

PRINCE 轻量级密码算法的差分故障分析

邹 伟 李 浪 焦 铭

(衡阳师范学院计算机学院 衡阳 421002)

摘 要 PRINCE 密码算法是于 ASIA CRYPT 2012 提出的轻量级的加密算法,用于在物联网环境下保护 RFID 标签以及智能卡等设备的通信安全。提出并讨论了一种针对 PRINCE 算法的差分故障分析方法。该方法采用半字节故障模型,对 PRINCEcore 最后一轮进行了差分故障分析。实验结果表明,在 PRINCEcore 最后一轮导入半字节随机故障,4 次故障注入可实现对 PRINCE 算法 PRINCEcore 部分的 64 位轮密钥的恢复。因此,未加防护措施的 PRINCE 加密系统将难以抵御差分故障分析手段。

关键词 PRINCE,轻量级密码算法,差分故障分析

中图法分类号 TP309 文献标识码 A

Differential Fault Analysis of PRINCE Lightweight Cryptographic Algorithm

ZOU Yi LI Lang JIAO Ge

(College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China)

Abstract PRINCE is a lightweight cryptographic algorithm proposed in CRYPT ASIA 2012, for the protection of RFID tags and the communication security of the devices such as smart cards in the Internet of things, as smart cards. A differential fault analysis method was proposed and discussed in this paper. The method is based on the PRINCE algorithm. In this method, a semi-byte fault model is adopted, and the differential fault analysis is carried out on the last PRINCEcore. Experimental results show that the semi-byte random fault is injected into the last round of the PRINCEcore, the key(k_1) can be cracked by 4 fault injections to PRINCEcore part. Therefore, without the protection, the PRINCE encryption algorithm will be difficult to resist differential fault analysis.

Keywords PRINCE, Lightweight cryptographic algorithms, Differential fault analysis

1 引言

近几年,物联网越来越深入到人们的生活中,同时物联网安全也引起了人们的高度关注。PRINCE 算法是 2012 年亚密会上由 Borgho 等人提出的轻量级分组密码算法^[1],该密码算法主要为资源受限的物联网智能卡加密研发,是一种超轻量级的对称加密算法。

旁路攻击^[2]是当前进行密码分析的一种有效手段,它是利用中间状态信息对密码算法的实现进行分析,其主要的攻击类型有故障攻击、时间攻击、功耗攻击等。Biham 和 Shamir 在 1997 年的美密会^[3]上针对数据加密标准(DES)提出了 DFA(差分故障攻击)技术,该技术利用加密算法的特点和最大似然估计技术来推测加密系统中的关键信息,其密钥的搜索空间远小于差分密钥分析和线性密钥分析等方法,可以破译出目前市场上大部分智能卡。由此可见,DFA 技术对加密系统而言是一种严重的威胁^[4-5],因此,对密码算法进行故障攻击的研究对信息安全系统具有重要意义。

2 PRINCE 密码算法简介

PRINCE 密码算法是一种 SPN 结构的轻量级分组密码算法^[1]。其明文分组长度为 64bit,用 P 表示;密钥长度为 128bit,用 K 表示,其中 $K=k_0 \parallel k_1$ 。

算法前后分别进行一次白化密钥,中间 PRINCEcore 部分经过 12 轮轮变换,包括两类轮函数 R 和 R' ,中间对合轮记为 2 轮。本文主要对 PRINCEcore 进行分析,其结构图以及 PRINCE 加密过程分别如图 1 和图 2 所示^[6-7]。



图 1 PRINCEcore 结构图

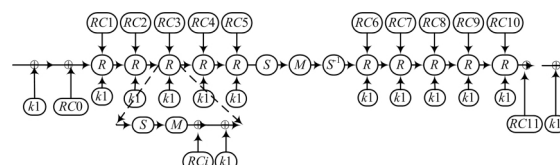


图 2 PRINCE 加密过程

本文受国家自然科学基金资助项目(61572174),湖南省自然科学基金资助项目(2015JJ4011),湖南省教育厅科学研究优秀青年项目(16B039),湖南省教育厅科研一般项目(15C0203),衡阳师范学院产学研一般项目(14CXYY02)资助。

邹 伟(1983—),女,硕士,讲师,主要研究方向为嵌入式与信息安全,E-mail:51674074@qq.com。

如图 2 所示,前 6 轮与后 6 轮的算法结构基本互逆,其中 $RC[i] \wedge RC[11-i] = \text{Constant} = \text{C0AC29B7C97C50DD}$, $RC[i]$ ($i=0,1,\dots,11$) 表示第 i 轮的轮常数,其值如表 1 所列。

表 1 $RC[i]$ 的取值

$RC[i]$	值
$RC[0]$	0000 0000 0000 0000
$RC[1]$	1319 8a2e 0370 7344
$RC[2]$	a409 3822 299f 31d0
$RC[3]$	082e fa98 ec4e 6c89
$RC[4]$	4528 21e6 38d0 1377
$RC[5]$	be54 66cf 34e9 0c6c
$RC[6]$	7ef8 4f78 fd95 5cb1
$RC[7]$	8584 0851 flac 43aa
$RC[8]$	C882 d32f 2532 3c54
$RC[9]$	64a5 1195 e0e3 610d
$RC[10]$	D3b5 a399 ca0c 2399
$RC[11]$	C0ac 29b7 c97c 50dd

假设 $X = X_{15} X_{14} \dots X_1 X_0$ 表示 64bit 的明文或密文或中间状态,并按序排列成 4×4 的状态矩阵,每个元素包含 4bit。其轮函数 R 由轮密钥加(AK)、轮常数加(AC)、矩阵乘和 S 盒变换 4 个部分组成,其步骤为:

1) 轮密钥加(AK):轮密钥与状态矩阵按位异或,输出为 64bit 中间状态。

2) 轮常数加(AC):轮常数与状态矩阵按位异或,输出为 64bit 中间状态。

3) S 盒变换:S 盒为 4×4 的结构,其具体变换如表 2 所列。

表 2 S 盒变换

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4

4) P 置换(矩阵乘): $M = SR * M'$, 其中 SR 是对状态矩阵的行移位, M' 是一个 64×64 的对角型对合矩阵,表示为 $M' = \text{diag}(\hat{M}^0, \hat{M}^1, \hat{M}^2, \hat{M}^3)$ 。其中, $\hat{M}^0$ 和 $\hat{M}^1$ 的矩阵结构如式(1)和式(2)所示, M_0, M_1, M_2, M_3 的矩阵结构如式(3)和式(4)所示。

$$\hat{M}^0 = \begin{pmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_1 & M_1 & M_2 \end{pmatrix} \quad (1)$$

$$\hat{M}^1 = \begin{pmatrix} M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \end{pmatrix} \quad (2)$$

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4)$$

轮函数的结构图(以 PRINCEcore 部分的最后一轮变换

为例)如图 3 所示。

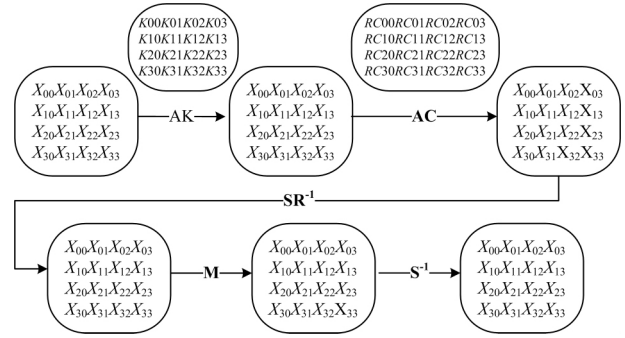


图 3 轮函数结构图

3 PRINCE 差分故障分析

3.1 故障模型

由于 PRINCE 算法的 S 盒为 16 个不同的基于半字节的非线性变换,因此本文采用的故障模型是面向半字节的随机故障模型^[8]。假设故障发生在 PRINCEcore 部分的最后一轮轮变换输入的某个随机半字节(发生翻转)上。对于同一明文 P , C 表示对应明文 P 正确加密时产生的密文, C' 表示对应明文在注入错误后加密产生的错误密文,攻击者可获得在同一密钥 $K1$ 作用下的正确密文 C 与错误密文 C' 。并假设对于同一位置,攻击者可以重复导入故障并获取相应的错误密文。通过采集到若干个密文对 (C, C') , 令 $\Delta C = C \oplus C'$, 差分故障攻击通过分析 ΔC 来推测密钥。

3.2 差分故障攻击原理

对密码体制进行故障攻击,大致按以下 3 个步骤展开:

- 1) 选择明文 P , 对加密过程进行故障诱导;
- 2) 数据收集: 获取明文对应的正确密文 C 与错误密文 C' ;
- 3) 对收集的数据进行分析,恢复密钥。

3.3 PRINCE 算法的差分故障攻击过程

攻击过程的相关定义如下。

PRINCEcore 部分 64bit 分组块输入表示为 $X_{ij} \in (F_2^4)^{16}$

(其中 $0 < i < 3, 0 < j < 3$), 用矩阵表示为:

$$\begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

64bit 密文输出表示为 $Y_{ij} \in (F_2^4)^{16}$ (其中 $0 < i < 3, 0 < j < 3$),

用矩阵表示为:

$$\begin{bmatrix} Y_{00} & Y_{01} & Y_{02} & Y_{03} \\ Y_{10} & Y_{11} & Y_{12} & Y_{13} \\ Y_{20} & Y_{21} & Y_{22} & Y_{23} \\ Y_{30} & Y_{31} & Y_{32} & Y_{33} \end{bmatrix}$$

K_{ij} 分别包括 64bit 的白化密钥 k_0 和轮密钥 k_1 , 用矩阵表示为:

$$\begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$State_{ij}$ 表示状态矩阵, 其中 $i=0,1,2,3; j=0,1,2,3$ 。

即:

$$\begin{bmatrix} State_{00} & State_{01} & State_{02} & State_{03} \\ State_{10} & State_{11} & State_{12} & State_{13} \\ State_{20} & State_{21} & State_{22} & State_{23} \\ State_{30} & State_{31} & State_{32} & State_{33} \end{bmatrix}$$

针对 PRINCE 算法而言, 假设故障发生在 PRINCEcore 的最后一轮的某个随机半字节(发生翻转)输入, 错误传播过程如图 4 所示。

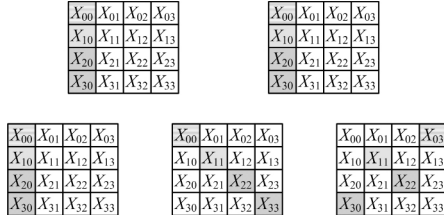


图 4 PRINCEcore 最后一轮输入注入半字节故障的差分传播过程

其攻击的基本过程如下:

首先, 假设在 PRINCEcore 的最后一轮输入中注入一个随机半字节的故障, 则可计算 P 置换对应列输出的可能差分列表 D , D 中包含 $4 \times 24 = 64$ 个 4 组半字节元素。

其次, 得到的正确密文和错误密文组成密文对 (C, C') 。如果故障被注入到状态矩阵的列数不同, 则错误密文与正确密文的不同值也不一样, 其错误传播轨迹分别为:

1) 当故障注入为第 1 列, 则 C' 和 C 的第 3, 5, 10, 12 个半字节的值不同;

2) 当故障注入为第 2 列, 则 C' 和 C 的第 0, 6, 11, 13 个半字节的值不同;

3) 当故障注入为第 3 列, 则 C' 和 C 的第 1, 7, 8, 14 个半字节的值不同;

4) 当故障注入为第 4 列, 则 C' 和 C 的第 2, 4, 9, 15 个半字节的值不同。

假设在 PRINCEcore 最后一轮的输入状态矩阵注入一个半字节故障后, 输出矩阵的第 i, j, k, l 4 个半字节对应的差分值为非零值, 故障影响到最后一轮密钥的 4 个半字节 K_i, K_j, K_k, K_l 参与的运算。通过穷举这 4 个半字节对应的 24 个候选值, 为每个候选值计算出:

$$\begin{aligned} \Delta i &= SB^{-1}(C_i - K_i) - SB^{-1}(C'_i - K_i) \\ \Delta j &= SB^{-1}(C_j - K_j) - SB^{-1}(C'_j - K_j) \\ \Delta k &= SB^{-1}(C_k - K_k) - SB^{-1}(C'_k - K_k) \\ \Delta l &= SB^{-1}(C_l - K_l) - SB^{-1}(C'_l - K_l) \end{aligned}$$

将计算得到的 4 个值与列表 D 中包含的 64 个 4 组半字节元素进行匹配, 匹配成功的候选值 (K_i, K_j, K_k, K_l) 存入列表 key 中。然后通过对于一个密文对 (C, C') 进行分析, 可得 key 表中包含的元素。经过多次在同一位置注入另一个故障并分析得到的密文对, 可确定 PRINCEcore 部分的 16bit 密钥 ($k1$)。

最后, 重复 4 次上述过程, 即可恢复 PRINCEcore 部分的轮密钥 $k1$ 的全部 64bit。在此基础上, 1 次故障注入即可白化密钥 $k0$ 。

3.4 攻击结果

在 PC 机(配置: CPU 为 Intel(R)Core(TM) 3.4GHz, 内存为 8GB)上使用 C++ 编程实现上述分析方法, 进行了 50 次实验, 选取其中的 8 次, 实验结果如表 3 所列。实验结果表明, 除全 1 或者全 0 的情况仅需 1 个故障密文外, 平均需要 4 个故障密文, 便可在该方案下恢复 64bit 初始密钥的 $k1$ 部分。

表 3 PRINCE 算法故障分析的实验结果

序号	分析 $k1$ 需要的故障密文
1	1
2	4
3	4
4	8
5	16
6	3
7	4
8	4

结束语 本文给出了一种针对 PRINCE 算法的面向半字节的差分故障攻击。与目前已有的方法对比, 尽管故障的前提假设相同, 但该方案导入的半字节故障有效提高了故障导入的效率, 降低了恢复子密钥所需的故障密文数。实验结果表明, 本文所提的方法有较好的利用率, 平均 4 个故障密文即可恢复 64bit 初始密钥的 $k1$ 部分。通过本文的分析可知, 差分故障攻击对 PRINCE 算法是十分有效的。为了避免这类攻击, 需要对加密设备进行保护, 阻止攻击者对其进行故障诱导。

下一步的工作将从对 PRINCE 算法进行抗攻击防护方面展开研究, 以提高算法本身的抗攻击分析的防御能力。

参考文献

- [1] BORGHO J, CANTEAUT A, GUNEYSU T, et al. PRINCE-A Low-latency Block Cipher Pervasive Computing Applications[C]// Proc of the 18th International Conference on the Theory and Application of Cryptology and Information Security. Beijing, China, 2012: 208-225.
- [2] KELSEY J, SCHNEIER B, WANGNER D, et al. Side channel crypt-analysis of product ciphers[C]// The European Symposium on Research in Computer Security-ESORICS'98. Louvain-la-Neuve, Belgium, 1998: 97-110.
- [3] BIHAM E, SHAMIR A. Differential fault analysis of secret key cryptosystems[C]// Proc of Advances in Cryptology-Crpto'97. 1997: 513-525.
- [4] PIRET G, QUISQUATER J J. A Differential Fault Attack Technique against SPN Structures, with Application to AES and KHAZAD[C]// Cryptographic Hardware & Embedded Systems (CHES), International Workshop. Cologne, Germany, 2003: 77-88.
- [5] 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.
- [6] 成磊, 孙兵, 李超. 对 PRINCE 算法的旁路 Cube 分析[C]// 2013 (第六届) 全国网络与信息安全学术会议. 2013: 107-114.
- [7] 李浪, 杜国权, 曾婷, 等. PRINCE 密码算法代数攻击研究[J]. 数学的实践与认识, 2015, 45(5): 153-159.
- [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.