分组密码 AES-128 的差分故障攻击

刘祥忠
(山东师范大学 第二附属中学,山东 济南 250014)

摘 要:AES 是美国数据加密标准的简称,又称 Rijndael 加密算法。它是当今最著名且在商业和政府部门应用最广泛的算法之一。AES 有三个版本,分别是 AES-128,AES-19 和 AES-256。AES 的分析是当今密码界的一个热点,文中使用差分故障攻击方法对 AES 进行分析。差分故障攻击假设攻击者可以给密码系统植入错误并获得正确密文和植入故障后密文,通过对两个密文分析比对从而得到密钥。文中提出了对 AES-128 的两种故障攻击方法,分别是在第 8 轮和第 7 轮的开始注入故障。两个分析方法分别需要 2 个和 4 个故障对,数据复杂度分别为 2^{34}(2^{112})次猜测密钥。

关键词:AES-128;分组密码;差分故障攻击
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2012)09-0221-04

A Differential Fault Analysis Attack Against AES-128

LIU Xiang-zhong
(No 2 Middle School Attached to Shandong Normal University, Jinan 250014, China)

Abstract: The advanced encryption standard is short for AES. It has another name Rijndael. It is one of the most popular ciphers in the world and is widely used for both commercial and government purposes. It has three versions(AES-128, AES-192 and AES-256). Differential fault analysis assumes that an attacker can induce faults into a system and collect the correct as well as the faulty behaviors. The attacker compares the two ciphers in order to retrieve the secret key. In this paper, present differential fault attacks on the block cipher AES-128 when error injected at the beginning of round 8 and round 7. The method proposed can recover subkey through 2 and 4 faults on average. The attack has a time complexity of 2^{34}(2^{112}) time for full key recovery for the two fault injected model correspondingly.

Key words: AES-128; block cipher; differential fault analysis

0 引 言

AES[1]是美国新的高级加密标准(Advanced Encryption Standard)的缩写,又称 Rijndael 加密算法,是美国联邦政府采用的一种区块加密标准。AES 是在 Square 密码基础上设计的一种结构为替换置换网络的分组密码,因为 Square 密码可以抵抗差分攻击和线性攻击,所以 AES 也可以抵抗这两种攻击。但由于 AES 线性层扩散的不完全性,先后出现了许多新攻击方法,包括不可能差分攻击[2,3]、飞来器攻击[4]、矩形攻击[5]、相关密钥攻击[6]、Biclique 攻击[7]等。文中提出了两种故障攻击方法。

故障攻击的概念是 1996 年由 Boneh 等人首次提出的,该方法利用了密码计算过程中的错误。这种攻击方法一经提出立即引起了人们的广泛关注,并展示了其对密码体制安全性的极大破坏性。1997 年 Biham 和 Shamir 将这种攻击方法应用于对称密码体制首次提出了差分故障攻击的概念并成功地攻击了 DES 算法,此后研究人员提出了各种不同的攻击方法成功攻击了多种密码体制如 ECC 公钥体制,RC4 算法等[8~12]。文中分别在第 7 轮和第 8 轮的开始注入故障,给出了 AES-128 的故障攻击,攻击只需要保证一定注入故障,不需要知道故障的确切数值。两个分析方法分别需要 2 个和 4 个故障对,数据复杂度分别为 2^{34}(2^{112})次猜测密钥。

1 AES 算法描述

AES 是一个明文分组为 128 比特的分组密码,其密钥长度有 128、192 和 256 比特三种,分别记作 AES-128、AES-192 和 AES-256。AES 针对不同的密钥长度需要的迭代轮数也不同,AES-128 需要迭代 10 轮,AES-192 迭代 12 轮,AES-256 迭代 14 轮。轮变换的每个状态可以形象地表示为一个 4×4 的矩阵形式,该矩阵中的每个元素是一个 8 比特字节,如表 1 所示。

AES 的轮函数由以下 4 种变换组成:非线性 8×8

收稿日期:2012-01-26;**修回日期**:2012-04-28
基金项目:山东省自然基金(Y2008G01);山东省高等学校优秀青年教师国内访问学者项目
作者简介:刘祥忠(1969-),男,山东烟台人,主要研究方向为密码分析和网络信息安全。

表 1 128 比特数据分组 AES 状态

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

的 S-盒替换、行循环移位、列混淆变换、密钥加法运算,除了第一轮之前有一个密钥加法运算,最后一轮没有列混淆运算,其他轮函数都是由 S-盒替换、行移位、列混淆、密钥加法这个顺序所构成。这里的列混淆变换就是把状态矩阵中的每列左乘矩阵 $\boldsymbol{M}$,$\boldsymbol{M}$ 的具体值、*AES* 算法详细和密钥编排算法请见文献[1]。

文中Δ表示一个随机故障,x 表示注入故障时状态矩阵的(1,1)元素,$\alpha = S(x+\Delta) + S(x)$,$y_i(i=1,2,3,4)$ 表示注入故障的下轮初始状态的$(i,1)$ 元素,$\beta_1 = S(2\alpha + y_1) + S(y_1)$,$\beta_2 = S(\alpha + y_2) + S(y_2)$,$\beta_3 = S(\alpha + y_3) + S(y_3)$,$\beta_4 = S(3\alpha + y_4) + S(y_4)$。

2 在第 8 轮开始注入故障的 AES-128 攻击

本节考虑在第 8 轮的开始注入故障的情形。若在第 8 轮的开始的第一个字节设置故障Δ,第 8 轮的差分扩散情况如下所示:

8 轮加密→

Δ			

8 轮初态

α			

进 S 盒

α			

Shift 后

2α			
α			
α			
3α			

乘 M 加 key

第 9 轮加密过程中的差分扩散情况如下所示:→

β_1			
β_2			
β_3			
β_4			

进 S 盒

β_1			
			β_2
		β_3	
	β_4		

Shift 后

$2\beta_1$	β_4	$3\beta_3$	β_2
β_1	β_4	$3\beta_3$	$2\beta_2$
β_1	$3\beta_4$	$2\beta_3$	β_2
$3\beta_1$	$2\beta_4$	β_3	β_2

乘 M

$2\beta_1$	β_4	$3\beta_3$	β_2
β_1	β_4	$3\beta_3$	$2\beta_2$
β_1	$3\beta_4$	$2\beta_3$	β_2
$3\beta_1$	$2\beta_4$	β_3	β_2

加 key

而将第 10 轮解密的过程如下:

10 轮解密←

$S^{-1}(c_1+k_1)$			
$S^{-1}(c_{14}+k_{14})$			
$S^{-1}(c_{11}+k_{11})$			
$S^{-1}(c_8+k_8)$			

进 S^{-1}、移位

c_1+k_1			
			$c_{14}+k_{14}$
		$c_{11}+k_{11}$	
	c_8+k_8		

脱密钥

	c_1		
			c_{14}
		c_{11}	
	c_8		

密文

这里 $c_1,\cdots,c_{16}$ 是密文中相应块。设 $c_1',\cdots,c_{16}'$ 为植入故障后所得密文相应块。基于以上扩散过程及第 9 轮加密的最后输出等于 10 轮解密完毕后的状态,提出以下攻击方法。

攻击细则:猜测最后一轮的 k_1,k_8,k_{11},k_{14} 共 32 比特,用它们解密密文,若所猜测密钥正确,则第 10 轮解密完毕后得结果与第 9 轮加密结束后的输出一致,也就是说,第 10 轮解密后输出的第 1 列满足关系式

$$\begin{aligned} &2S^{-1}(c_1' + k_1) + S^{-1}(c_1 + k_1) \\ &= S^{-1}(c_2' + k_8) + S^{-1}(c_2 + k_8) \\ &= S^{-1}(c_3' + k_{11}) + S^{-1}(c_3 + k_{11}) \\ &= 3S^{-1}(c_4' + k_{14}) + S^{-1}(c_4 + k_{14}) \end{aligned} \tag{1}$$

若(1)式有一个等号不成立,则也说明所猜测密钥是错误的。

计算复杂度与所需故障对个数:

对于四个随机数上面等式成立的概率为 $2^{-32} \times 2^8 = 2^{-24}$,所以一旦上述等式成立,就认为猜测正确,这样用一个故障对,过滤 k_1,k_8,k_{11},k_{14} 的全空间得 $2^{32} \times 2^{-24} = 2^8$ 个候选密钥,再考察另一个故障对,对所得 2^8 个候选密钥再次过滤,所剩候选密钥 $2^8 \times 2^{-24}$ 个,由于正确密钥一定被留下,所以两组(正确和错误对)故障对便可以概率 1 恢复密钥。因判断密钥是否正确的等式里的变量只与密文和第 10 轮密钥有关。于是可以恢复 k_1,k_8,k_{11},k_{14},同理可以分 3 次恢复其余 12 字节密钥,于是此法需要 2 个故障对,时间复杂度为 $2^{32} \times 4 = 2^{34}$。

对于故障攻击,判断攻击是否成功的准则是故障植入越早,就会被认为更有效,于是还考虑了更早植入故障的情形。

3 在第 7 轮开始注入故障的 AES-128 攻击

猜测第 10 轮的 $k_1,\cdots,k_{12},k_{15},k_{16}$ 共 112 比特的密钥。根据 AES 密钥编排规则,由所猜密钥可以求出第 9 轮中间两列密钥比特,如下所示 9 轮密钥之第 2、3 列。

根据第 10 轮密钥计算出的第 9 轮密钥

$k_1+s(k_{10}+k_{14})+Rcon$	k_1+k_5	k_5+k_9	k_9+k_{13}
$k_2+s(k_{11}+k_{15})$	k_2+k_6	k_6+k_{10}	$k_{10}+k_{14}$
$k_3+s(k_{12}+k_{16})$	k_3+k_7	k_7+k_{11}	$k_{11}+k_{15}$
$k_4+s(k_{13}+k_9)$	k_4+k_8	k_8+k_{12}	$k_{12}+k_{16}$

第 10 轮密钥

k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}
k_4	k_8	k_{12}	k_{16}

根据密文及猜测密钥解密第 10 轮的 8 个字节,继续将第 9 轮加密结果的中间两列脱密钥,再乘 $\boldsymbol{M}^{-1}$,接着循环移位,进 S 盒逆。就可以得到第 9 轮初始状态的 8 个字节,这 8 个字节应该与第 8 轮加密的最终状态的相同位置的 8 个字节对应相等,从而相应的原始明文的中间状态与植入故障后所得状态 8 个字节的差分也相等。如下所示:

第 8 轮加密结束后的状态差分

	β_4	β_3	
		$3\beta_3$	$2\beta_2$
β_1			β_2
$3\beta_1$	$2\beta_4$		

据猜测密钥和密文计算第 9 轮加密前状态差分

	ΔP_9^5	ΔP_9^9	
		ΔP_9^{10}	ΔP_9^{14}
ΔP_9^3			ΔP_9^{15}
ΔP_9^4	ΔP_9^8		

第 7 至第 10 轮加解密过程如下所示:

第 7 轮加密过程中差分变化情况 $\alpha = S(x+\Delta) + S(x)$

Δ			

初态差分

α			

进 S 盒后差分

α			

Shift 后差分

2α			
α			
α			
3α			

乘 M 加 key 后差分

↓

第 8 轮加密过程中差分变化情况

β_1			
β_2			
β_3			
β_4			

进 S 盒

β_1			
			β_2
		β_3	
	β_4		

Shift 后

$2\beta_1$	β_4	$3\beta_3$	β_2
β_1	β_4	$3\beta_3$	$2\beta_2$
β_1	$3\beta_4$	$2\beta_3$	β_2
$3\beta_1$	$2\beta_4$	β_3	β_2

乘 M

$2\beta_1$	β_4	$3\beta_3$	β_2
β_1	β_4	$3\beta_3$	$2\beta_2$
β_1	$3\beta_4$	$2\beta_3$	β_2
$3\beta_1$	$2\beta_4$	β_3	β_2

加 key

第 9 轮解密进 S^{-1}

	$P_9^5=S^{-1}(14\gamma_1+11\gamma_2+13\gamma_3+9\gamma_4)$	$P_9^9=S^{-1}(14\gamma_5+11\gamma_6+13\gamma_7+9\gamma_8)$	
		$P_9^{10}=S^{-1}(9\gamma_1+14\gamma_2+11\gamma_3+13\gamma_4)$	$P_9^{14}=S^{-1}(9\gamma_5+14\gamma_6+11\gamma_7+13\gamma_8)$
$P_9^3=S^{-1}(13\gamma_5+9\gamma_6+14\gamma_7+11\gamma_8)$			$P_9^{15}=S^{-1}(13\gamma_1+9\gamma_2+14\gamma_3+11\gamma_4)$
$P_9^4=S^{-1}(11\gamma_1+13\gamma_2+9\gamma_3+14\gamma_4)$	$P_9^8=S^{-1}(11\gamma_5+13\gamma_6+9\gamma_7+14\gamma_8)$		

第 9 轮解密右侧循环移位

	$14\gamma_1+11\gamma_2+13\gamma_3+9\gamma_4$	$14\gamma_5+11\gamma_6+13\gamma_7+9\gamma_8$	
		$9\gamma_1+14\gamma_2+11\gamma_3+13\gamma_4$	$9\gamma_5+14\gamma_6+11\gamma_7+13\gamma_8$
$13\gamma_5+9\gamma_6+14\gamma_7+11\gamma_8$			$13\gamma_1+9\gamma_2+14\gamma_3+11\gamma_4$
$11\gamma_1+13\gamma_2+9\gamma_3+14\gamma_4$	$11\gamma_5+13\gamma_6+9\gamma_7+14\gamma_8$		

↑

第 9 轮解密乘 M^{-1}

	$14\gamma_1+11\gamma_2+13\gamma_3+9\gamma_4$	$14\gamma_5+11\gamma_6+13\gamma_7+9\gamma_8$	
	$9\gamma_1+14\gamma_2+11\gamma_3+13\gamma_4$	$9\gamma_5+14\gamma_6+11\gamma_7+13\gamma_8$	
	$13\gamma_1+9\gamma_2+14\gamma_3+11\gamma_4$	$13\gamma_5+9\gamma_6+14\gamma_7+11\gamma_8$	
	$11\gamma_1+13\gamma_2+9\gamma_3+14\gamma_4$	$11\gamma_5+13\gamma_6+9\gamma_7+14\gamma_8$	

第 9 轮解密脱密钥

	$\gamma_1=S^{-1}(c_5+k_5)+k_1+k_5$	$\gamma_5=S^{-1}(c_5+k_5)+k_5+k_9$	
	$\gamma_2=S^{-1}(c_2+k_2)+k_2+k_6$	$\gamma_6=S^{-1}(c_6+k_6)+k_6+k_{10}$	
	$\gamma_3=S^{-1}(c_{15}+k_{15})+K3+k7$	$\gamma_7=S^{-1}(c_3+k_3)+k_7+k_{11}$	
	$\gamma_4=S^{-1}(c_{12}+k_{12})+k_4+k_8$	$\gamma_8=S^{-1}(c_{16}+k_{16})+k_8+k_{12}$	

↑

第 10 轮解密

S 盒逆移位

	$S^{-1}(c_5+k_5)$	$S^{-1}(c_9+k_9)$	
	$S^{-1}(c_2+k_2)$	$S^{-1}(c_6+k_6)$	
	$S^{-1}(c_{15}+k_{15})$	$S^{-1}(c_3+k_3)$	
	$S^{-1}(c_{12}+k_{12})$	$S^{-1}(c_{16}+k_{16})$	

脱密钥

c_1+k_1	c_5+k_5	c_9+k_9	
c_2+k_2	c_6+k_6	$c_{10}+k_{10}$	
c_3+k_3	c_7+k_7	$c_{11}+k_{11}$	
c_4+k_4	c_8+k_8	$c_{12}+k_{12}$	

密文

	c_5	c_9	
c_2	c_6		
c_3			c_{15}
		c_{12}	c_{16}

看此时是否有第 1 列后两个元素之比为 1 比 3,第 2 列 1、4 两元素之比为 1 比 2,第 3 列前两元素之比为 1 比 3,第 4 列中间两元素之比为 2 比 1 同时成立,也就是说下列 4 个等式

$$S^{-1}\{13[S^{-1}(c_5+k_5)+k_5+k_9]+9[S^{-1}(c_6+k_6)+k_6+k_{10}]+14[S^{-1}(c_3+k_3)+k_7+k_{11}]+11[S^{-1}(c_{16}+k_{16})+k_8+k_{12}]\}+S^{-1}\{13[S^{-1}(c_5'+k_5)+k_5+k_9]+9[S^{-1}(c_6'+k_6)+k_6+k_{10}]+14[S^{-1}(c_3'+k_3)+k_7+k_{11}]+11[S^{-1}(c_{16}'+k_{16})+k_8+k_{12}]\}=3S^{-1}\{11[S^{-1}(c_5+k_5)+k_1+k_5]+13[S^{-1}(c_2+k_2)+k_2+k_6]+9[S^{-1}(c_{15}+k_{15})+k_3+k_7]+14[S^{-1}(c_{12}+k_{12})+k_4+k_8]\}+3S^{-1}\{11[S^{-1}(c_5'+k_5)+k_1+k_5]+13[S^{-1}(c_2'+k_2)+k_2+k_6]+9[S^{-1}(c_{15}'+k_{15})+k_3+k_7]+14[S^{-1}(c_{12}'+k_{12})+k_4+k_8]\} \quad (2)$$

$$S^{-1}\{14[S^{-1}(c_5+k_5)+k_1+k_5]+11[S^{-1}(c_2+k_2)+k_2+k_6]+13[S^{-1}(c_{15}+k_{15})+k_3+k_7]+9[S^{-1}(c_{12}+k_{12})+k_4+k_8]\}+S^{-1}\{14[S^{-1}(c_5'+k_5)+k_1+k_5]+11[S^{-1}(c_2'+k_2)+k_2+k_6]+13[S^{-1}(c_{15}'+k_{15})+k_3+k_7]+9[S^{-1}(c_{12}'+k_{12})+k_4+k_8]\}=2S^{-1}\{11[S^{-1}(c_5+k_5)+k_5+k_9]+13[S^{-1}(c_6+k_6)+k_6+k_{10}]+9[S^{-1}(c_3+k_3)+k_7+k_{11}]+14[S^{-1}(c_{16}+k_{16})+k_8+k_{12}]\}+2S^{-1}\{11[S^{-1}(c_5'+k_5)+k_5+k_9]+13[S^{-1}(c_6'+k_6)+k_6+k_{10}]+9[S^{-1}(c_3'+k_3)+k_7+k_{11}]+14[S^{-1}(c_{16}'+k_{16})+k_8+k_{12}]\} \quad (3)$$

$$S^{-1}\{14[S^{-1}(c_5+k_5)+k_5+k_9]+11[S^{-1}(c_6+k_6)+k_6+k_{10}]+13[S^{-1}(c_3+k_3)+k_7+k_{11}]+9[S^{-1}(c_{16}+k_{16})+k_8+k_{12}]\}+S^{-1}\{14[S^{-1}(c_5'+k_5)+k_5+k_9]+11[S^{-1}(c_6'+k_6)+k_6+k_{10}]+13[S^{-1}(c_3'+k_3)+k_7+k_{11}]+9[S^{-1}(c_{16}'+k_{16})+k_8+k_{12}]\}=3S^{-1}\{9[S^{-1}(c_5+k_5)+k_1+k_5]+14[S^{-1}(c_2+k_2)+k_2+k_6]+11[S^{-1}(c_{15}+k_{15})+k_3+k_7]+13[S^{-1}(c_{12}+k_{12})+k_4+k_8]\}+3S^{-1}\{9[S^{-1}(c_5'+k_5)+k_1+k_5]+14[S^{-1}(c_2'+k_2)+k_2+k_6]+11[S^{-1}(c_{15}'+k_{15})+k_3+k_7]+13[S^{-1}(c_{12}'+k_{12})+k_4+k_8]\} \quad (4)$$

$$S^{-1}\{9[S^{-1}(c_5+k_5)+k_1+k_5]+14[S^{-1}(c_2+k_2)+k_2+k_6]+11[S^{-1}(c_{15}+k_{15})+k_3+k_7]+13[S^{-1}(c_{12}+k_{12})+k_4+k_8]\}+S^{-1}\{9[S^{-1}(c_5'+k_5)+k_1+k_5]+14[S^{-1}(c_2'+k_2)+k_2+k_6]+11[S^{-1}(c_{15}'+k_{15})+k_3+k_7]+13[S^{-1}(c_{12}'+k_{12})+k_4+k_8]\}=2S^{-1}\{13[S^{-1}(c_5+k_5)+k_5+k_9]+9[S^{-1}(c_6+k_6)+k_6+k_{10}]+14[S^{-1}(c_3+k_3)+k_7+k_{11}]+11[S^{-1}(c_{16}+k_{16})+k_8+k_{12}]\}+2S^{-1}\{13[S^{-1}(c_5'+k_5)+k_5+k_9]+9[S^{-1}(c_6'+k_6)+k_6+k_{10}]+14[S^{-1}(c_3'+k_3)+k_7+k_{11}]+11[S^{-1}(c_{16}'+k_{16})+k_8+k_{12}]\} \quad (5)$$

同时成立。若有一个不成立,也说明所猜测密钥错误。这样每运行一个故障可使猜测量减少为原来的 2^{-32},对于 2^{112} 个猜测密钥,只需四个明文便以概率 1 排除所有错误密钥,也就只剩下正确密钥,从而恢复密钥。于是此方法只需 4 个故障对,数据复杂度为 2^{112} 个猜测密钥。

4 结束语

文中分别在第 7 轮和第 8 轮的开始注入故障,给出了 AES-128 的故障攻击,攻击只需要保证一定注入故障,不需要知道故障的确切数值。两个分析方法分别需要 2 个和 4 个故障对,数据复杂度分别为 2^{34} (2^{112})次猜测密钥。如何更早的设置故障或者降低在第 7 轮设置故障时的数据复杂度是需要进一步研究的问题。

参考文献:

[1] Advanced Encryption Standard(AES), FIPS Publication 197 [S/OL]. 2001-11-26. http://csrc. nist. gov/encryption/aes.

[2] 陈 杰,胡予濮,张跃宇. 不可能差分分析高级加密标准[J]. 中国科学 E 辑:信息科学,2007,37(2):191-198.

[3] Phan W. Impossible differential cryptanalysis of 7-round advanced encryption standard[J]. Inf Proc Lett,2004,91:33-38.

[4] Wagner D. The boomerang attack[C]//Proceedings of Fast Software Encryption'99. Berlin: Springer-Verlag,1999:156-170.

[5] Biham E, Dunkelman O, Neller N. The rectangle attack-rectangling the Serpent[C]//Proceedings of Eurocrypt'2001. Berlin:Springer-Verlag,2001:340-357.

[6] Biham E, Dunkelman O, Keller N. Related-key impossible differential attacks on 8-round AES-192[C]//Proceeding of CT-RSA 2006. Berlin:Springer-Verlag,2006:21-33.

[7] Bogdanov A, Khovratovich D, Rechberger C. Biclique Cryptanalysis of the Full AES[R]. [s. l.]:[s. n.],2011.

[8] 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.

[9] 李 琳,李瑞林,谢端强,等. KeeLoq 和 SHACAL-1 算法的差分故障攻击[J]. 武汉大学学报(理学版),2008,54(5):507-512.

[10] 魏悦川,李 琳,李瑞林,等. SHACAL-2 算法的差分故障攻击[J]. 电子与信息学报,2010,32(2):323-328.

[11] 赵新杰,王 韬,王素贞,等. MIBS 深度差分故障分析研究[J]. 通信学报,2010(12):82-89.

[12] 王 哲,张文英. 对 5 轮 Square 的中间相遇攻击[J]. 计算机技术与发展,2011,21(5):132-135.

分组密码 AES-128的差分故障攻击

作者：刘祥忠
作者单位：山东师范大学 第二附属中学,山东 济南 250014
刊名：计算机技术与发展
英文刊名：Computer Technology and Development
年，卷(期)：2012(9)