

EMV 应用密文的差分错误注入分析

彭乾, 李增局, 史汝辉

(国家金融 IC 卡安全检测中心, 北京 100070)

摘要: 研究了 EMV 规范中应用密文的生成过程, 发现过程密钥存在部分冗余位, 结合 DES 算法 S 盒的压缩特性, 利用基于碰撞的 safe-error 攻击实现对 EMV 规范中应用密文主密钥的破解。提出了针对应用密文生成的差分错误注入的物理模型和实施步骤, 深入分析了影响攻击效果的 2 个关键因素(密钥错误产生概率和碰撞概率), 尤其是对不同错误模型进行了理论数据分析。实验分析表明, 实际攻击中, 只要不同密钥的碰撞概率差大于 0.003 5 即可区分, 结果表明, 过程密钥的冗余位提高了碰撞概率, 有利于对正确密钥的区分。最后, 针对该攻击方法, 提出了几种防御方案。

关键词: EMV; 应用密文; 碰撞攻击; safe-error

中图分类号: TP309.7

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2016.00044

Differential fault analysis on EMV application cryptogram

PENG Qian, LI Zeng-ju, SHI Ru-hui

(National Financial IC Card Test Center, Beijing 100070, China)

Abstract: The process of application cryptogram in EMV was researched and dummy bits in session key were found. Based on the session key's dummy bits and compressive property of DES's Sbox, much information of the application cryptogram master key was got by using safe-error attack. The differential fault attack model and steps to implement the attack were proposed, two key factors(the probability of generating wrong key and the probability of collision happening) affecting an attacking result were analyzed. The theoretical result and simulation of the attack were given. The experiment results show that the two keys could be distinguished in a real attacking when the difference of two key's collision probability was bigger than 0.003 5. The dummy bits in the key will increase the difference and make distinguishing easier. Finally, several countermeasures against the attack were proposed.

Key words: EMV, application cryptogram, collision attack, safe-error

1 引言

EMV 规范是由国际三大信用卡组织 Europay、MasterCard、Visa 联合制定的, 其中, 金融集成电路卡支付标准是目前国际上绝大多数银行卡使用的标准。我国的 PBOC3.0 规范也是基于 EMV 规范建立的, 并根据特定需求, 在 EMV 规范框

架上进行了细微的调整。

在 EMV 规范中, 标识金融 IC 卡不可抵赖性有对称密钥体系和非对称密钥体系。对称密钥体系主要是完成金融 IC 卡的联机交易, 基于该对称密钥的应用密文和 MAC 等机制保证了 IC 卡交易的不可抵赖性和可信性。非对称密钥体系主要是基于三级密钥证书体系保证金融 IC 卡在脱机交

收稿日期: 2016-01-14; 修回日期: 2016-03-27。通信作者: 李增局, liecas@sina.com

基金项目: 国家科技重大专项基金资助项目 (No.2014ZX01032401)

Foundation Item: National Science and Technology Major Project (No.2014ZX01032401)

易中的可信性和不可抵赖性。如果能破解金融 IC 卡的对称密钥,则完成了卡片复制,甚至进行部分发卡行数据的改写,完成伪卡交易。

目前,EMV 规范^[1]中主要用对称密码算法 3DES,密钥长度为 128 位,实际有效长度为 112 位。3DES 的差分错误注入分析在 1997 年已经被提出。经过多年的发展,有很多关于 3DES 的差分错误注入分析模型^[2~5]。随着错误注入的发展,针对 EMV 规范中 RSA 签名算法的攻击方式成果也很多。Coron 等^[6]提出了针对 EMV 规范签名机制的攻击。但是,到目前为止,还没有针对 EMV 规范中 3DES 的分析,主要原因是在 EMV 规范中使用的加密算法协议,明密文不断变化,无法对同一明密文进行差分错误注入分析。因此,本文首先提出了针对 EMV 规范中 3DES 的差分错误注入的分析。

2 背景知识

2.1 EMV 应用密文

EMV 规范中对应用密文^[1]的定义为:在交易过程中,由 IC 卡将相关交易数据(如交易金额、交易货币代码、终端国家代码、终端验证结果、交易日期、交易类型、随机数、卡片应用交互特征以及卡片验证结果等)经 3DES(CBC 模式、双倍长密钥)加密生成的密文。终端收到卡片应用密文,直接送到后台,后台可以将应用密文的有效性返回给终端,终端发送给卡片,确定该应用密文是否有效。

应用密文生成时使用的 3DES 加密密钥为过程密钥,由 IC 卡使用应用密文主密钥对交易计数器(ATC)进行 3DES 加密生成。ATC 为 2 个字节长度,初始为 1,每笔交易后自动加 1,上限为 65 535,因此过程密钥为一次一密,每笔交易都不同。应用密文的过程密钥产生过程如下。

算法 1 过程密钥生成算法

如果用双长度 DES 密钥生成,步骤如下。

1) 生成过程密钥的卡片密钥为:数据加密 3DES 密钥 A 和 B (ENC UDK)。

2) 将两字节的 ATC 右对齐,前面补 6 个字节 00,使用数据加密 3DES 密钥 A 和 B 加密生成过程密钥 KA 。

3) 将两字节的 ATC 取反后右对齐,前面补 6 个字节 00,使用数据加密 3DES 密钥 A 和 B 加密生成过程密钥 KB 。

上述 1) 中使用的 DES 密钥 A 和 B 是每张卡片唯一的应用密文主密钥。

应用密文产生过程如下。

算法 2 应用密文生成算法

1) 将交易数据块分成 8 字节一组: $D1$ 、 $D2$ 、 $D3$...

2) 如果最后一块数据块的长度为 8 个字节,后面补 8 个字节数据块: 80 00 00 00 00 00 00 00。

如果最后一块数据块的长度小于 8 个字节,后面补一个字节 80,如果仍然不够 8 个字节,补 00 直到 8 个字节。

3) 使用算法 1 产生的过程密钥用对称密钥算法生成应用密文。

图 1 是使用过程密钥 KA 和 KB 生成应用密文的流程。其中, I =输入; D =数据块; $DEA(a)$ =数据加密算法(加密模式); KA =密钥 A ; $DEA(d)$ =数据加密算法(加密模式); KB =密钥 B ; O =输出; $+$ =异或。

2.2 碰撞攻击

碰撞攻击算法最早是出现在散列算法中^[7]。碰撞攻击要求算法的映射函数是多对一映射,只有满足该条件才有可能产生碰撞。而 DES 算法 Sbox 映射是 64 输入对应 16 输出,因此,它具有碰撞攻击的可行性。随着功耗分析、差分错误分析的提出,研究者逐渐关注 DES 的碰撞性质,并利用碰撞性质得到密钥^[3,4]。在功耗分析技术中,攻击者主要利用碰撞性质及其功耗特征来确定碰撞行为是否发生。如果发生了碰撞,则可以通过建立方程或增加密钥约束条件的方法缩小 DES 密钥的搜索空间。

碰撞攻击比差分功耗分析或相关性功耗分析等方法普适性要差一些,主要原因有 3 点: 1) 较难建立合适的碰撞模型; 2) 由于噪声的存在,使通过功耗分析确定碰撞发生有一定难度,而差分错误分析、错误注入发生概率的问题,使碰撞分析难度加大; 3) 通过碰撞直接得到密钥的难度很大。

2.3 Safe-error

Safe-error 攻击是差分错误分析发展过程中

最简便易行的一种攻击手段。Safe-error 攻击可以分为操作性 safe-error^[8]和数据性 safe-error^[9]。最初, RSA 算法为了防御简单功耗分析和时间分析, 会在实现过程中插入冗余运算。Safe-error 不需要关注错误的结果, 而只是通过判断最终的结果正确与否来获得信息。如果冗余操作与密钥位有关系, 比如密钥位为 1, 则不加入冗余操作; 而密钥位为 0, 则加入冗余操作, 通过对操作注入错误, 比较最终结果, 如果最终结果出现错误, 则操作不是冗余的, 得到的密钥位为 1; 反之, 则操作是冗余的, 得到密钥位为 0。通过结果的正确与否, 即可以判断密钥信息。这种攻击过程属于操作性的 safe-error。

在 DES 运算过程中, 有一种错误模型, 无论其原始值是多少, 能将 DES 任意密钥比特置为 1。因此, 针对某个 DES 密钥比特进行操作, 如果加密结果正确, 说明 DES 原始密钥比特为 1, 反之, 则说明 DES 该比特为 0。依次对不同密钥进行这种实验, 就可以逐个比特得到 DES 密钥。这种方法就是属于数据性 safe-error。

3 攻击原理

3.1 错误模型及碰撞位置

根据错误注入发生错误的模型分类, 可分为固定值错误、固定翻转错误、随机翻转错误、部分比特固定错误等类型。本文研究 3 种错误模型: 目标数据只发生 0→1 翻转错误、目标数据只发生

1→0 翻转错误、目标数据发生随机翻转错误。

本文假设目标数据的比特位之间是相互独立的, 且每个比特产生错误的概率是相同的。目标数据的多个比特的错误概率可以使用二项分布的模型来计算。

本文采用碰撞攻击的方法, DES 运算中的 Sbox 变换为 6 进 4 出的非线性运算, 因此, 存在碰撞的可能性。Sbox 变换式为

$$Sboxout = Sbox(k \oplus datain) \quad (1)$$

本文研究对象为式(1)中的 k , 对 k 注入错误得到 k' 。因此, 对于明文输入 $datain$, 从 0~63 遍历, 计算每一个明文分别对应 k 和 k' 的 $Sboxout$ 的值, 如果两者相等, 则存在碰撞。

攻击的目标操作为 DES 或 3DES 运算的最后一轮的 Sbox 运算, 3.2 节会详细分析选择最后一轮作为攻击目标的原因。

3.2 攻击模型

在生成应用密文过程密钥的过程中, 如果应用密文主密钥加密 ATC 生成的密文结果 (即应用密文过程密钥) 每个字节的最低位 (奇偶校验位) 产生错误, 卡片仍然能生成正确的应用密文, 因为对应的这些奇偶校验位作为密钥时不参与 DES 加密运算, 不会影响 3DES 的结果, 定义为冗余位。因此, 根据银行后台对应用密文校验的结果, 可以判断应用密文是否正确, 从而形成 safe-error 攻击。本节将详细分析过程密钥的 8 个冗余位带来的碰撞和 Sbox 自身碰撞的性质是否有利于破

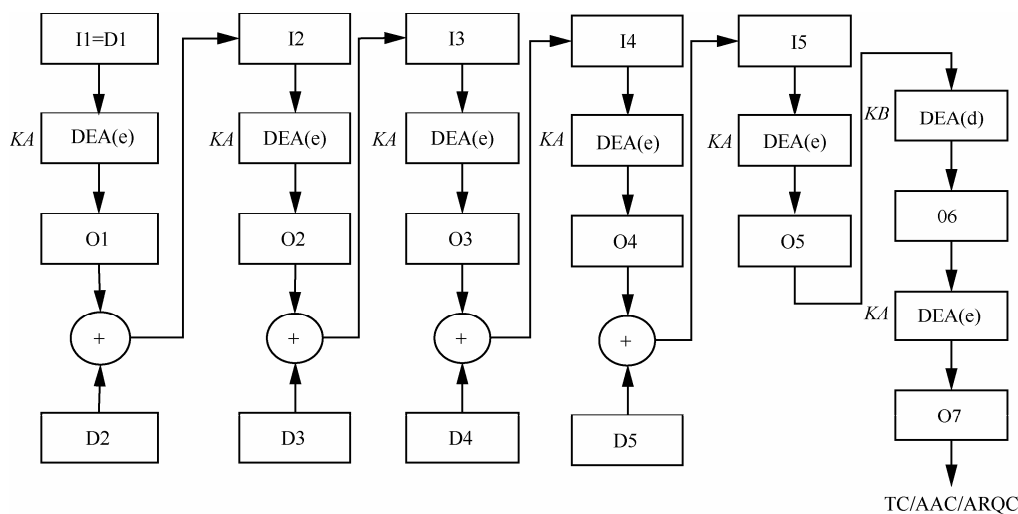


图1 用双长度 DES 密钥生成应用密文

解卡片的应用密文主密钥。

DES/3DES 最后一轮的运算如图 2 所示。其中, L 和 R 分别是 DES 的左寄存器和右寄存器, F 是 DES 和 S 盒置换, P 是 32 位置换表, IP^{-1} 是 DES 的最后置换。

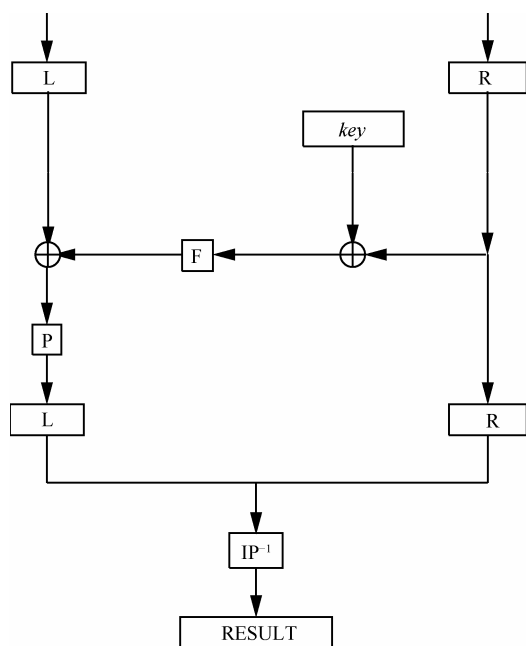


图2 DES/3DES 最后两轮运算流程

根据 DES 运算的流程可知, 主密钥加密结果中有 8 位冗余位, 即应用密文过程密钥的 8 位校验位, 分别为第 8、16、24、32、40、48、56、64 位, 这些比特不参与应用密文的加密运算。DES 的末尾逆置如图 3 所示, 该置换是一个一一映射的置换, 通过数据变化增强了数据的混淆程序。

40	8	48	16	56	24	64	40
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

图3 IP^{-1} 置换

根据图 2 的算法流程可知, 第 16 轮的 L||R 经过了 IP^{-1} 置换得到最终结果, 因此, 密钥的 8 为

校验位在 IP^{-1} 置换前分别为第 32、31、30、29、28、27、26、25 位, 即为第 16 轮 L 寄存器结果的最后一个字节。图 4 是 DES 每轮中使用的置换, 通过置换和 Sbox 的非线性效应, 达到了混淆和置乱的目的, 使密码算法具有雪崩效应。

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

图4 P 置换

根据图 2 的算法流程可知, 第 16 轮的 L 寄存器结果是由 Sbox 输出经 P 置换得到, 因此, L 寄存器的最后一个字节分别对应 Sbox 输出的第 19、13、30、6、22、11、4、25 位, 即 Sbox5 的第 3 位、Sbox4 的第 1 位、Sbox8 的第 2 位、Sbox2 的第 2 位、Sbox6 的第 2 位、Sbox3 的第 3 位、Sbox1 的第 4 位、Sbox7 的第 1 位。

3.1 节中的模型对一般的碰撞定义为

$$Sbox(k \oplus datain) = Sbox(k' \oplus datain) \quad (2)$$

在增加了 8 位冗余位之后, 碰撞的定义需要将冗余位考虑进去, 对于 Sbox1, 冗余位为第 4 位, 因此只需要考虑前 3 位, 新的碰撞定义为

$$Sbox(k \oplus datain) \& 0xE = Sbox(k' \oplus datain) \& 0xE \quad (3)$$

对比式(2)、式(3)可知, 式(3)的解包含了式(2)的解。因此, 使用 3.1 节的攻击模型, 同时引入 8 位冗余位的信息, 理论上可以提高 Sbox 碰撞的概率, 增加密钥的区分度。其他 Sbox 的碰撞方程变化原理与 Sbox1 类似。

考虑到应用过程密钥产生时使用 ATC, 对 3DES 进行加密, 攻击 3DES 第 48 轮时, 可以认为每个 Sbox 的输入明文具有充分的统计随机性, 因此, 对于 DES Sbox 的 2 个密钥 k 和 k' 的碰撞概率, 本文定义为数据 $datain$ 从 0~63 遍历, 然后观察式(3)成立的个数。如果个数越多, 说明碰撞概率越大。

由于应用密文过程生成过程为双倍长密钥的 3DES 加密, 使用 3.1 节中的攻击模型只能对 3DES 的最后一轮进行攻击, 因此只能攻击 3DES 的第一个密钥, 第二个密钥需要暴力破解。对单个 DES

密钥的破解不在本文的讨论范围内。

由于应用密文的产生过程是受 ATC 控制的, 正常交易 ATC 是 2 个字节长度, 即最大值为 65 536。由 2.1 节描述可知, 应用密文产生过程中两部分过程密钥存在联系, 一次交易过程中产生两次应用密文和一次外部认证, 其中一次应用密文和一次外部认证的 DES 运算结果是否正确可以得知。综上, 使用应用密文主密钥进行的有效 DES 运算约 130 000 ($65\,536 \times 2$) 次, 并且这些运算的结果是否正确都可得知。考虑到实际中错误注入会存在概率, 在有限执行次数中需要一定概率差值才能区分不同的密钥。

3.3 影响攻击的关键因素

在有限次运算中, 如果在对 Sbox 子密钥进行错误注入后, 不同的 Sbox 子密钥产生碰撞概率的差异足够大, 那么就可以通过密文的正确率来区分不同的子密钥。设 P_{happen} 为错误注入导致子密钥产生错误的概率, P_{collide} 为因子密钥发生错误而产生碰撞的概率, 则密文的正确率为

$$P_{\text{right}} = 1 - P_{\text{happen}}(1 - P_{\text{collide}}) \quad (4)$$

因此, 影响应用密文正确率的因素有 2 个: 一是密钥错误产生概率 P_{happen} ; 二是碰撞概率 P_{collide} 。

3.3.1 密钥错误产生概率

对于错误发生概率, 假设单比特错误发生概率为 P , 在 0->1 翻转错误模型下, 设 6 比特 Sbox 子密钥中 0 的比特个数为 m , 则二阶近似模型的错误发生概率为 $C_m^2 P^2 (1-P)^{m-2}$ 。

以子密钥数据为 0x1 为例, $m=5$, 则 2 个比特发生错误概率为 $C_5^2 P^2 (1-P)^3$, 不发生错误的概率为 $C_5^5 P^0 (1-P)^5$, 发生一个比特错误概率为 $C_5^1 P^1 (1-P)^4$ 。

实际中错误发生概率较小时, 如 $P=0.2$, 在 0->1 翻转模型下, 若 $m<3$, 不可能发生 3 比特以上错误; 若 $m=3$, 发生 3 比特错误概率 ($0.2^3=0.008$) 比发生 2 个比特错误概率 ($C_3^2 \cdot 0.2^2 \cdot 0.8 = 0.092$) 小很多; 若 $m=4$, 发生 3 比特以上错误概率 (0.027) 比发生 2 比特错误概率 ($C_4^2 \cdot 0.2^2 \cdot 0.8^2 = 0.154$) 小很多; 若 $m=5$, 发生 3 比特以上错误概率 (0.062) 比发生 2 个比特错误概率 (0.205) 小很多; 若 $m=6$, 则发生 3 个比特以上错误概率 (0.1) 比

发生 2 个比特错误概率 (0.246) 小很多。并且随着 P 减小, 2 比特错误概率要高于 3 比特以上错误发生概率。

对于 0->1 的错误模型和 1->0 的错误模型, 不同密钥数据的错误概率是不同的, 而对于随机错误模型来说, 密钥数据的错误概率是相同的。例如 b000001 和 b111110 对于 0->1 和 1->0 的错误模型的错误概率是完全不同的, 但是对于随机错误模型则是相同的。

而由式(4)可知, 对于不同密钥数据, 若 P_{collide} 不变, 则 P_{happen} 的变化越大, 则密钥越容易区分。因此若只考虑 P_{happen} , 则 0->1 的错误模型和 1->0 的错误模型要优于随机错误模型。

3.3.2 碰撞概率

在 DES 实现中, 存在串行 DES Sbox 查表方案, 如果采用这种方案, 则可以独立对每个 Sbox 查表过程进行错误注入分析。由前面分析可知, EMV 应用密文生成过程中存在 130 000 次可攻击过程, 而需要分别验证不同 Sbox 的密钥, 因此, 每个 Sbox 可利用过程约为 $16\,000 \left(\frac{130\,000}{8} \right)$ 次。

下面以 Sbox1 为例分析获取密钥的可能性, 首先讨论错误产生的概率大小对碰撞的影响, 然后计算每个 Sbox 在不同错误模型下的碰撞概率。

计算 Sbox1 的碰撞概率, 根据式(3), 其他 Sbox 的碰撞概率表都参考实现。这里, 首先计算 Sbox1 的不同密钥碰撞概率表。碰撞概率计算方式根据式(3)进行, 使用算法是算法 3。

算法 3 Sbox1 碰撞概率计算算法

输入: 无

输出: Sbox1 碰撞概率

for $i=0:63$

for $j=0:63$

for $k=0:63$

if $\text{Sbox}(i \wedge k) \& 0xE == \text{Sbox}(j \wedge \text{datain}) \& 0xE$

Count[i][j]++;

Count[i][j] /= 64;

根据算法 3 计算 Sbox1 的碰撞概率表, 错误模型采用 2.1 节中描述的 3 种模型。

假设 Sbox1 密钥每个比特独立同分布, 将错误模型细分为一阶碰撞、二阶碰撞和高阶碰撞。一阶碰撞为密钥 key 翻转任意 1 比特产生的碰撞。

二阶碰撞为翻转任意 2 比特产生的碰撞。高阶碰撞则为翻转任意 3 比特以上的碰撞。

由于 DES Sbox 具有良好的非线性性质, 输入改变任 1 比特, 输出至少改变 2 个比特, 因此, 由式(3)可知, 不存在一阶碰撞的可能性。而由 3.3.1 节分析可知, 高阶碰撞的错误概率要远小于二阶碰撞的错误概率, 因此只研究二阶碰撞。

由于随机错误模型包含 0->1 错误模型和 1->0 的错误模型, 因此将二阶随机模型重新定义为: 翻转的 2 比特中 0->1 和 1->0 同时发生。

表 1 为 Sbox1 的二阶碰撞概率。由表 1 可知, 0->1 错误模型和 1->0 的错误模型对于互补的数据碰撞概率是相同的; 随机翻转的碰撞概率较其他模型变化较小, 这主要是由于随机模型的二阶翻转变化较少。

表 1

Sbox1 二阶碰撞概率

数据	翻转概率 (0 到 1)	翻转概率 (1 到 0)	翻转概率 (随机)	数据	翻转概率 (0 到 1)	翻转概率 (1 到 0)	翻转概率 (随机)
000000	0.129 166 66	0.0	0.0	100000	0.121 875	0.0	0.143 75
000001	0.121 875	0.0	0.143 75	100001	0.093 75	0.062 5	0.164 062 5
000010	0.112 5	0.0	0.162 5	100010	0.109 375	0.25	0.128 906 25
000011	0.098 958 336	0.187 5	0.144 531 25	100011	0.062 5	0.166 666 67	0.138 888 9
000100	0.140 625	0.0	0.106 25	100100	0.151 041 67	0.218 75	0.101 562 5
000101	0.119 791 664	0.031 25	0.148 437 5	100101	0.093 75	0.104 166 664	0.149 305 55
000110	0.130 208 33	0.187 5	0.121 093 75	100110	0.177 083 33	0.218 75	0.083 333 336
000111	0.093 75	0.135 416 67	0.138 888 9	100111	0.093 75	0.156 25	0.113 281 25
001000	0.137 5	0.0	0.112 5	101000	0.130 208 33	0.125	0.128 906 25
001001	0.151 041 67	0.25	0.097 656 25	101001	0.125	0.145 833 33	0.125
001010	0.109 375	0.093 75	0.148 437 5	101010	0.104 166 664	0.156 25	0.128 472 22
001011	0.125	0.177 083 33	0.114 583 336	101011	0.093 75	0.161 458 33	0.109 375
001100	0.140 625	0.0	0.136 718 75	101100	0.156 25	0.114 583 336	0.125
001101	0.135 416 67	0.093 75	0.138 888 9	101101	0.093 75	0.114 583 336	0.144 531 25
001110	0.104 166 664	0.093 75	0.149 305 55	101110	0.187 5	0.145 833 33	0.109 375
001111	0.062 5	0.125	0.140 625	101111	0.0	0.140 625	0.106 25
010000	0.140 625	0.0	0.106 25	110000	0.125	0.062 5	0.140 625
010001	0.145 833 33	0.187 5	0.109 375	110001	0.093 75	0.104 166 664	0.149 305 55
010010	0.114 583 336	0.093 75	0.144 531 25	110010	0.093 75	0.135 416 67	0.138 888 9
010011	0.114 583 336	0.156 25	0.125	110011	0.0	0.140 625	0.136 718 75
010100	0.161 458 33	0.093 75	0.109 375	110100	0.177 083 33	0.125	0.114 583 336
010101	0.156 25	0.104 166 664	0.128 472 22	110101	0.093 75	0.109 375	0.148 437 5
010110	0.145 833 33	0.125	0.125	110110	0.25	0.151 041 67	0.097 656 25
010111	0.125	0.130 208 33	0.128 906 25	110111	0.0	0.137 5	0.112 5
011000	0.156 25	0.093 75	0.113 281 25	111000	0.135 416 67	0.093 75	0.138 888 9
011001	0.218 75	0.177 083 33	0.083 333 336	111001	0.187 5	0.130 208 33	0.121 093 75
011010	0.104 166 664	0.093 75	0.149 305 55	111010	0.031 25	0.119 791 664	0.148 437 5
011011	0.218 75	0.151 041 67	0.101 562 5	111011	0.0	0.140 625	0.106 25
011100	0.166 666 67	0.062 5	0.138 888 9	111100	0.187 5	0.098 958 336	0.144 531 25
011101	0.25	0.109 375	0.128 906 25	111101	0.0	0.112 5	0.162 5
011110	0.062 5	0.093 75	0.164 062 5	111110	0.0	0.121 875	0.143 75
011111	0.0	0.121 875	0.143 75	111111	0.0	0.129 166 66	0.0

4 仿真实验

4.1 仿真环境

本仿真实验平台为 PC 一台, 配置如下。

操作系统 Windows XP Professional

CPU Intel Pentium(双核 2.7 G)

内存 2 GB

仿真工具 Microsoft Visual Studio 2005

编程语言 C

4.2 实验步骤

参数说明: N 为 DES 随机密钥 (K) 的个数, K_{R16} 为 DES 第 16 轮的子密钥, $DES_{15}()$ 为 DES 的前 15 轮运算, $Sbox()$ 为 DES 的 $Sbox$ 置换运算。

- 1) $i = 0$;
- 2) 判断 i 是否小于 N , 若是, 执行 3), 否则执行 12);
- 3) $ATC = 1$, 生成 DES 算法 64 bit 随机密钥 K ;
- 4) 使用 KEY 生成 DES16 轮子密钥 K_{R16} ;
- 5) 以 P_1 、 P_2 、 P_3 、 P_4 、 P_5 的概率对 K_{R16} 进行错误注入 (采用 A、B、C 3 种模型) 得到 K_{R16}' ;
- 6) 判断 ATC 是否小于 65 536, 若是执行 7), 否则执行 11);
- 7) 计算 $R_{15} = DES_{15}(K, ATC)$,
 $Sboxout_{16} = Sbox(R_{15}, K_{R16})$,
 $Sboxout_{16}' = Sbox(R_{15}, K_{R16}')$;
- 8) 分别在考虑冗余位和不考虑冗余位的情况下, 判断 $Sboxout_{16}$ 和 $Sboxout_{16}'$ 是否碰撞, 若是, 执行 9), 否则执行 10);

9) 将发生碰撞所对应 $Sbox$ 子密钥计数器加 1;

10) ATC 加 1, 执行 6);

11) 将 KEY_{R16} 对应的 8 个 $Sbox$ 的密钥计数器加 1, i 加 1, 执行 2);

12) 计算碰撞发生的概率。

4.3 仿真结果分析

图 5 为 $Sbox1$ 在 0->1 翻转模型下的不同出错概率 P_{happen} 的输出正确概率。由图可知, 随着 P_{happen} 的增大, 碰撞的概率逐渐减小, 这是因为当出错概率提高后, 结果出错的概率增大, 因此正确率下降。同时可以看出, 随着 P 的增大, 不同密钥的正确率的差异变小, 正确率的变化幅度变小。由之前的分析可知, 本文的攻击方案的密钥区分器是以正确率的差异作为重要指标的, 因此, 为了提高攻击成功率, 出错的概率不宜过大。

图 6 为 $Sbox1$ 在随机翻转模型下不同出错概率 P_{happen} 的输出正确率。与图 5 对比可知, 随机模型较 0->1 模型的概率曲线波动要小很多, 并且随着 P_{happen} 的增大, 波动越来越小。

图 7 为 $Sbox1$ 在 0->1 模型下的正确率对比, 虚线为考虑冗余位的概率曲线, 实线为未考虑冗余位, 即仅 $Sbox$ 自身性质导致的。由图可知, 虚实线在相同的 P 的情况下, 趋势一致。但是随着 P 的增大, 虚线与实线的差距越来越大, 说明随着 P 的增大, 冗余位导致的影响越来越大。

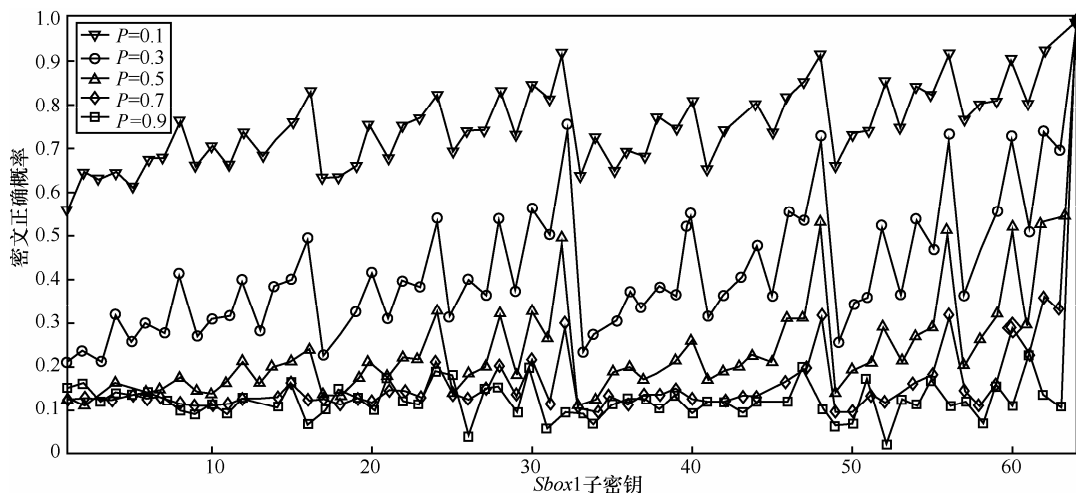


图5 0->1 翻转模型下不同错误概率的 $Sbox1$ 正确率

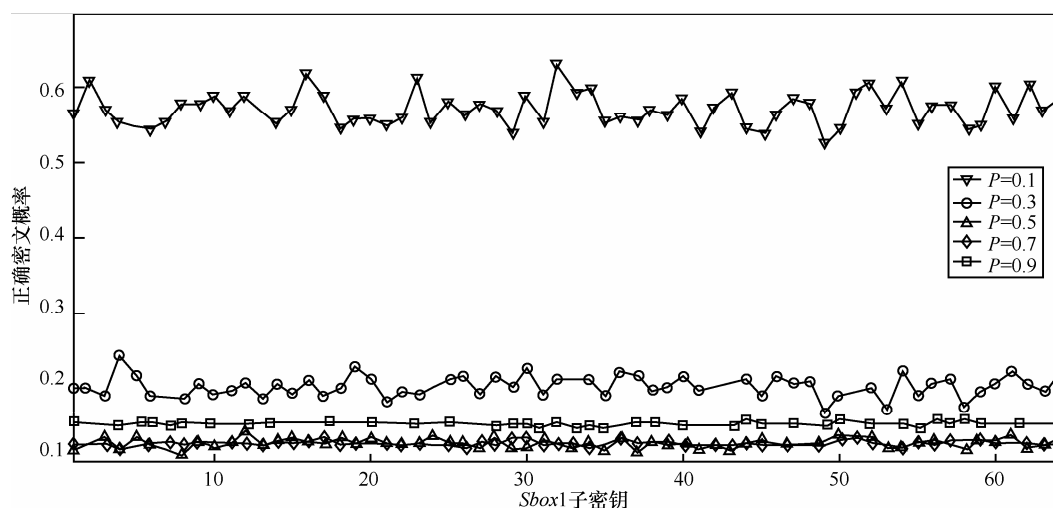


图6 随机翻转模型下不同错误概率的 Sbox1 正确率

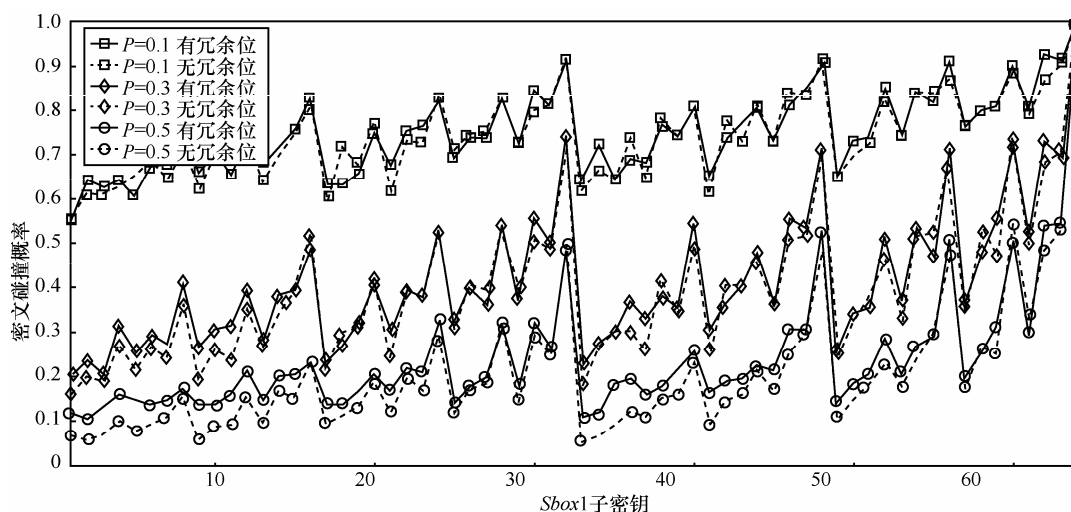


图7 0->1 模型下的 Sbox1 正确率对比

5 防御方案

1) 反向校验

根据应用密文攻击原理分析, 本文提出了一种简单有效的防御策略: 应用密文过程密钥生成后, 进行反向运算 (3DES 解密运算), 然后比较验证加密的输入与解密的输出是否一致, 如果一致, 说明未受到错误注入, 输出结果; 否则, 触发警报, 停止运算, 清除敏感数据。使用反向校验措施, 可以校验 8 个冗余比特是否正确, 从而降低碰撞概率。

2) 使用替代算法

由 2.1 节的分析可知, 冗余位是应用密文生成过程中的弱点, 提高了碰撞的概率, 因此可以考虑让冗余位参与到实际的运算中, 冗余位的改变

会使最后的应用密文改变, 那么也可以避免该漏洞。从现行 EMV 规范看, 使用 AES 或者 SMS4(PBOC 3.0)算法, 可以避免类似的碰撞攻击。

3) 数据校验算法

在数字电路设计中, 可以使用 CRC 等算法对数据过程结果进行实时保护计算, 保证数据运算正确性, 这样冗余的比特可以通过类似校验算法进行保护, 因此可以抵抗本文提出的攻击方法。

6 结束语

本文通过研究发现, 在 EMV 规范中 3DES 算法产生应用密文的过程中, 存在部分过程密钥的冗余。攻击者可以利用这种冗余产生碰撞性攻击, 并利用 safe-error 方法判断碰撞攻击是否存在。通过统

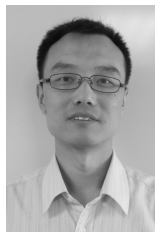
计结果的正确性, 可以获取很多比特的密钥信息, 从而对卡片的密钥信息造成损害。基于这种攻击方法的原理, 本文给出了几种建议的防御方案, 以提高卡片防御类似攻击手段的能力。

参考文献:

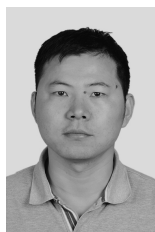
- [1] EMV. Integrated circuit card specifications for payment systems, book 2: security and key management, version4.2[EB/OL]. <http://www.emvco.com>.
- [2] BIHAM E, SHAMIR A. Differential fault analysis of secret key cryptosystems[J]. Lncs, 1999, 1294:513-525.
- [3] SCHRAMM K, WOLLINGER T, PAAR C. A new class of collision attacks and its application to DES[C]//International Conference Fast Software Encryption(FSE). c2003: 206-222.
- [4] RIVAIN M. Differential fault analysis on DES middle rounds[J]. Lecture Notes in Computer Science, 2009, 5747: 457-469.
- [5] LEE C. A new approach of differential fault analysis on block ciphers with S-box[J]. International Journal on Information, 2013(16): 1915-1928.
- [6] CORON J S, NACCACHE D, TIBOUCHI M. Fault attacks against EMV signatures[C]//International Conference Topics in Cryptology. c2010: 208-220.
- [7] CONTINI S, YIN Y L. Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions[M]// Advances in Cryptology. Berlin Heidelberg: Springer, 2006:37-53.
- [8] YEN S M, KIM S, LIM S, et al. A countermeasure against one physical cryptanalysis may benefit another attack[C]//International Conference Information Security and Cryptology. c2002: 414-427.
- [9] YEN S M, JOYE M. Checking before output may not be enough

against fault-based cryptanalysis[J]. IEEE Transactions on Computers, 2002, 49(9): 967-97.

作者简介:



彭乾(1982-), 男, 蒙古族, 内蒙赤峰人, 硕士, 国家金融 IC 卡安全检测中心工程师, 主要研究方向为金融 IC 卡、金融 Pos 机安全技术。



李增局(1982-), 男, 山东莘县人, 硕士, 国家金融 IC 卡安全检测中心工程师, 主要研究方向为侧信道、错误注入、密码学工程实现以及借贷记交易安全性等。



史汝辉(1985-), 男, 山东烟台人, 硕士, 国家金融 IC 卡安全检测中心工程师, 主要研究方向为侧信道、错误注入、密码学工程实现以及借贷记交易安全性等。