文章编号：2096-1618(2019)05-0457-05

SM4 算法前四轮约减轮故障注入分析

王 恺[1]，吴 震[1]，杜之波[1]，王 敏[1]，王 燚[1]，习 伟[2]
(1. 成都信息工程大学网络空间安全学院，四川 成都 610225；2. 南方电网科学研究院有限公司，广东 广州 510080)

摘要：针对 SM4 密码算法，提出一种新型的约减轮故障攻击方法，故障注入于加密算法的前 4 轮中，使加密算法的后续迭代轮数减少，对故障数据进行筛选和分析，理论上由 4 个错误密文就可以恢复 128 bit 的初始密钥，从而完成攻击。方法对比现有针对 SM4 的差分故障攻击方法有很大的简化，提高了攻击效率。利用攻击方法对无防护 SM4 算法的智能卡进行了约减轮故障注入攻击，实验结果表明攻击方法行之有效，也可以扩展到其他分组密码。

关 键 词：信息安全；硬件安全；SM4 密码算法；故障注入；分组密码；约减轮故障
中图分类号：TP915.08 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2019.05.002

0 引言

SM4 密码算法是中国自主设计的分组密码算法，数据分组长度为 128 bit，密钥长度为 128 位。SM4 算法于 2016 年 8 月正式成为国家标准[1-2]，并广泛应用于无线局域网领域，因此 SM4 密码算法的安全性分析是当前研究的热点。

正常情况下，密码算法的运行都是软件程序或硬件设备按照加解密流程正确地执行加密或解密运算，最终返回加密后的密文或解密后的明文。但若存在人为干扰，密码运算的设备或程序就可能产生一些故障，生成对应的故障信息。这些信息和正在运行的加解密运算有很强的相关性，利用这些故障信息进行分析并对密钥进行恢复的方法就是密码故障注入分析方法[3]。Boneh 等[4]用故障注入的方法成功攻破了 RSA 签名算法。

提出一种针对 SM4 密码算法的约减轮故障攻击方法，对 SM4 密码算法加密过程的 1 ~ 4 轮进行故障注入，中断后续的 28 轮迭代轮函数，直接进入反序过程，利用故障注入产生的错误密文进行分析，获取 SM4 密码算法前 4 轮的轮输出，通过这 4 轮输出和已知的明文恢复前 4 轮的轮子密钥，可最终恢复出初始密钥。

1 SM4 密码算法

SM4 密码算法是分组密码算法[4-6]。分组密码算法是指将明文数据按固定位数进行分组，然后每个分组使用相同的密钥进行加密运算，将各个明文分组变换成相对应的密文分组的密码算法。分组密码算法使用相同的密钥进行加解密，同时为了防范攻击者对密码系统进行统计学分析，分组密码系统的设计需要遵循扩散和混淆两个最本质的操作。现在分组密码算法通过代替和换位的多轮迭代来实现扩散和混淆。

1.1 SM4 算法简述

为了实现更高的安全性，SM4 密码算法在简单的线性变化的基础上加入了 S 盒部分对数据进行非线性变化。SM4 密码算法的明文分组长度为128 bit，加解密密钥长度也为128 bit，加解密过程均使用 32 轮非线性迭代结构，产生每轮子密钥的密钥扩展算法同样采用了 32 轮非线性迭代结构，其加解密算法的结构相同，进行解密运算时每轮的子密钥使用反向的加密运算的轮密钥，解密轮子密钥是加密轮子密钥的逆序。

SM4 密码算法加解密过程使用含有非线性变化的迭代结构，每进行一次迭代运算称为一轮变换，并且轮子密钥的生成同样使用了非线性的迭代结构，加密以及密钥扩展运算都采用了 32 轮迭代结构，从而保证密码系统的抗统计分析能力。

1.2 SM4 密码算法加密流程

SM4 密码算法的轮函数采用的基本运算为：模 2 加法和循环移位，主要包括线性变换和非线性变换部分，其中非线性变换部分由 S 盒构成。SM4 密码算法的加密运算全流程如图 1 所示。

明文分组 X，大小为128 bit，进行加密运算，首先使用轮函数 F 进行 32 轮迭代加密，然后将最后 4 轮的

收稿日期：2019-03-27
基金项目：国家重点研发计划资助项目(2018YFB0904900，2018YFB0904901)；国家科技重大专项基金资助项目(2014ZX01032401)；“十三五”国家密码发展基金资助项目(MMJJ20180244)；四川省教育厅重点科研基金资助项目(17ZB0082)；四川省重点研发资助项目(2019YFG0096)

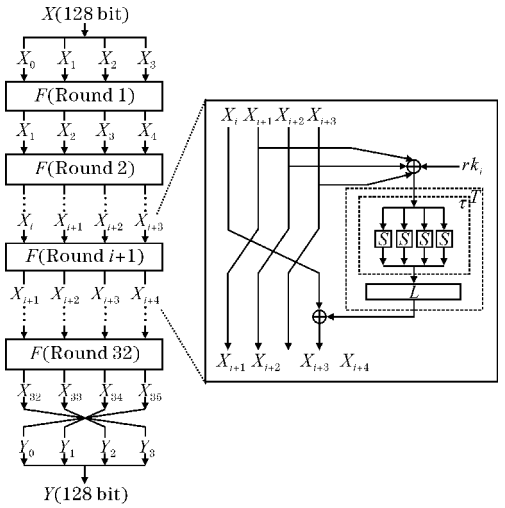

图 1 SM4 加密运算流程

输出进行反序变换成为密文 Y,Y 大小也是128 bit。加密运算时明文分组 X 表示为$(X_0,X_1,X_2,X_3)\in(Z_2^{32})^4$,($(Z_2^{32})$表示 32 bit 的向量集),轮子密钥表示为$rk_i\in Z_2^{32}$ ($i=0,1,2,\cdots,31$),其中 i 为轮数,则加密轮函数表达式为

$$F(X_0,X_1,X_2,X_3,rk)=X_0\oplus T(X_1\oplus X_2\oplus X_3\oplus rk) \tag{1}$$

每轮迭代的输出为 X_{i+4},则具体输出表达式为

$$X_{i+4}=F(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i)=X_i\oplus T(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i) \tag{2}$$

其中 T 是由非线性变换 τ 和线性变换 L 构成的合成置换。轮加密函数 F 的流程如图 2 所示。

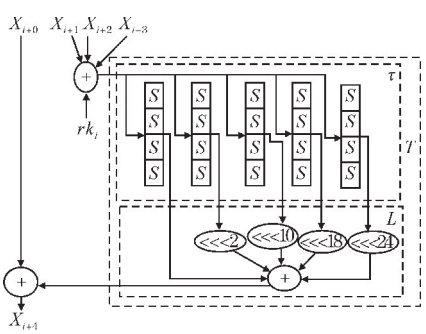

图 2 轮函数流程图

非线性变换 τ 有 4 个并列的 S 盒(Sbox)构成,Sbox 是根据 Sbox 置换表进行非线性变换的。具体的变换规则为:用输入的前4 bit的十六进制数作为行号,后4 bit的十六进制数作为列号进行查表,行列号共同确定的列表数据作为输出。SM4 密码算法的 Sbox 输入和输出都为8 bit,非线性变化的输出为 $B\in Z_2^{32}$,令其作为线性变化 L 的输入,则线性变换输出为 $C\in Z_2^{32}$,具体表达式为

$$C=B\oplus(B<<<2)\oplus(B<<<10)\oplus(B<<<18)\oplus(B<<<24) \tag{3}$$

1.3 密钥扩展算法

轮函数 F 中使用的轮秘钥 rk_i(长度为32 bit)由原128 bit的密钥通过密钥扩展算法计算生成,密钥扩展算法同样使用轮函数进行了 32 轮迭代。与加密轮函数对比,密钥扩展轮函数在线性变化部分存在区别。若初始密钥为 $MK=(MK_0,MK_1,MK_2,MK_3)$,$MK_i\in Z_2^{32}$,则密钥扩展算法表达式为

$$(K_0,K_1,K_2,K_3)=(FK_0\oplus MK_0,FK_1\oplus MK_1,FK_2\oplus MK_2,FK_3\oplus MK_3) \tag{4}$$

$$rk_i=K_{i+4}=K_i\oplus T'(K_{i+1}\oplus K_{i+2}\oplus K_{i+3}\oplus CK_i) \tag{5}$$

T'与 T 的区别在于线性变化部分,密钥扩展算法的线性变化 L'为

$$L'=B\oplus(B<<<13)\oplus(B<<<23) \tag{6}$$

其中 $FK_i(i=1,2,3)$为系统参数,采用 16 进制表示为:FK_0 = A3B1BAC6, FK_1 = 56AA3350, FK_2 = 677D9197, FK_3 = B27022DC。

CK_i 为固定参数,其取值方法为:设 CK_i 的第 j 字节为 $ck_{i,j}(i=0,1,\cdots,31;j=0,1,2,3)$,即 $CK_i=(ck_{i,0},ck_{i,1},ck_{i,2},ck_{i,3})\in(Z_2^8)^4$,则 $ck_{i,j}=((4i+j)\times7)\ mod\ 256$。可计算出 32 个固定参数 CK_i 用 16 进制表示为

00070e15, 1c232a31, 383f464d, 545b6269,
70777e85, 8c939aa1, a8afb6bd, c4cbd2d9,
e0e7eef5, fc030a11, 181f262d, 343b4249,
50575e65, 6c737a81, 888f969d, a4abb2b9,
c0c7ced5, dce3eaf1, f8ff060d, 141b2229,
30373e45, 4c535a61, 686f767d, 848b9299,
a0a7aeb5, bcc3cad1, d8dfe6ed, f4fb0209,
10171e25, 2c333a41, 484f565d, 646b7279

2 密码故障注入分析

运行密码算法的硬件设备正常状态下能够正确地执行加解密运算,但在有外部干扰的情况下,密码运算的模块可能会产生一些故障,如寄存器故障或运算异常。密码故障注入分析,就是利用这些人为产生的故障所造成的错误信息对密钥或者明文进行分析或者恢复。

现阶段密码设备对故障注入的防护措施存在欠缺,比较容易受到攻击。近年来,针对智能芯片的故障注入技术开始飞速发展。

2.1 故障注入攻击

故障注入攻击通常由故障注入和错误数据分析两个步骤构成。故障注入是指在适合的时间通过外部干扰诱导密码运算的某些中间状态产生故障,使密码设

备或软件产生错误信息。根据侵入物理设备的程度,将故障注入分为 3 大类,非侵入式故障注入、半侵入式故障注入和侵入式故障注入。

非侵入式故障注入:不需要直接接触芯片核心,通过外界干扰的方式,如加载异常电压、时钟、磁场等,来引发密码设备异常工作产生故障信息。其优点是操作相对容易,不需要额外的特殊设备。

侵入式故障注入和半侵入式故障注入:必须直接物理接触密码设备的核心芯片,因此要将密码设备先进行预处理,暴露密码芯片。此过程需要特殊的设备,如用于剖片处理的设备、化学腐蚀设备等。其优点是故障注入成功率高,注入时机更容易掌控。

错误数据分析是利用故障注入所产生的错误信息,并使用与其匹配的分析方法对密钥进行恢复,攻击能否成功一般既依赖于密码系统的设计和实现,也依赖于其算法规范。

2.2　攻击步骤

故障注入攻击一般分为 4 个步骤。

2.2.1　确定故障模型

故障模型指在故障注入之前需要明确的故障注入时机、故障注入位置、故障注入动作和注入效果的模型。

2.2.2　故障注入

根据故障模型,选定故障注入的手段,如电流、时钟、激光和射线等干扰方式进行错误注入。常用的故障注入方法分有 3 类:直接从外部加入非正常信息,如异常的时钟、电压和温度等;结合探测攻击,从内部引入错误;去除钝化层之后,使用激光或重粒子辐照芯片制定区域,引入错误。

2.2.3　错误样本筛选

进行故障注入后,根据具体的需求对所有故障注入产生的错误信息进行筛选,文中主要筛选依据为设备运行的时间,也可以是错误信息的统计学特征等,再结合故障模型筛选出理想的故障模型。

2.2.4　密钥恢复

对筛选出的理想的错误样本进一步分析,主要分析密码运行正确输出和错误输出之间的关系,使用与其对应的错误分析方法,再结合密码算法的设计对密钥进行恢复。

2.3　现有的针对 SM4 密码算法的故障攻击

差分故障[7-11]分析是现阶段最成熟的应用于 SM4 密码算法故障注入攻击的方法,主要实现方式是:先记录故障注入后密码设备或软件产生的错误密文和相同明文情况下正常运行所产生的正确密文,再将错误密文和正确密文进行比对,通过两者间的相互关联性对密钥进行恢复。

文献[7]描述的攻击方法恢复 128 bit 的 SM4 密码算法的密钥理论上需要 32 个故障注入产生的错误密文,故障注入模型如下:注入时机为轮函数加密过程;故障注入点为文献中特定存储单元。此故障注入方法对故障注入时机和故障注入点的精确度要求很高,因此故障注入成功率很低。

文献[9]在文献[7]的基础上主要对故障注入时机和故障注入点两个方面进行了改进。文献[7]的故障注入点需要精确指定的特定存储单元,而文献[9]的故障注入点扩展为密钥扩展算法执行过程的随意一个存储单元。故障注入时机改变为轮子密钥的生成阶段,大大降低了错误注入的难度,提高了错误注入的成功率。理论上,此方法有 8 个错误密文即可成功完成攻击。

基于差分思想的故障分析对故障注入的位置要求高,需要的密文个数多,根据故障信息恢复密钥的过程比较复杂,方法的通用性差。下文提出的攻击方法可以解决这些问题,并且可以较方便地扩展到其他迭代密码算法。

3　基于约减轮的故障注入

分组密码算法每一轮的轮函数相对较弱,经过多次迭代最终形成安全性较强的密码函数。因为采用迭代结构,分组密码算法迭代的轮数越多其安全性就越强,扩散和混淆也就进行得越彻底。所以,通过故障注入的方式减少密码设备或软件实际运行的轮数是对分组密码算法最直接的攻击方式。

Anderson[12]提出了基于约减轮故障注入的思想的实际攻击方法,此方法攻击目标为芯片电源或者时钟以产生毛刺达到破坏循环变量或条件转移运算的目的。Choukri 等[13]对未加防护的 Slivercard 上运行的 AES 算法成功实现了攻击。

针对 SM4 密码算法加密运算的前 4 轮约减轮故障攻击的实施步骤如下:

(1)假设密钥 K,选定一组明文 $X=(X_0,X_1,X_2,X_3)$,进行正常加密运算并采集密码设备的功耗曲线 T,分析功耗曲线 T 从 32 轮加密运算中确定前 4 轮执行的时间段。

(2)进行故障注入,注入时机为 SM4 加密运算的前 4 轮进行的时间段,诱导加密算法提前跳出正常的轮函数迭代过程,跳转到反序变换,记录输出的错误密文 Y_i。

实际操作为依据 SM4 密码算法的加密功能的 C 语言实现如图 3 所示。

```
void SMS4_Decryption(INT32U X[], INT32U rk[], INT32U Y[])
{
    INT32U tempX[4] = {0};
    INT8U  i = 0;

    for (i=0; i<4; i++)
        tempX[i] = X[i];

    for (i=0; i<SMS4_ROUND; i++)
        tempX[i%4] ^= T1(tempX[(i+1)%4] ^ tempX[(i+2)%4] ^ tempX[(i+3)%4] ^ rk[(31-i)]);

    for (i=0; i<4; i++)
        Y[i] = tempX[3-i];
}
```

图 3 SM4 算法加密函数

故障注入位置为密码设备中参数 SMS4_ROUND 所在的寄存器,SMS4_ROUND 表示当前轮函数迭代的轮数,故障注入需要使 SMS4_ROUND 参数产生异常。即当 SMS4_ROUND =1,2,3,4 时,直接跳出当前 for 循环进行反序变换,并记录输出的错误密文 $Y_i(i=1,2,3,4)$。

(3)对故障样本进行筛选的依据是设备运行的时间,具体是通过采集故障注入后设备运行的功耗时间曲线,与正常加密的功耗时间曲线进行比对,根据加密的执行时间长短对故障样本进行筛选。加密运算的迭代轮数可以通过采集的功耗曲线判断,选择理想的故障样本进行错误密文分析。

(4)攻击加密流程第 2 轮,通过第(2)、(3)步可以得到理想的错误密文 Y_1。根据 SM4 密码算法加密流程可知,密文 Y_1 通过逆反序变换 R' 可以得到第一轮的轮输出 X_4,而明文 $X=(X_0,X_1,X_2,X_3)$ 已知。根据已知参数可恢复轮子密钥 rk_0。

(5)继续攻击第 3 轮、第 4 轮和第 5 轮。重复上述步骤,同理恢复出轮子密钥 rk_1,rk_2,rk_3。

(6)根据密钥扩展算法利用已恢复的 rk_0,rk_1,rk_2,rk_3 恢复密钥 K。

4 试验与分析

4.1 试验平台构成

试验中功耗采集使用的设备如表 1 所示。

表 1 实验功耗采集环境

设备名称	详细型号
密码设备	智能卡
驱动设备	Riscure Power Tracer
故障注入设备	Riscure VC Glitcher
服务器	DELL PowerEdge T330
侧信道分析	Riscure Inspector
示波器	WaveRunner 6 Zi Oscilloscopes

4.2 采集智能卡正常工作功耗曲线

设置 SM4 密码算法的明文 $X=(X_0,X_1,X_2,X_3)$ 为 0x01 0x23 0x45 0x67 0x89 0xab 0xcd 0xef 0xfe 0xdc 0xba 0x98 0x76 0x54 0x32 0x10,智能卡进行常规加密运算,生成密文 Y 为 0x68 0x1e0xdf0x34 0xd2 0x06 0x96 0x5e0x86 0xb3 0xe9 0x4f0x53 0x6e0x42 0x46,采集的功耗曲线如图 4 所示。

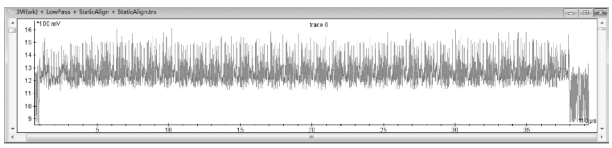

图 4 SM4 加密功耗曲线

4.3 试验过程

根据图 4 所示的功耗曲线的时间轴,可区分 SM4 密码算法加密过程前 4 轮轮函数运行的时间区间。在这个时间区间里,对存储轮函数运行次数(SMS4_ROUND)的寄存器进行故障注入。试验进行了 1500 次故障注入攻击,成功 1047 次,其余的 453 次故障注入失败。故障注入成功时采集到的功耗曲线如图 5 所示,此曲线对应的故障注入返回的错误密文是理想的故障样本。

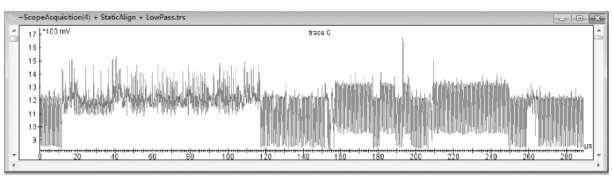

图 5 理想的注入成功的曲线

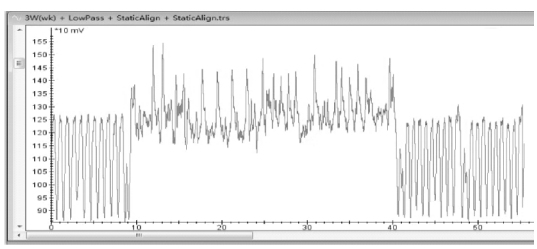

图 6 SM4 错误密文能量曲线

分析图 6 对应的错误注入试验返回的错误密文 Y_1 为 27 FA D3 45 76 54 32 10 FE DC BA 98 89 AB CD EF,与设定的明文后96 bit相同。可得知:此错误密文为 SM4 算法加密运算只进行了第 1 轮加密反序输出的数据,即 X_4,X_3,X_2,X_1,结合输入的明文可推出第 1 轮的轮子密钥 rk_0 = F12186F9。

再次分析图 6 对应的错误注入试验返回的错误密文 Y_2 为 A1 8B 4C B2 27 FA D3 45 76 54 32 10 FE DC

BA 98,与 Y_1 后96 bit相同。由此确定该数据是只进行了前两轮轮加密,将 X_2, X_3, X_4, X_5 进行反序然后输出的。第 2 轮加密轮函数的输入为 X_1, X_2, X_3, X_4,则可推出第 2 轮的轮子密钥 rk_1 =41662B61。

同理,可推出 rk_2 = 5A6AB19A,rk_3 = 7BA92077。

4.4　结果验证

根据 rk_0、rk_1、rk_2 和 rk_3 可恢复出初始密钥 K,用此密钥加密明文 X 得到的密文 Y 与设备正常工作所得的密文一致,确认攻击成功。

5　结束语

提出了一种针对 SM4 密码算法加密过程前 4 轮的基于约减轮的故障注入攻击方法。实验表明,基于约减轮思想的故障注入攻击,可以很大限度地降低基于差分思想的故障注入攻击恢复轮密钥的复杂程度,理论上只需要 4 个错误密文即可恢复出初始密钥。未来的研究应着手于提高故障注入的效率,以及筛选样本的自动化程度,以及本思想在其他分组密码中的应用。

参考文献:

[1] 全国信息安全标准化技术委员会. GB/T 32907-2016 信息安全技术 SM4 分组密码算法[S]. 北京:中国质检出版社,2016.

[2] 国家商用密码管理办公室. 无线局域网产品使用的 SMS4 密码算法[EB/OL]. http://www.oscca.gov.cn/upfile/200621016423197990.pdf,2006.

[3] Dan B, Richard A, Demillo R, et al. On the importance of checking cryptographic protocols for faults [C]. 1997:1175-1213.

[4] Kocher P C. Timing attacks on implementations ofDiffie-Hellman, RSA, DSS, and other systems [C]. In: Advances in Cryptology-CRYPTO 1996. Springer Berlin Heidelberg,1996:104-113.

[5] Johannes, Jean P S. Fault based cryptanalysis of the advanced encryption standard (AES) [M]. Springer Berlin Heidelberg,2002:162-181.

[6] Piret G, Quisquater J J. A differential fault attack technique againstspn structures, with application to the AES and khazad[C]. Cryptographic Hardware and Embedded Systems-CHES 2003, International workshop,2003:77-88.

[7] 张蕾,吴文玲. SMS4 密码算法的差分故障攻击[J]. 计算机学报,2006,29(9):1596-1602.

[8] 李玮,谷大武. 基于密钥编排故障的 SMS4 算法的差分故障分析[J]. 通信学报,2008,29(10):135-142.

[9] Li R L, Sun B, Li C, et al. Differential fault analysis on SMS4 using a single fault[J]. Information Processing Letters,2011,111(4):156-163.

[10] 荣雪芳,吴震,王敏,等. 基于随机故障注入的 SM4 差分故障攻击方法[J]. 计算机工程,2016,42(7):129-133.

[11] Li W, Gu D. An improved method of differential fault analysis on the SMS4 cryptosystem[C]. International Symposium on Data Privacy and E-commerce,2007:175-180.

[12] Ross A, Markus K. Low cost attacks on tamper resistant devices[C]. Springer,1997:125-136.

[13] Hamid C, Michael T. Round reduction using faults [J]. FDTC,2005,5:13-24.

Analysis of the First Four Rounds of Reduction Wheel Fault Injection in SM4 Algorithm

WANG Kai[1], WU Zheng[1], DU Zhibo[1], WANG Min[1], WANG Yi[1], XI Wei[2]

(1. College of Cyberspace Security, Chengdu University of Information Technology, Chengdu 610225, China; 2. China Southern Power Grid Science Research Institute Co., Ltd., Guangzhou 510080, China)

Abstract: A new method of round-trip fault attack is proposed. For the SM4 cryptographic algorithm, the fault is injected into the first 4 rounds of the encryption algorithm, so that the number of subsequent iterations of the encryption algorithm is reduced. Compared with the existing differential fault attack method for SM4, this method greatly and the attack efficiency. The experimental results show that the attack method is effective. This method can also be extended to other block ciphers.

Keywords: information security; hardware security; SM4 cryptographic algorithm; fault injection; block cipher; reduction wheel failure