见图2）

1.4.2 对DES的第16轮攻击

假设在第16轮开始的时候，R_{15}的一些比特翻转，那么由图1可得R_{16}和R^*_{16}的差分满足等式：

$$\Delta R_{16} = f_{K_{16}} \oplus f_{K_{16}}(L^*_{16}) \tag{1}$$

由图2可知，f函数中的S盒独立进行计算，所以上式可以转换成8个等式,其中$(1 \leq i \leq 8)$：

$$P_i^{-1}(\Delta R_{16}) = S_i(E_i(L_{16}) \oplus K_{16,i}) \oplus S_i(E_i(L^*_{16}) \oplus K_{16,i}) \tag{2}$$

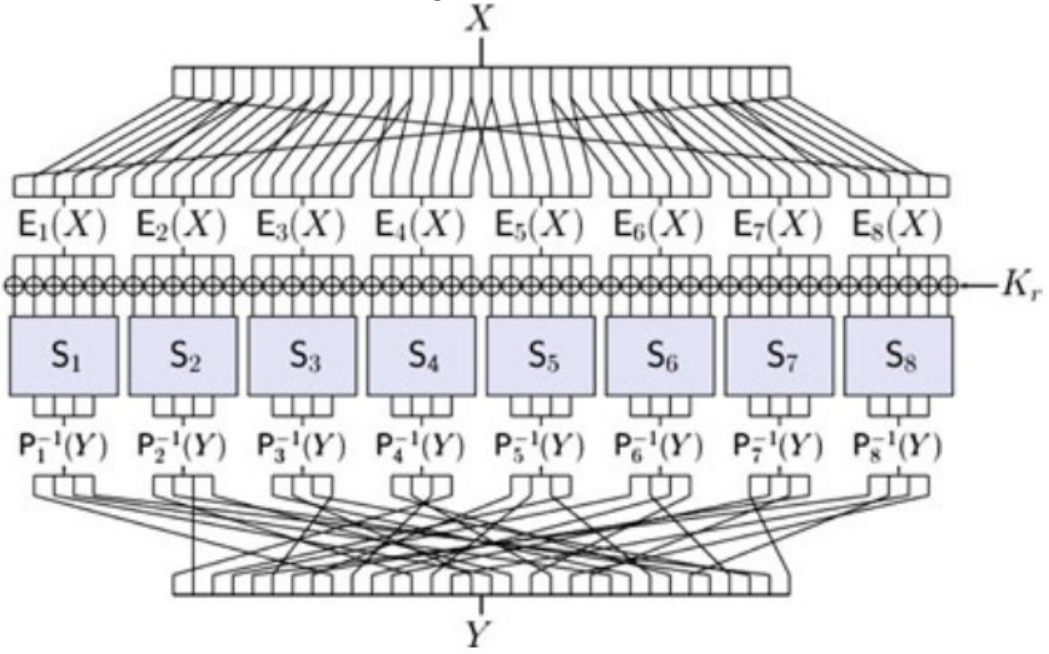
因此攻击者可以通过验证$K_{16,i}$的所有可能值$0,1^6$,排除不满足等式2的值，最终得到密钥。值得注意的是本攻击只适用于只影响R_{15}的故障故障类型，并且R_{15}被故障翻转的比特数越多，有效的S盒也越多（即受影响的S盒），所需要的密文对(C, C^*)则越少。

1.4.3 对DES的第15轮攻击

假设在第15轮开始的时候，在R_{14}上注入单比特故障，令$R^*_{14} = R_{14} \oplus \epsilon$，由图3可得：

$$\Delta R_{16} = f_{K_{16}}(L_{16}) \oplus f_{K_{16}}(L^*_{16}) \oplus \epsilon \tag{3}$$

Figure 2: f函数



在上式中K_{16}不是唯一的未知参数，因此接下来需要确定或者缩小ϵ的取值范围。利用式4可以推出第15轮的有效S盒，进而可以根据S盒的有效情况缩小ϵ的取值范围，如果两个S盒有效，则ϵ的值有2种可能，因为由图2可知每对S盒最多共享两个输入比特；同理如果只有一个S盒有效，ϵ的值也有两种可能，因为有效S盒的6个输入比特中与相邻S盒无关的只有2个比特。

$$\Delta L_{16} = f_{K_{15}}(R_{14}) \oplus f_{K_{15}}(R_{14}) \oplus \epsilon \tag{4}$$

一旦ϵ的范围确定，就可以用等式3推出8等式5(其中$1 \leq i \leq 8$），进而筛选出密钥K_{16}。

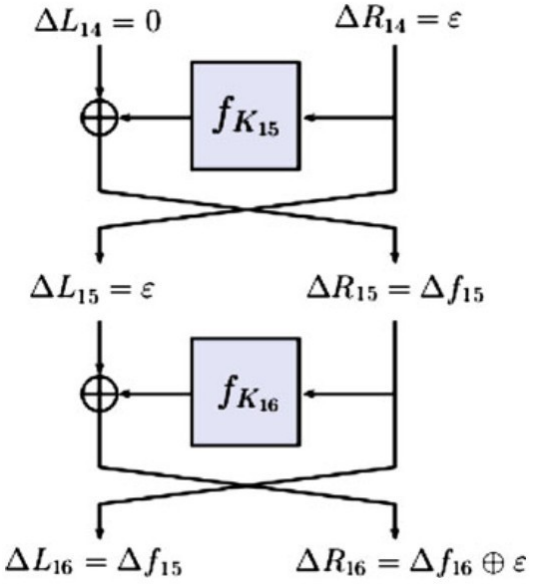
$$P_i^{-1}(\Delta R_{16} \oplus \epsilon) = S_i(E_i(L_{16}) \oplus K_{16,i}) \oplus S_i(E_i(L_{16}^*) \oplus K_{16,i}) \tag{5}$$

与上一节的攻击相比，本攻击为单比特故障模型（或者只有很少的几个比特故障），因为需要缩小ϵ的取值范围。又因为在第16轮错误被扩散到多个S盒，所以K_{16}的筛选效率并不比第16轮的攻击低多少。

1.4.4 对DES的中间轮攻击

针对中间轮DFA的基本原理是式6。在前面的攻击中,攻击者利用ΔL_{15}的值已知或者候选值很少的情况推出密钥。事实上，实施攻击并不一定要恢复

Figure 3: 15轮的错误扩散



出ΔL_{15}的值,只要ΔL_{15}的统计分布明显偏置，就能构造错误密钥区分器,进而筛选出密钥。

$$\Delta R_{16} = f_{K_{16}}(L_{16}) \oplus f_{K_{16}}(L_{16}^*) \oplus \Delta L_{15} \tag{6}$$

ΔL_{15}的统计分布跟故障翻转的平均比特数和故障位置到L_{15}的路径经过的f函数次数有关。例如，如果在L_{13}导入故障ϵ，那么$\Delta L_{15} = \Delta L_{13} = \epsilon$。如果$\epsilon$的汉明重量很小，那么$\Delta L_{15}$的分布将严重偏离均匀分布。更一般的，由图4可知，$L_{12}$到$L_{15}$仅经过f函数1次，$L_{11}$到$L_{15}$经过2次。由于DES的f函数的扩散不是很快，在$L_{12}$和$L_{11}$注入的故障将导致$\Delta L_{15}$的分布明显偏离。

定义$g_i((C, C^*), k) = S_i(E_i(L_{16}) \oplus k) \oplus S_i(E_i(L_{16}^*) \oplus k) \oplus P_i^{-1}(\Delta L_{15})$为预测$P_i^{-1}(\Delta L_{15})$的函数，$p_i(\delta)$为$P_i^{-1}(\Delta L_{15}) = \delta$的概率，则由分组密码分析中著名的错误密钥假设可得

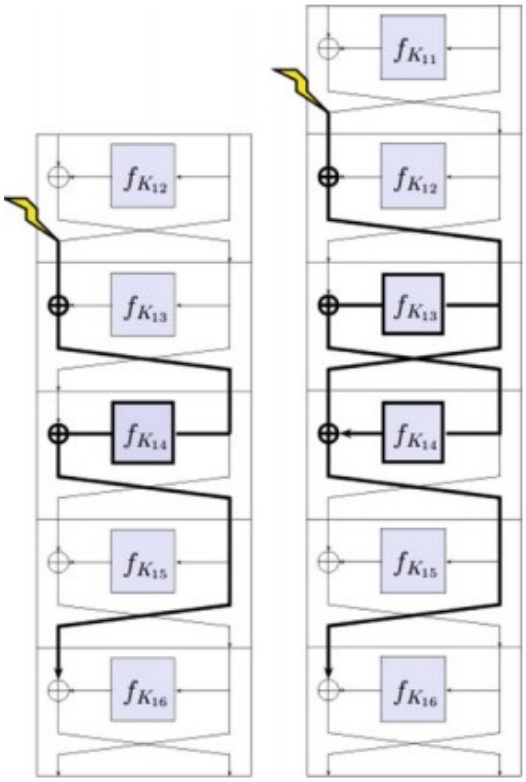
$$Pr[g_i((C, C^*), k) = \delta] = \begin{cases} p_i(\delta) & if \quad k = K_{16,i} \\ \frac{1}{16} & otherwise \end{cases}$$

因为$p_i(\delta)$明显偏离均匀分布，所以可以用数理统计中经典的极大似然估计或者用欧氏距离估计（SEI）来区分出正确密钥$K_{16,i}$，对应的区分器分别为

$$d(k) = \sum_{n=1}^{N} log(p_i(g_i(C_n, C_n^*, k)))$$

和

Figure 4: L_{11}和L_{12}到L_{15}的错误扩散路径



注入轮	区分器	比特故障		字节故障	
		选择位置	随机位置	选择位置	随机位置
12	极大似然	7	11	9	17
	欧氏距离	14	12	17	21
11	极大似然	11	44	210	460
	欧氏距离	30	71	500	820
10	极大似然	290	1500	13400	18500
	欧氏距离	940	2700	26400	23400
9	极大似然	$3.4 * 10^5$	2.2×10^7	$> 10^8$	$> 10^8$
	欧氏距离	1.4×10^6	$> 10^8$	$> 10^8$	$> 10^8$

Table 1: 针对DES中间轮的攻击结果

$$d(k) = \sum_{\delta \in \{0,4\}^4} (\frac{\{\#n; g_i(C_n, C_n^*, k) = \delta\}}{N} - \frac{1}{16})^2$$

表1为模拟实验的攻击结果，由表可知本节的攻击对于第11轮和第12轮注入的故障所需的故障数很少，对于第10轮和第9轮注入的故障所需的故障数相对较多，所以为了抵抗本节的DFA，相应的DES算法至少保护最后6轮。

1.4.5 AES故障攻击

自从2000年被选为Advanced Encryption Standard(AES)，Rijndael得到了越来越多的应用，出现了大量关于AES故障攻击的研究。本节选取几种最为典型的针对AES的故障攻击加以描述。

由AES的密钥生成算法已知最后一轮的轮密钥，攻击者可以恢复128比特AES的主密钥。对于192,256比特AES需要知道倒数两轮的轮密钥才可以完整恢复主密钥。以下的几种故障攻击方法都专注于讨论如何恢复最后一轮子密钥。对于192,256比特AES，可以先使用同样方法恢复最后一轮子密钥，然后将故障注入轮数提前一轮，所得正确错误密文用最后一轮子密钥解密一轮，将得出的中间状态值视为当前情况的正确错误密文对，使用和恢复最后一轮轮密钥类似方法可以恢复出倒数第二轮子密钥，形成完整的攻击。

下面描述的第一种攻击采用随机单字节故障模型，故障注入在倒数第二轮且在列置换操作之前的任一字节。此时攻击的目标状态为倒数第二轮的列置换之前，经过一轮的故障传播在该目标状态的每一列有且仅有一个字节的正确错误状态的差分值非零。攻击者猜测最后一轮子密钥的四字节，对正确和错误密文分别进行部分解密，根据目标状态相应列是否符合前句所述的模式即可排除错误的密钥的猜测。每组这样的猜测可以用于确定相

关四字节的子密钥，因此通过四组这样的猜测就可以最后一轮子密钥的恢复。已有的模拟实验显示通过2组正确错误密文对攻击者能以92

第二种攻击仍然采用随机单字节故障模型，但是故障注入的轮数相对第一种攻击提前一轮。此时在倒数第二轮的列置换之前的状态的每个字节的差分值均不为0，正确错误的中间状态可用正确错误密文对和最后一轮子密钥标示。同第一种方法类似，攻击者猜测最后一轮子密钥的四字节并做部分解密检测在目标状态的每个字节的差分值是否都非0，若满足，则保留为候选密钥。对于子密钥$K_0^r, K_7^r, K_{10}^r, K_{13}^r$，满足的等式如 。模拟实验显示约1000对密文对可以将密钥空间降低为2^{40}。

$$MC^{-1}|_0(SB^{-1}(C_0 \oplus K_0^r)) \oplus MC^{-1}|_0(SB^{-1}(C_0' \oplus K_0^r) \neq 0 MC^{-1}|_1(SB^{-1}(C_{13} \oplus K_{13}^r)) \oplus MC^{-1}|_1(SB^{-1} \tag{7}$$

上述攻击的故障模型比较简洁，第三种攻击采用更为复杂的故障模型。可以观察到，当故障发生在倒数第三轮的输入状态的某一对角线时，本轮结束时的状态的差分值全部集中在一列上。在倒数第二轮的列置换之前每一列仍然只有一个字节的差分值不为0.若故障注入在第一对角线，经过推导可以的得出如下等式。

$$SB^{-1}(C_0 \oplus K_0^r) \oplus SB^{-1}(C_0' \oplus K_0^r) = 2(SB^{-1}(C_{13} \oplus K_{13}^r) \oplus SB^{-1}(C_{13}' \oplus K_{13}^r)) SB^{-1}(C_{10} \oplus K_{10}^r) \oplus SB^{-1}(\tag{8}$$

攻击者首先猜测K_0^r和K_{13}^r，如果猜测值满足上述第一个等式则保留为候选密钥字节，否则为错误密钥。结合K_{13}^r的候选值和K_{10}^r的所有可能值并测试其是否满足上述第二个等式可以进一步减少密钥空间，随后将同样的方法运用到第三个等式，通过以上三组操作$(K_0^r, K_7^r, K_{10}^r, K_{13}^r)$的可能值的数量平均为$2^8 = 256$个。

对于目标状态的其他三列也有类似的结果，这样最后可以将最后一轮轮密钥的密钥空间降低到$(2^8)^4 = 2^{32}$。上述讨论中故障所处的对角线已知，如果故障发生的对角线未知，需要猜测所有可能四种情况，密钥空间是已知对角线情况的4倍。

上述攻击可以被扩展到在倒数第三轮的输入状态上至多三个对角线上的值受影响的故障模型。此时可以列出类似上述所示等式组，具体的等式会稍微比较复杂，如果没有发生故障的对角线已知，等式组成的密钥区分器可以将猜测的4字节的密钥空间降低到2^{24}。在这样的故障模型下，使用四对正确错误密文对，最后一轮轮密钥可以以很大的概率确定。

1.4.6 RSA故障攻击

1.4.7 ECC故障攻击

1.4.8 其他算法包括轻量级算法的故障分析

1.4.9 流密码和哈希函数的故障攻击简介

1.5 故障攻击实验环境：包括已有的故障攻击实施工具和攻击过程，如电压和时钟Glitch，激光和电磁辐射等

故障注入是一次实际的完整的故障攻击的第一步，故障注入的结果直接决定了故障模型、相应的攻击方法和之后故障分析的复杂度如何。故障分析采用的故障模型必须能够和某种故障注入方式相吻合。故障注入的方式多种多样，原理、攻击平台、攻击效果各不相同，完成攻击所需要的造价也有很大差异。按照其作用效果可分为全局型的故障注入和局部型的故障注入，瞬时的故障和永久的故障。按照攻击前对目标芯片的处理程度可分为侵入式攻击和非侵入式攻击。本章将逐一列举几种常见的且比较有效的攻击方式和在这方面的一些攻击结果。由于故障注入的多样性，以下的小节中将以特定的攻击实验来组织描述，试图达到窥一斑而知全豹的效果。

1.5.1 电压和时钟的故障实验环境和注入结果

电压和时钟攻击在芯片的电压或时钟输入管脚提供不正常的输入来影响电路运行使之出错。不正常的输入可以使过高或是过低的电压，电压或是时钟上的毛刺等。电压和时钟的攻击是一种全局性的注入方式，故障影响整个芯片电路，故障注入可以再时间上有比较精确的控制但是无法对某一些运算单元单独注入故障。同时完成电压和时钟注入所需的成本较低，因此的到了广泛的应用和研究。 在CMOS电路中，每个门电路上都存在着电容因此都会有传播延时，在门电路的输入端的变化需要一段时间才会产生相应输出。传播延时的长短是由好多因素共同决定的。其中最主要的决定因素是门电路所做的运算，例如，在与门中，一个输入端为0的话，无论另一个输入端如何变化，结果恒为0；而或门就没有这样的特性。输入信号的形状也会影响传播延时，信号经过长的连线后会有比较平滑的边缘，而有良好缓存只经过短的线路的信号有比较陡峭的边缘，连线的长短也间接地影响了传播延时。其他会影响传播延时的因素有连线之间的耦合，一些门电路的不确定的响应时间和电路中产生的毛刺等。因而必须在等电路输出稳定之后才可以获得需要的结果。 在时序电路中，时钟信号并行的与所有组合电路发生关联，这意味着当时钟信号的上升边缘来临时所有门电路都已完成结果的稳定输出。这样最长的传播路径就决定了电路的最大时钟频率。这个最大的传播延时就叫做启动时间。如果出于一些原因使时钟间隔变短或启动时间变长，时钟间隔小于启动时间，错误就有可能发生。 实验结果显示传播延时会随着供电电压的变低而逐步变长，在实验中使用的电

路的极限供电电压0.4V下，输出结果经过很长时间都没有收敛。如果该传播延时没有影响在最长传播路径，则没有故障；否则，时钟上升沿在输出信号变化时采样，就有一定概率出错或是逻辑门电路输出值为错误结果就会导致故障。实验结果显示温度也会对传播延时产生影响但是影响较小。

本节中讨论的实验是针对ASIC实现的AES。故障注入设备包括与智能卡读卡器，稳压电源和一个任意波形的信号发生器。这样就可以为智能卡提供各种不同的电压和时钟信号。实验中使用的智能卡是130nm的ASIC，额定供电电压是1.2V。故障注入过程选取的智能卡能正常工作同时又有一定出错概率的电压范围为775-825mV，电压变化的步长为0.5mV，共100组实验。每组实验包括20，000次故障注入，在每组实验中密钥被设置为固定值，明文可以发生变化。实验显示通过降低供电电压可以产生一般的故障分析所需要的单个字节的故障模型。在实验中可观察到随着电压的降低发生故障的可能性逐步增大，但是在所有故障中单字节故障发生的可能性有所不同，单字节故障随电压变化基本符合钟形分布，在800mV的时候概率最大，约占所有故障的30

1.5.2 电磁辐射的故障实验环境和注入结果

1.5.3 激光的故障实验环境和注入结果

光照故障注入是另一类被广泛采用的注入技术，光照故障注入通常需要更昂贵的设备和各专业的操作，但是光照注入技术往往可以达到比较好的空间精度，获得一些全局故障注入技术无法获取的信息。

最简单的光照注入技术采用很强的同时精确聚焦的光速来对芯片进行故障注入。对晶体管进行强光照射可在其上的电介质上形成暂时的导电通路，类似于晶体管正确偏置时其上产生的通路。这样就实现以一种精确和可控的方式改变了晶体管的状态。为了达到比较精确的聚焦光束，相机闪光灯产生的光束通过显微镜目镜聚焦到载物台上，在载物台放置了需要注入的芯片。为了防止过度照射对芯片电路的永久性修改攻击者必须仔细的控制显微镜镜头的放大倍数。同时攻击者还需要设计异步电路来控制闪光灯与芯片同步工作。简单光照注入的最大的不足在于自然光无极性，相干性较差，自然光的波长是当前蚀刻工艺门电路的电介质部分的十倍左右，同时显微镜也无法达到精确的聚焦，因而就无法实现对单个逻辑门电路的故障注入。由于闪光灯无法在短时间内产生一系列的闪光，这种技术也无法在设备运行时进行多次的故障注入。

对上述注入技术最直接的改进就是用激光代替闪光灯。故障注入的原理与闪光灯的基本相同，但是激光故障可以更有效在单次照射中注入故障同时也拥有更高的空间精度。同时激光还可以照射芯片背面来进行故障注入。这是通过使用红外激光来实现的，纯的硅晶体对1 Um到5 um的光线有50

1.5.4 多点故障注入的实验环境和实验结果

1.5.5 抗故障攻击的防御方法：已有的抗故障攻击防御技术

1.5.6 冗余计算的防护方案

1.5.7 掩码防护方案

1.5.8 抗故障攻击的电路单元