密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research, 2021, 8(1): 1–13

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

8 轮 PRINCE 的快速密钥恢复攻击*

段春晖[1], 谭 林[1,2], 戚文峰[1,2]

1. 中国人民解放军战略支援部队信息工程大学, 郑州 450001
2. 数学工程与先进计算国家重点实验室, 郑州 450001

通信作者: 谭林, E-mail: tanlin100@163.com

摘 要: PRINCE 算法是 J. Borghoff 等在 2012 年亚密会上提出的一个轻量级分组密码算法, 它模仿 AES 并采用 α-反射结构设计, 具有加解密相似的特点. 2014 年, 设计者发起了针对 PRINCE 实际攻击的公开挑战, 使得该算法的安全性成为研究的热点. 目前对 PRINCE 攻击的最长轮数是 10 轮, 其中 P. Derbez 等利用中间相遇技术攻击的数据和时间复杂度的乘积 $D \times T = 2^{125}$, A. Canteaut 等利用多重差分技术攻击的复杂度 $D \times T = 2^{118.5}$, 并且两种方法的时间复杂度都超过了 2^{57}. 本文将 A. Canteaut 等给出的多重差分技术稍作改变, 通过考虑输入差分为固定值, 输出差分为选定的集合, 给出了目前轮数最长的 7 轮 PRINCE 区分器, 并应用该区分器对 8 轮 PRINCE 进行了密钥恢复攻击. 本文的 7 轮 PRINCE 差分区分器的概率为 $2^{-56.89}$, 8 轮 PRINCE 的密钥恢复攻击所需的数据复杂度为 $2^{61.89}$ 个选择明文, 时间复杂度为 $2^{19.68}$ 次 8 轮加密, 存储复杂度为 $2^{15.21}$ 个 16 比特计数器. 相比目前已知的 8 轮 PRINCE 密钥恢复攻击的结果, 包括将 A. Canteaut 等给出的 10 轮攻击方案减少到 8 轮, 本文给出的攻击方案的时间复杂度和 $D \times T$ 复杂度都是最低的.

关键词: 分组密码; PRINCE; 差分分析

中图分类号: TN918.1 **文献标识码:** A DOI: 10.13868/j.cnki.jcr.000415

中文引用格式: 段春晖, 谭林, 戚文峰. 8 轮 PRINCE 的快速密钥恢复攻击[J]. 密码学报, 2021, 8(1): 1–13. [DOI: 10.13868/j.cnki.jcr.000415]

英文引用格式: DUAN C H, TAN L, QI W F. Faster key recovery attack on 8-round PRINCE[J]. Journal of Cryptologic Research, 2021, 8(1): 1–13. [DOI: 10.13868/j.cnki.jcr.000415]

Faster Key Recovery Attack on 8-Round PRINCE

DUAN Chun-Hui[1], TAN Lin[1,2], QI Wen-Feng[1,2]

1. PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China
2. State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Corresponding author: TAN Lin, E-mail: tanlin100@163.com

Abstract: PRINCE is a lightweight block cipher proposed by J. Borghoff et al. at ASIACRYPT 2012. Imitating AES and using α-reflection design, it possesses the similarity of encryption and decryption. In 2014, the designers launched a public challenge on finding practical attacks on PRINCE. Currently, attacks on PRINCE can reach up to 10 encryption rounds. P. Derbez et al. used meet-in-the-middle

* 基金项目: 国家自然科学基金 (61521003); 国家密码发展基金 (MMJJ20170103, MMJJ20180204)
Foundation: National Natural Science Foundation of China (61521003); National Cryptography Development Fund (MMJJ20170103, MMJJ20180204)
收稿日期: 2020-01-07 定稿日期: 2020-05-09

technique to attack PRINCE with the data complexity and time complexity satisfying $D \times T = 2^{125}$, and A. Canteaut et al. used multiple differential cryptanalysis to attack PRINCE with the data complexity and time complexity satisfying $D \times T = 2^{118.5}$. The time complexity of both the two attacks exceeds 2^{57}. This paper slightly changes the multiple differential cryptanalysis given by A. Canteaut. By considering the case when the input difference is a fixed value and the output difference falls into a selected set, a distinguisher on 7-round PRINCE with the longest number of rounds is given, which can be used to lunch a key recovery attack on 8-round PRINCE. The differential probability of 7-round PRINCE differential distinguisher designed in this paper is $2^{-56.89}$. The key recovery attack on 8-round PRINCE is given with data complexity being $2^{61.89}$ chosen plaintext, time complexity being $2^{19.68}$ 8-round PRINCE encryption, and memory complexity being $2^{15.21}$ of 16-bit counters. Compared with the results of key recovery attacks on 8-round PRINCE, including reducing the 10-round attack given by A. Canteaut et al. to 8-round, the time complexity and $D \times T$ complexity given in this paper are both the lowest.

Key words: block cipher; PRINCE; differential cryptanalysis

1 引言

随着移动通信和物联网的发展, 射频识别系统 (RFID) 和智能卡等设备的加密被广泛应用, 海量信息加密和有限资源处理间的矛盾日益突显, 传统的加密算法无法适用于资源受限的环境, 密码算法的轻量化越来越受到关注. PRESENT [1], SIMON 和 SPECK 等 [2] 轻量级密码算法先后被提出, 它们使用小规模的密码组件, 在硬件实现方面有着明显的优势. PRINCE [3] 密码算法是 J. Borghoff 等在 2012 年的亚密会上提出的一个轻量级分组密码, 它基于 FX 结构 [4] 设计, 具有 α-反射性质, 解密过程可以通过稍微改变密钥进行加密来实现. 这种特性使 PRINCE 在硬件实现上具有优势, 但也使得其安全性受到担忧, H. Soleimany 等 [5] 针对某些特定的 α 值在相关密钥模式下可以攻击全轮的 PRINCE 变种, 但这里不包括 PRINCE 算法的 α 值.

学者们对 PRINCE 算法的安全性进行了大量分析, 表 1 罗列了目前在单密钥模式下对 PRINCE 算法不同轮数版本的主要分析结果. J. Jean 等给出了 4 轮和 6 轮 PRINCE 的积分攻击 [6]. 王小云团队使用中间相遇攻击方法攻击了 8 轮和 9 轮 PRINCE [7]. 虽然 PRINCE 的轮密钥除了首尾白化外都相同, 但仍假设差分特征概率等于各轮差分概率的乘积, A. Canteaut 等 [8] 构造了 5 轮和 6 轮的多重差分区分器, 攻击了 9 轮和 10 轮 PRINCE; 赵光耀等 [9] 给出了 5 轮和 6 轮的截断差分区分器, 并攻击了 7 轮 $\text{PRINCE}_{\text{core}}$. P. Derbez 等 [10] 将中间相遇方法和 SAT 方法相结合, 攻击了 10 轮 PRINCE. P. Morawiecki 利用积分和高阶差分分析, 给出了 4 轮至 7 轮 PRINCE 实际的攻击 [11]. 随后, R. Posteuca 等改进了 6 轮 PRINCE 的积分攻击, 降低了数据量和计算量 [12]. 利用预存储技术, S. Rasoolzadeh 等改进了 4 至 6 轮的积分攻击和 7 轮的高阶差分攻击 [13]. L. Grassi 等利用子空间路径给出了只需要 8 个选择明文的 4 轮 PRINCE 的截断差分攻击 [14].

本文将文献 [8] 的多重差分技术稍作改变, 考虑输入差分为固定值, 输出差分为选定的集合, 给出了目前轮数最长的 7 轮 PRINCE 区分器, 并对 8 轮 PRINCE 进行了密钥恢复攻击. 本文给出的 7 轮差分区分器的概率为 $2^{-56.89}$, 8 轮 PRINCE 的密钥恢复攻击所需的数据复杂度为 $2^{61.89}$ 个选择明文, 时间复杂度为 $2^{19.68}$ 次 8 轮加密, 存储复杂度为 $2^{15.21}$ 个 16 比特计数器. 相比目前已知的 8 轮 PRINCE 密钥恢复攻击的结果, 包括将 A. Canteaut 等的 10 轮攻击方案减到 8 轮, 本文的时间复杂度和 $D \times T$ 复杂度都是最低的.

2 PRINCE 密码算法简介

PRINCE 算法是一个 SPN 型的分组密码, 分组长度为 64 比特, 密钥长度为 128 比特, 迭代轮数为 12 轮, 算法结构如图 1 所示. 128 比特的密钥被分为 2 个 64 比特的子密钥 k_0 和 k_1, 其中 k_1 应用于

表 **1** 不同轮数 PRINCE 算法的攻击结果
Table 1 Cryptanalysis results of different rounds PRINCE

轮数	数据复杂度 [1]	时间复杂度	存储复杂度	攻击方法	参考文献
4	2^{10}	—[2]	$\ll 2^{27}$	中间相遇分析	[10]
	2^{3}	$2^{18.3}$	—	截断差分分析	[14]
	2^{6}	$2^{7.4}$	—	积分分析	[13]
5	2^{5}	$2^{21.4}$	2^{5}	积分分析	[13]
	2^{13}	2^{13}	2^{5}	积分分析	[13]
6	2^{16}	$2^{33.7}$	$2^{31.9}$	中间相遇分析	[10]
	2^{13}	$2^{24.6}$	2^{13}	积分分析	[13]
7	$2^{34.6}$	$2^{52.1}$	$2^{34.6}$	高阶差分分析	[11]
	2^{33}	$2^{44.3}$	2^{33}	高阶差分分析	[13]
8	2^{53}	2^{60}	2^{30}	中间相遇分析	[7]
	2^{16}	$2^{50.7}*^{3}$	$2^{84.9}$	中间相遇分析	[10]
	2^{16}	$2^{65.7}*$	$2^{68.9}$	中间相遇分析	[10]
	2^{16}	$2^{66.3}$	$2^{49.9}$	中间相遇分析	[10]
	$2^{61.89}$	$2^{19.68}$	$2^{15.21}$	多重差分分析	第 5 节
9	$2^{46.9}$	$2^{51.2}$	$2^{52.2}$	多重差分分析	[8]
	2^{57}	2^{64}	$2^{57.3}$	中间相遇分析	[10]
10	$2^{57.9}$	$2^{60.6}$	$2^{61.5}$	多重差分分析	[8]
	2^{57}	$2^{68}*$	2^{41}	中间相遇分析	[10]

1. 数据量均为选择明文
2. —表示复杂度几乎为 0
3. * 处标注数值为线上计算的复杂度

核心部件 $\text{PRINCE}_{\text{core}}$, k_0 和 k_0' 用于算法输入、输出两端的白化, 这里 $k_0' = (k_0 \ggg 1) \oplus (k_0 \gg 63)$. $\text{PRINCE}_{\text{core}}$ 采用对称结构, 中间 2 轮是对合的, 前后 5 轮除轮常数不同外互为逆变换, 轮常数满足 $\text{RC}_i \oplus \text{RC}_{11-i} = \alpha, 0 \le i \le 11$, 这里 α 是个固定常数. PRINCE 算法可以通过加密操作来实现解密, 即 $D_{(k_0 \| k_0' \| k_1)}(\cdot) = E_{(k_0' \| k_0 \| k_1 \oplus \alpha)}(\cdot)$.

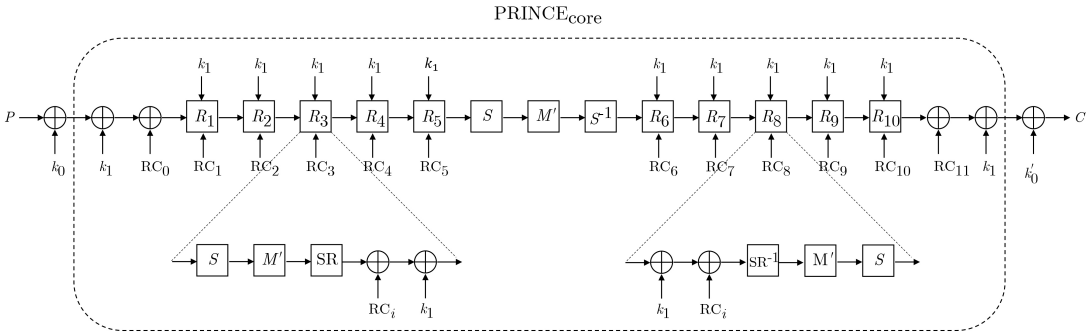

图 1 PRINCE 算法结构图
Figure 1 Structure of PRINCE

将 PRINCE 的 64 比特状态 X 看成一个 4×4 的矩阵, 每一个块单元为 4 比特, 本文中我们记 $X^{(l)}$ 为表示 X 的第 l 块, $l \in \{0, 1, \cdots, 15\}$, 块的顺序如图 2 所示. 算法的轮函数可表示为 $R = \text{AK} \circ \text{ARC} \circ$

SR $\circ M' \circ S$, 包括以下 5 个变换:

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

图 2 状态 X 的 16 个块标记
Figure 2 State X

S 层: 16 个块同时查询一个 4 比特的 S 盒, S 盒如表 2 所示.

表 **2** PRINCE 的 S 盒
Table 2 S-box of PRINCE

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

扩散层 M': 扩散层对应对角矩阵 $M' = \mathrm{diag}(\hat{M}^0, \hat{M}^1, \hat{M}^1, \hat{M}^0)$, 作用在状态 X 上为

$$M' \cdot X = (\hat{M}^0 \cdot X_1 || \hat{M}^1 \cdot X_2 || \hat{M}^1 \cdot X_3 || \hat{M}^0 \cdot X_4)$$

这里 X_i 表示 X 的第 i 列, $1 \le i \le 4$, 矩阵 $\hat{M}^0, \hat{M}^1$ 分别为:

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

其中

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

行移位 SR: 与 AES 的行移位操作相同, 将状态的第 i 行向左循环移动 i 个块, $0 \le i \le 3$.
轮常数加 ARC: 比特异或一个 64 比特轮常数 $\mathrm{RC}_i, 0 \le i \le 11$.
密钥加 AK: 比特异或 64 比特密钥 k_1.

3 准备工作

设 R 是一个 SPN 型分组密码的轮函数, Δ_{in} 和 Δ_{out} 分别表示其输入差分和输出差分, 则经过 R 的差分概率 $P(\Delta_{\mathrm{in}} \xrightarrow{R} \Delta_{\mathrm{out}})$ 等于活动 S 盒的差分概率的乘积. 设 $\Delta_{\mathrm{in}} = \Delta_0 \to \Delta_1 \to \cdots \to \Delta_{r-1} \to \Delta_r = \Delta_{\mathrm{out}}$ 是密码经过 r 轮的一条差分路径, 根据马尔科夫链的假设, 该差分特征概率等于各轮差分概率的乘

积, 即

$$P(\Delta_{\text{in}} \to \Delta_1 \to \cdots \to \Delta_{r-1} \to \Delta_{\text{out}}) = \prod_{i=0}^{r-1} P(\Delta_i \xrightarrow{R} \Delta_{i+1})$$

设 Δ_{in} 和 Δ_{out} 分别表示 r 轮的输入差分和输出差分, 则 r 轮差分概率

$$P(\Delta_{\text{in}} \xrightarrow{R^r} \Delta_{\text{out}}) = \sum P(\Delta_{\text{in}} \to \Delta_1 \to \cdots \to \Delta_{r-1} \to \Delta_{\text{out}})$$

这里的求和式是对所有输入差分为 Δ_{in}、输出差分为 Δ_{out} 的 r 轮差分特征概率求和. 由于我们很难穷尽所有的差分特征, 所以通常使用部分差分特征概率求和来近似计算 $P(\Delta_{\text{in}} \xrightarrow{R^r} \Delta_{\text{out}})$. 如果考虑输出差分不是某个固定值, 而是某个集合 Λ, 则输入差分为 Δ_{in}、输出差分属于集合 Λ 的概率

$$P(\Delta_{\text{in}} \xrightarrow{R^r} \Lambda) = \sum_{\Delta_{\text{out}} \in \Lambda} P(\Delta_{\text{in}} \xrightarrow{R^r} \Delta_{\text{out}})$$

PRINCE 算法 S 盒的差分分布表见附录, 记 $p(\alpha \to \beta)$ 表示 S 盒输入差分为 α、输出差分为 β 的概率, 例如 $p(1 \to 1) = \frac{4}{16} = 2^{-2}$.

下面介绍本文用到的八种差分模式, 如图3所示, 记作 $\Delta^i, i \in \{1, 2, \cdots, 8\}$. 每个差分模式 Δ^i 只有 4 个块有差分, 其余块差分为 0, 属于模式 Δ^i 的差分由 4 个非零差分块决定, 将其记为 $\Delta^i_{(a,b,c,d)}$, 其中 a, b, c, d 是非零差分块的差分值.

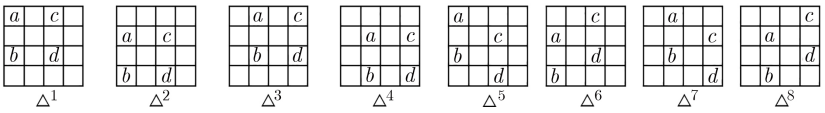

图 3 差分模式 $\Delta^i, i = 1, 2, \cdots, 8$
Figure 3 Difference pattern $\Delta^i, i = 1, 2, \cdots, 8$

文献 [9] 指出了矩阵 $\hat{M}^0$ 和 $\hat{M}^1$ 具有如下性质: 如果 $\delta \in \{1, 4, 5\}$, 则

$$\begin{aligned}
\hat{M}^0 \cdot (0, \delta, 0, \delta)^{\mathrm{T}} &= (\delta, 0, \delta, 0)^{\mathrm{T}} \\
\hat{M}^0 \cdot (\delta, 0, \delta, 0)^{\mathrm{T}} &= (0, \delta, 0, \delta)^{\mathrm{T}} \\
\hat{M}^1 \cdot (0, \delta, 0, \delta)^{\mathrm{T}} &= (0, \delta, 0, \delta)^{\mathrm{T}} \\
\hat{M}^1 \cdot (\delta, 0, \delta, 0)^{\mathrm{T}} &= (\delta, 0, \delta, 0)^{\mathrm{T}}
\end{aligned}$$

如果 $\delta \in \{2, 8, 10\}$,

$$\begin{aligned}
\hat{M}^0 \cdot (0, \delta, 0, \delta)^{\mathrm{T}} &= (0, \delta, 0, \delta)^{\mathrm{T}} \\
\hat{M}^0 \cdot (\delta, 0, \delta, 0)^{\mathrm{T}} &= (\delta, 0, \delta, 0)^{\mathrm{T}} \\
\hat{M}^1 \cdot (0, \delta, 0, \delta)^{\mathrm{T}} &= (\delta, 0, \delta, 0)^{\mathrm{T}} \\
\hat{M}^1 \cdot (\delta, 0, \delta, 0)^{\mathrm{T}} &= (0, \delta, 0, \delta)^{\mathrm{T}}
\end{aligned}$$

4 7 轮 PRINCE 的差分区分器

由于轮密钥加和轮常数加不影响差分, 在下面的分析中我们将忽略这两个操作. 7 轮 PRINCE 可以表示成:

$$R^7 = R^{-1} \circ R^{-1} \circ S^{-1} \circ M' \circ S \circ R \circ R \circ R = S^{-1} \circ R'^{-1} \circ F_{\text{mid}} \circ R' \circ S \circ R$$

其中 $R=\mathrm{SR}\circ M'\circ S, R'=S\circ\mathrm{SR}\circ M', F_{\mathrm{mid}}=M'\circ\mathrm{SR}^{-1}\circ S^{-1}\circ M'\circ S\circ\mathrm{SR}\circ M'$. 为了计算 7 轮 PRINCE 在限定输入和输出差分集合的差分概率, 我们逐层考虑差分路径中各个节点的差分概率. 首先考虑 R 层的输入差分集合

$$\sum\nolimits_{\mathrm{in}}^{R}=\{\Delta^{i}_{(1,1,1,1)},\Delta^{i}_{(1,1,1,8)},\Delta^{i}_{(1,1,8,1)},\Delta^{i}_{(1,8,1,1)},\Delta^{i}_{(8,1,1,1)},\Delta^{i}_{(1,1,8,8)},\Delta^{i}_{(8,8,1,1)},\Delta^{i}_{(8,8,8,8)},i=5,6,7,8\}$$

和输出差分集合

$$\sum\nolimits_{\mathrm{out}}^{R}=\{\Delta^{j}_{(1,1,1,1)},\Delta^{j}_{(4,4,4,4)},\Delta^{j}_{(1,4,4,1)},\Delta^{j}_{(4,1,1,4)},\Delta^{j}_{(8,8,8,8)},j=1,2,3,4\}$$

对每个输入差分 $\Delta_{\mathrm{in}}\in\sum_{\mathrm{in}}^{R}$, 输出差分 $\Delta_{\mathrm{out}}\in\sum_{\mathrm{out}}^{R}$, 计算经过 R 层的差分概率 $P(\Delta_{\mathrm{in}}\xrightarrow{R}\Delta_{\mathrm{out}})$, 并将其写成一个 32×20 的矩阵 D_R, 其中第 $8k+1$ 至 $8k+8$ 行依次表示输入差分为 $\Delta^{k+5}_{(1,1,1,1)},\Delta^{k+5}_{(1,1,1,8)},\Delta^{k+5}_{(1,1,8,1)},\Delta^{k+5}_{(1,8,1,1)},\Delta^{k+5}_{(8,1,1,1)},\Delta^{k+5}_{(1,1,8,8)},\Delta^{k+5}_{(8,8,1,1)},\Delta^{k+5}_{(8,8,8,8)},k=0,1,2,3$; 第 $5t+1$ 至 $5t+5$ 列依次表示输出差分为 $\Delta^{t+1}_{(1,1,1,1)},\Delta^{t+1}_{(4,4,4,4)},\Delta^{t+1}_{(1,4,4,1)},\Delta^{t+1}_{(4,1,1,4)},\Delta^{t+1}_{(8,8,8,8)},t=0,1,2,3$. 概率 $P(\Delta_{\mathrm{in}}\xrightarrow{R}\Delta_{\mathrm{out}})$ 等于四个活动 S 盒差分概率的乘积, 例如,

$$P(\Delta^{5}_{(1,1,1,1)}\xrightarrow{R}\Delta^{1}_{(8,8,8,8)})=P(\Delta^{7}_{(1,1,1,1)}\xrightarrow{R}\Delta^{2}_{(8,8,8,8)})=p(1\to 8)^4=2^{-8}$$
$$P(\Delta^{5}_{(1,1,1,1)}\xrightarrow{R}\Delta^{4}_{(1,1,1,1)})=P(\Delta^{7}_{(1,1,1,1)}\xrightarrow{R}\Delta^{3}_{(1,1,1,1)})=p(1\to 1)^4=2^{-8}$$

它们经过 R 层的具体差分路径如图 4 所示.

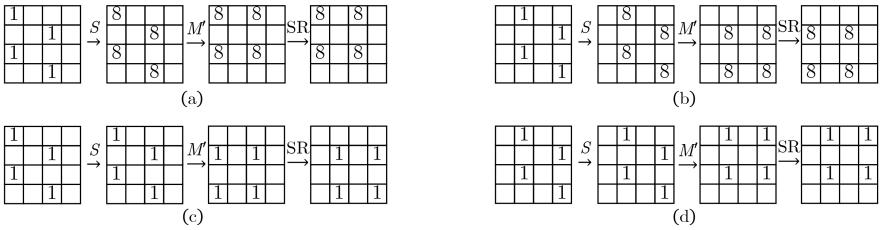

图 4 R 层的 4 条差分特征
Figure 4 4 differential characteristics of R

由于扩散层中矩阵 $\hat{M}^0,\hat{M}^1$ 具有的特殊性质, 矩阵 D_R 可以写成分块矩阵的形式:

$$D_R=\begin{pmatrix}0&0&0&A&B&0&0&0\\A&0&0&0&0&0&0&B\\0&0&A&0&0&B&0&0\\0&A&0&0&0&0&B&0\end{pmatrix}$$

其中

$$A=2^{-12}\times\begin{pmatrix}16&0&0&0&0&0&0&0\\1&2&2&2&2&4&4&16\\4&0&0&8&8&0&16&0\\4&8&8&0&0&16&0&0\end{pmatrix}^{\mathrm{T}}$$
$$B=2^{-12}\times\begin{pmatrix}16&8&8&8&8&4&4&1\end{pmatrix}^{\mathrm{T}}$$

将 R 层的输出差分集合作为 S 层的输入差分集, 即 $\sum_{\text{in}}^{S}=\sum_{\text{out}}^{R}$, 考虑 S 层的输出差分集合

$$\sum_{\text{out}}^{S}=\{\Delta_{(a',a',c',c')}^{j}|(a',c')\in\{1,4\}\times\{2,8,10\}\cup\{2,8,10\}\times\{1,4\},j=1,2,3,4\}$$

同样地计算每个输入差分 $\Delta_{\text{in}}\in\sum_{\text{in}}^{S}$, 输出差分 $\Delta_{\text{out}}\in\sum_{\text{out}}^{S}$, 经过 S 层的差分概率 $P(\Delta_{\text{in}}\xrightarrow{S}\Delta_{\text{out}})$, 将其写成 20×48 的矩阵 D_S. 由于 S 层分别独立地作用于 16 个块, 不会改变输入和输出的差分模式, 所以当 $i\neq j$ 时, $P(\Delta_{(a,b,c,d)}^{i}\xrightarrow{S}\Delta_{(a',a',c',c')}^{j})=0$ 且 $P(\Delta_{(a,b,c,d)}^{i}\xrightarrow{S}\Delta_{(a',a',c',c')}^{i})$ 等于四个活动 S 盒的差分概率的乘积, 即 $p(a\to a')\times p(b\to a')\times p(c\to c')\times p(d\to c')$. 矩阵 D_S 也可以写成分块矩阵的形式:

$$D_S=\begin{pmatrix}C&0&0&0&C&0&0&0\\0&C&0&0&0&C&0&0\\0&0&C&0&0&0&C&0\\0&0&0&C&0&0&0&C\end{pmatrix}$$

其中

$$C=2^{-12}\times\begin{pmatrix}0&0&16&4&0&0\\1&1&1&1&1&1\\0&0&4&2&0&0\\0&0&4&2&0&0\\0&4&0&4&0&0\end{pmatrix}$$

考虑 R' 层的输入差分集合和输出差分集合都为集合 $\sum_{\text{out}}^{S}$, 即

$$\sum_{\text{in}}^{R'}=\sum_{\text{out}}^{R'}=\sum_{\text{out}}^{S}=\{\Delta_{(a',a',c',c')}^{j}|(a',c')\in\{1,4\}\times\{2,8,10\}\cup\{2,8,10\}\times\{1,4\},j=1,2,3,4\}$$

计算每个输入差分 $\Delta_{\text{in}}\in\sum_{\text{in}}^{R'}$, 输出差分 $\Delta_{\text{out}}\in\sum_{\text{out}}^{R'}$, 经过 R' 层的差分概率 $P(\Delta_{\text{in}}\xrightarrow{R'}\Delta_{\text{out}})$, 将其写成一个 48×48 矩阵 $D_{R'}$. 同样矩阵 $D_{R'}$ 也可以写成分块矩阵的形式:

$$D_{R'}=\begin{pmatrix}G&0&0&0&G&0&0&0\\0&0&0&G&0&0&0&G\\0&0&0&G&0&0&0&G\\G&0&0&0&G&0&0&0\\0&G&0&0&0&G&0&0\\0&0&G&0&0&0&G&0\\0&0&G&0&0&0&G&0\\0&G&0&0&0&G&0&0\end{pmatrix}$$

其中

$$G=2^{-12}\times\begin{pmatrix}0&0&4&0&0&0\\0&0&1&0&0&0\\0&0&0&4&0&0\\0&2&0&2&0&0\\0&0&0&2&0&0\\0&0&0&1&0&0\end{pmatrix}$$

对 F_{mid} 层考察输入差分集合和输出差分集合都为集合 $\sum_{\text{out}}^{S}$, 即 $\sum_{\text{in}}^{F_{\text{mid}}}=\sum_{\text{out}}^{F_{\text{mid}}}=\sum_{\text{out}}^{S}$, 计算每个输入差分 $\Delta_{\text{in}}\in\sum_{\text{in}}^{F_{\text{mid}}}$, 输出差分 $\Delta_{\text{out}}\in\sum_{\text{out}}^{F_{\text{mid}}}$, 经过 F_{mid} 层的差分概率 $P(\Delta_{\text{in}}\xrightarrow{F_{\text{mid}}}\Delta_{\text{out}})$, 将其

写成 48×48 矩阵 $D_{F_{\mathrm{mid}}}$. 因为 $F_{\mathrm{mid}}=M'\circ \mathrm{SR}^{-1}\circ S^{-1}\circ M'\circ S\circ \mathrm{SR}\circ M'$ 不包含轮密钥加, 是一个与密钥无关的确定的函数, 可以通过计算机程序来计算我们需要的差分概率, 文献 [8] 也是如此计算中间层的差分概率的. 矩阵 $D_{F_{\mathrm{mid}}}$ 也可以写成分块矩阵的形式:

$$D_{F_{\mathrm{mid}}}=\begin{pmatrix} H & 0 & H & 0 & 0 & H & 0 & H\\ 0 & H & 0 & H & H & 0 & H & 0\\ 0 & H & 0 & H & H & 0 & H & 0\\ H & 0 & H & 0 & 0 & H & 0 & H\\ H & 0 & H & 0 & 0 & H & 0 & H\\ 0 & H & 0 & H & H & 0 & H & 0\\ 0 & H & 0 & H & H & 0 & H & 0\\ H & 0 & H & 0 & 0 & H & 0 & H \end{pmatrix}$$

其中

$$H=2^{-22}\times\begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 8 & 0 & 0 & 2\\ 0 & 0 & 0 & 6 & 2 & 0\\ 0 & 0 & 0 & 2 & 4 & 0\\ 0 & 0 & 2 & 0 & 0 & 3 \end{pmatrix}$$

考虑 R'^{-1} 层的输入差分集合和输出差分集合均为 $\sum_{\mathrm{out}}^{S}$, 考虑 S^{-1} 层的输入差分集合为 $\sum_{\mathrm{out}}^{S}$, 输出差分集合为 $\sum_{\mathrm{in}}^{S}=\sum_{\mathrm{out}}^{R}$, 则它们对应的差分概率矩阵 $D_{R'^{-1}}={D_{R'}}^{\mathrm{T}}, D_{S^{-1}}={D_S}^{\mathrm{T}}$. 利用上述每一层函数的差分概率矩阵, 我们计算输入差分 $\Delta_{\mathrm{in}}\in\sum_{\mathrm{in}}^{R}$, 输出差分 $\Delta_{\mathrm{out}}\in\sum_{\mathrm{out}}^{R}$, 经过 7 轮 PRINCE 后的差分概率 $P(\Delta_{\mathrm{in}}\xrightarrow{R^7}\Delta_{\mathrm{out}})$ 构成的 32×20 的差分概率矩阵

$$D_{R^7}=D_R\cdot D_S\cdot D_{R'}\cdot D_{F_{\mathrm{mid}}}\cdot {D_{R'}}^{\mathrm{T}}\cdot {D_S}^{\mathrm{T}}$$

将 D_{R^7} 写成分块矩阵的形式

$$D_{R^7}=\begin{pmatrix} N & N & N & N\\ N & N & N & N\\ N & N & N & N\\ N & N & N & N \end{pmatrix}$$

其中行表示输入差分 $\Delta^{i}_{(a,b,c,d)}\in\sum_{\mathrm{in}}^{R}, i=5,6,7,8$ 四种差分模式, 列表示 $\Delta^{j}_{(a',b',c',d')}\in\sum_{\mathrm{out}}^{R}, j=1,2,3,4$ 四种差分模式, 这 16 种输入和输出差分模式组合在限定的 8 个输入差分到 5 个输出差分的概率子矩阵 N 相同:

$$N=\begin{pmatrix} 2^{-59.74} & 2^{-62.68} & 2^{-61.58} & 2^{-61.58} & 2^{-62.82}\\ 2^{-62.32} & 2^{-65.13} & 2^{-64.15} & 2^{-64.15} & 2^{-65.25}\\ 2^{-62.32} & 2^{-65.13} & 2^{-64.15} & 2^{-64.15} & 2^{-65.25}\\ 2^{-62.32} & 2^{-65.13} & 2^{-64.15} & 2^{-64.15} & 2^{-65.25}\\ 2^{-62.32} & 2^{-65.13} & 2^{-64.15} & 2^{-64.15} & 2^{-65.25}\\ 2^{-61.64} & 2^{-64.54} & 2^{-63.48} & 2^{-63.48} & 2^{-64.67}\\ 2^{-61.64} & 2^{-64.54} & 2^{-63.48} & 2^{-63.48} & 2^{-64.67}\\ 2^{-62.99} & 2^{-65.49} & 2^{-64.83} & 2^{-64.83} & 2^{-65.67} \end{pmatrix}$$

矩阵 N 中有 12 个值大于 2^{-63}, 也就是说我们得到 7 轮 PRINCE 的 $12\times 16=192$ 对概率大于

2^{-63} 的差分, 由于计算时只使用了部分差分特征, 所以实际的差分概率要比 D_{R^7} 中给出的值更大.

下面我们给出区分 7 轮 PRINCE 和随机置换的差分区分器. 设明文输入差分为 $\Delta_{\rm in}=\Delta^5_{(1,1,1,1)}$, 任意固定的 $j\in\{1,2,3,4\}$, 对 7 轮 PRINCE 而言, 输出差分落入集合

$$\Omega^j_{\rm out}=\{\Delta^j_{(1,1,1,1)},\Delta^j_{(4,4,4,4)},\Delta^j_{(1,4,4,1)},\Delta^j_{(4,1,1,4)},\Delta^j_{(8,8,8,8)}\}$$

的概率为

$$\begin{aligned}P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Omega^j_{\rm out})=&P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Delta^j_{(1,1,1,1)})+P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Delta^j_{(4,4,4,4)})\\&+P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Delta^j_{(1,4,4,1)})+P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Delta^j_{(4,1,1,4)})\\&+P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Delta^j_{(8,8,8,8)})\\=&\,2^{-58.89}\end{aligned}$$

考察输出差分集合 $\Omega_{\rm out}=\cup_{j=1}^4\Omega^j_{\rm out}$, 任意一对输入差分为 $\Delta_{\rm in}=\Delta^5_{(1,1,1,1)}$ 的明文, 经过 7 轮 PRINCE 后输出差分属于集合 $\Omega_{\rm out}$ 的概率为

$$P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}\Omega_{\rm out})=2^{-58.89}\times 4=2^{-56.89}$$

而对随机置换而言, 其密文差分属于集合 $\Omega_{\rm out}$ 的概率为 $2^{-64}\times 5\times 4=2^{-59.68}$. 选择 $2^{56.89}$ 对差分为 $\Delta^5_{(1,1,1,1)}$ 的明文对, 计算其相应的密文差分, 对 7 轮 PRINCE 而言以概率为 $1-(1-2^{-56.89})^{2^{56.89}}\approx 63.21\%$, 存在一对密文差分属于集合 $\Omega_{\rm out}$; 而对随机置换而言, 存在一对密文差分属于集合 $\Omega_{\rm out}$ 的概率为 $1-(1-2^{-59.68})^{2^{56.89}}\approx 13.46\%$. 该区分器所需数据复杂度约为 $2^{57.89}$ 个选择明文, 计算复杂度约为 $2^{56.89}$ 次密文异或.

同样的方法也可以用来从选择密文的方向来区分 7 轮 PRINCE. 选择密文差分 $\Delta_{\rm out}=\Delta^1_{(1,1,1,1)}$, 明文差分集合

$$\Omega_{\rm in}=\{\Delta^i_{(1,1,1,1)},\Delta^i_{(1,1,1,8)},\Delta^i_{(1,1,8,1)},\Delta^i_{(8,1,1,1)},\Delta^i_{(1,1,8,8)},\Delta^i_{(8,8,1,1)},\Delta^i_{(8,8,8,8)},i=5,6,7,8\}$$

任意一对差分为 $\Delta^1_{(1,1,1,1)}$ 的密文, 对 7 轮 PRINCE 而言, 其明文差分属于集合 $\Omega_{\rm in}$ 的概率 $P(\Omega_{\rm in}\xleftarrow{R^7}\Delta^1_{(1,1,1,1)})=2^{-56.53}$; 而对随机置换而言, 其明文差分属于集合 $\Omega_{\rm in}$ 的概率为 2^{-59}.

5 8 轮 PRINCE 的密钥恢复攻击

利用 7 轮 PRINCE 差分区分器, 我们给出 8 轮 PRINCE 的密钥恢复攻击, 其数据复杂度为 $2^{61.89}$, 时间复杂度为 $2^{19.68}$, 存储复杂度为 $2^{15.21}$, 这是目前 8 轮 PRINCE 时间复杂度最低的密钥恢复攻击. 将上节的 7 轮 PRINCE 差分区分器稍做改变, 考虑明文差分为 $\Delta^5_{(1,1,1,1)}$, 输出差分集合为

$$W_{R^7}=\{\Delta^1_{(1,1,1,1)},\Delta^1_{(4,4,4,4)},\Delta^1_{(1,4,4,1)},\Delta^1_{(4,1,1,4)},\Delta^4_{(8,8,8,8)}\}$$

利用差分概率矩阵 D_{R^7}, 计算输入差分为 $\Delta^5_{(1,1,1,1)}$ 的明文, 经过 7 轮 PRINCE 后输出差分属于集合 W_{R^7} 的概率 $P(\Delta^5_{(1,1,1,1)}\xrightarrow{R^7}W_{R^7})=2^{-58.89}$, 而随机置换的概率为 $2^{-64}\times 5=2^{-61.68}$. 首先利用此差分区分器来恢复密钥 $\hat{k}=k_1\oplus k_0'$, 过程如图 5 所示.

记 $\hat{k}^{(l)}$ 表示 $\hat{k}$ 的第 l 块, $0\le l\le 15$, 块的顺序同图 2. 在明文差分为 $\Delta^5_{(1,1,1,1)}$ 的条件下, 选择密文差分属于模式 Δ^6 的密文对 (C,C'), 即 $C\oplus C'=(0\eta_0\eta_1\,0000\,\eta_20\eta_30\,0000),\eta_i\ne 0,i=0,1,2,3$. 猜测 $\hat{k}$

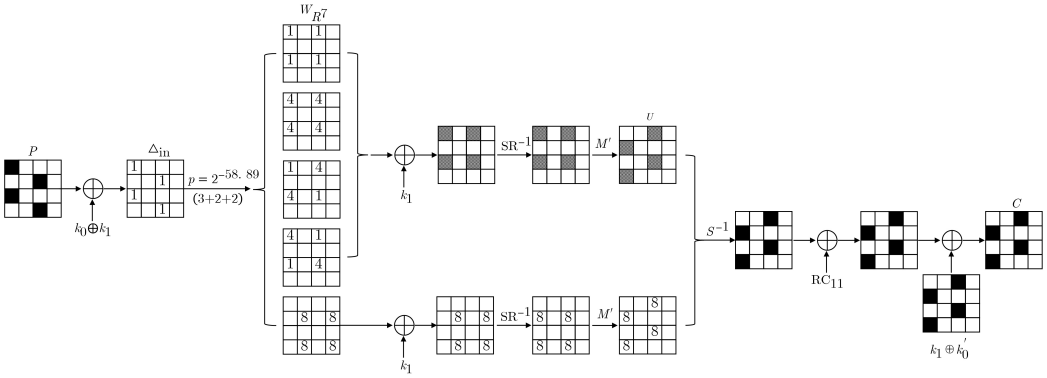

图 5 8 轮 PRINCE 密钥恢复攻击 ($\hat{k}$)
Figure 5 Key recovery attack on 8-round PRINCE($\hat{k}$)

的 4 个块 $\hat{k}^{(1)}$,$\hat{k}^{(3)}$,$\hat{k}^{(8)}$,$\hat{k}^{(10)}$ 并执行部分解密, 计算

$$
\begin{aligned}
x &= S(C^{(1)} \oplus \hat{k}^{(1)} \oplus \mathrm{RC}_{11}^{(1)}) \oplus S(C'^{(1)} \oplus \hat{k}^{(1)} \oplus \mathrm{RC}_{11}^{(1)}) \\
y &= S(C^{(3)} \oplus \hat{k}^{(3)} \oplus \mathrm{RC}_{11}^{(3)}) \oplus S(C'^{(3)} \oplus \hat{k}^{(3)} \oplus \mathrm{RC}_{11}^{(3)}) \\
z &= S(C^{(8)} \oplus \hat{k}^{(8)} \oplus \mathrm{RC}_{11}^{(8)}) \oplus S(C'^{(8)} \oplus \hat{k}^{(8)} \oplus \mathrm{RC}_{11}^{(8)}) \\
w &= S(C^{(10)} \oplus \hat{k}^{(10)} \oplus \mathrm{RC}_{11}^{(10)}) \oplus S(C'^{(10)} \oplus \hat{k}^{(10)} \oplus \mathrm{RC}_{11}^{(10)})
\end{aligned}
$$

如果 $(x, y, z, w) \in \{(1,1,1,1), (4,4,4,4), (1,1,4,4), (4,4,1,1), (8,8,8,8)\}$, 则 $\hat{k}^{(1)}$,$\hat{k}^{(3)}$,$\hat{k}^{(8)}$,$\hat{k}^{(10)}$ 对应猜测值的计数器增加 1. 对满足条件的密文对重复上述过程, 直到有一个计数器明显高于其他计数器, 则该计数器对应的值就是 $\hat{k}^{(1)}$,$\hat{k}^{(3)}$,$\hat{k}^{(8)}$,$\hat{k}^{(10)}$ 的候选值. 根据 Biham 和 Shamir [15] 关于差分分析数据量的分析, 我们攻击模型的信噪比为: $\frac{2^{16}\times 2^{-58.89}}{2^{-48}\times 5} = 6.91$, 所以选择 $2^{58.89} \times 4 = 2^{60.89}$ 对明文即可以高概率区分 $\hat{k}^{(1)}$,$\hat{k}^{(3)}$,$\hat{k}^{(8)}$,$\hat{k}^{(10)}$ 的正确值和错误值. 猜测 $\hat{k}$ 的 4 个块可以分别独立地进行, 再将可能落入集合的候选 x, y, z, w 联合在一起最终确定, 所以时间复杂度为 $2^{60.89} \times 2^{-48} \times 2 \times 2^4 \times 4 = 2^{19.89}$ 次 S 盒运算, 相当于 $2^{12.89}$ 次 8 轮 PRINCE 加密.

使用同样的方法可恢复 $\hat{k}$ 的其余 48 个比特, 只是考虑的输出差分集合不同而已, 恢复块 $\hat{k}^{(5)}$,$\hat{k}^{(7)}$,$\hat{k}^{(12)}$,$\hat{k}^{(14)}$ 时, 输出差分集合 $W_{R^7} = \{\Delta^2_{(1,1,1,1)}, \Delta^2_{(4,4,4,4)}, \Delta^2_{(1,4,4,1)}, \Delta^2_{(4,1,1,4)}, \Delta^3_{(8,8,8,8)}\}$; 恢复块 $\hat{k}^{(4)}$,$\hat{k}^{(6)}$,$\hat{k}^{(13)}$,$\hat{k}^{(15)}$ 时, 输出差分集合 $W_{R^7} = \{\Delta^3_{(1,1,1,1)}, \Delta^3_{(4,4,4,4)}, \Delta^3_{(1,4,4,1)}, \Delta^3_{(4,1,1,4)}, \Delta^2_{(8,8,8,8)}\}$; 恢复块 $\hat{k}^{(0)}$,$\hat{k}^{(2)}$,$\hat{k}^{(9)}$,$\hat{k}^{(11)}$ 时, 输出差分集合 $W_{R^7} = \{\Delta^4_{(1,1,1,1)}, \Delta^4_{(4,4,4,4)}, \Delta^4_{(1,4,4,1)}, \Delta^4_{(4,1,1,4)}, \Delta^1_{(8,8,8,8)}\}$. 它们构成的差分区分器具有相同的概率 $P(\Delta^5_{(1,1,1,1)} \xrightarrow{R^7} W_{R^7}) = 2^{-58.89}$. 由于明文差分相同, 故四次攻击可以共享相同的明密文数据, 所以恢复 $\hat{k}$ 的全部 64 比特需要约 $2^{61.89}$ 个选择明文, 时间复杂度为 $2^{14.89}$ 次 8 轮 PRINCE 加密, 存储复杂度为 $2^{60.89} \times 2^{-48} \times 5 = 2^{15.21}$ 个 16 比特计数器.

利用恢复出来的 $\hat{k}$, 将 8 轮 PRINCE 的密文解密一轮得到 7 轮的输出, 再利用 6.5 轮差分区分器可以恢复内层密钥 k_1. 首先 6.5 轮 PRINCE 可以表示成: $R^{6.5} = R'^{-1} \circ F_{\rm mid} \circ R' \circ S \circ R$, 利用上节的方法计算输入差分 $\Delta_{\rm in} \in \sum_{\rm in}^{R^{6.5}}$, 输出差分 $\Delta_{\rm out} \in \sum_{\rm out}^{R^{6.5}} = \sum_{\rm out}^{S}$, 经过 6.5 轮后的差分概率 $P(\Delta_{\rm in} \xrightarrow{R^{6.5}} \Delta_{\rm out})$ 构成的矩阵 $D_{R^{6.5}} = D_R \cdot D_S \cdot D_{R'} \cdot D_{F_{\rm mid}} \cdot D_{R'}{}^{\rm T}$. 将其写成分块矩阵的形式

$$
D_{R^{6.5}} = \begin{pmatrix}
Z & Z & Z & Z & Z & Z & Z & Z \\
Z & Z & Z & Z & Z & Z & Z & Z \\
Z & Z & Z & Z & Z & Z & Z & Z \\
Z & Z & Z & Z & Z & Z & Z & Z
\end{pmatrix}
$$

其中矩阵 Z 为一个 8×6 的矩阵:

$$Z=\begin{pmatrix}
2^{-56.89} & 2^{-58.89} & 2^{-52.91} & 2^{-53.87} & 2^{-53.91} & 2^{-54.91}\\
2^{-57.61} & 2^{-59.61} & 2^{-55.50} & 2^{-56.40} & 2^{-56.50} & 2^{-57.50}\\
2^{-57.61} & 2^{-59.61} & 2^{-55.50} & 2^{-56.40} & 2^{-56.50} & 2^{-57.50}\\
2^{-57.61} & 2^{-59.61} & 2^{-55.50} & 2^{-56.40} & 2^{-56.50} & 2^{-57.50}\\
2^{-57.61} & 2^{-59.61} & 2^{-55.50} & 2^{-56.40} & 2^{-56.50} & 2^{-57.50}\\
2^{-57.61} & 2^{-59.61} & 2^{-55.50} & 2^{-56.40} & 2^{-56.50} & 2^{-57.50}\\
2^{-57.83} & 2^{-59.83} & 2^{-54.82} & 2^{-55.75} & 2^{-55.82} & 2^{-56.82}\\
2^{-57.83} & 2^{-59.83} & 2^{-54.82} & 2^{-55.75} & 2^{-55.82} & 2^{-56.82}\\
2^{-56.61} & 2^{-58.61} & 2^{-56.17} & 2^{-57.11} & 2^{-57.17} & 2^{-58.17}
\end{pmatrix}$$

记 $k_1^{(l)}$ 表示 k_1 的第 l 块, $0\le l\le 15$, 块的顺序同图2. 我们以恢复 k_1 的四个块 $k_1^{(0)},k_1^{(2)},k_1^{(8)},k_1^{(10)}$ 为例来说明攻击的过程, 如图6所示.

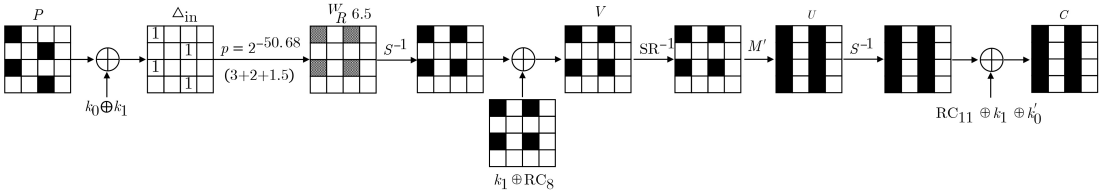

图 6　8 轮 PRINCE 密钥恢复攻击 (k_1)
Figure 6　Key recovery attack on 8-round PRINCE(k_1)

根据差分概率矩阵 $D_{R^{6.5}}$, 计算差分为 $\Delta^5_{(1,1,1,1)}$ 的明文经过 6.5 轮 PRINCE 后, 差分属于集合

$$W_{R^{6.5}}=\{\Delta^1_{(a',a',c',c')}|(a',c')\in\{1,4\}\times\{2,8,10\}\cup\{2,8,10\}\times\{1,4\}\}$$

的概率为 $P(\Delta^5_{(1,1,1,1)}\overset{R^{6.5}}{\to}W_{R^{6.5}})=2^{-50.68}$, 而随机置换的概率为 $2^{-64}\times 12=2^{-60.42}$. 利用第一阶段攻击 $\hat{k}$ 的数据, 选择密文差分满足第 2、4 列差分为 0 的密文对 (C,C'), 即 $C\oplus C'=(\eta_0\eta_1\eta_2\eta_3\,0000\,\eta_4\eta_5\eta_6\eta_7\,0000),\eta_i\neq 0,i=0,\cdots,7$. 使用 $\hat{k}$ 进行一轮解密得到 7 轮输出的状态 $V=\text{SR}\circ M'\circ S\circ\text{ARC}\circ\text{AK}(C)$, 并留下满足差分 $V\oplus V'=(\eta_0 0\eta_1 0\,0000\,\eta_2 0\eta_3 0\,0000),\eta_i\neq 0,i=0,1,2,3$ 的状态对 (V,V'). 猜测 $k_1^{(0)},k_1^{(2)},k_1^{(8)},k_1^{(10)}$, 对每一对留下的 (V,V'), 计算:

$$\begin{aligned}
x&=S(V^{(0)}\oplus k_1^{(0)}\oplus\text{RC}_{11}^{(0)})\oplus S(V'^{(0)}\oplus k_1^{(0)}\oplus\text{RC}_{11}^{(0)})\\
y&=S(V^{(2)}\oplus k_1^{(2)}\oplus\text{RC}_{11}^{(2)})\oplus S(V'^{(2)}\oplus k_1^{(2)}\oplus\text{RC}_{11}^{(2)})\\
z&=S(V^{(8)}\oplus k_1^{(8)}\oplus\text{RC}_{11}^{(8)})\oplus S(V'^{(8)}\oplus k_1^{(8)}\oplus\text{RC}_{11}^{(8)})\\
w&=S(V^{(10)}\oplus k_1^{(10)}\oplus\text{RC}_{11}^{(10)})\oplus S(V'^{(10)}\oplus k_1^{(10)}\oplus\text{RC}_{11}^{(10)})
\end{aligned}$$

如果 $(x,y,z,w)\in\{(a,a,c,c)|(a,c)\in\{1,4\}\times\{2,8,10\}\cup\{2,8,10\}\times\{1,4\}\}$, 则 $k_1^{(0)},k_1^{(2)},k_1^{(8)},k_1^{(10)}$ 对应猜测值的计数器增加 1, 直到有一个计数器的值明显高于其他计数器, 则该计数器对应的就是 $k_1^{(0)},k_1^{(2)},k_1^{(8)},k_1^{(10)}$ 的候选值. 同样依据文献 [15] 计算信噪比为 $\frac{2^{16}\times 2^{-50.68}}{2^{-48}\times 12}=852.19$, 所以成功区分 $k_1^{(0)},k_1^{(2)},k_1^{(8)},k_1^{(10)}$ 的正确值和错误值所需明文对为 $2^{50.68}\times 4=2^{52.68}$. 这些数据可以在第一阶段攻击 $\hat{k}$ 的 $2^{60.89}$ 对数据中随机选择, 它们对应的密文对满足第 2、4 列差分为 0 的对数约为 $2^{52.68}\times 2^{-32}=2^{20.68}$, 所以需要对 $2^{21.68}$ 个密文进行一轮解密. 解密后状态差分 $V\oplus V'=(\eta_0 0\eta_1 0\,0000\,\eta_2 0\eta_3 0\,0000),\eta_i\neq 0,i=0,1,2,3$ 的状态对 (V,V') 大约有 $2^{20.68}\times 2^{-16}=2^{4.68}$ 对, 判断 (x,y,z,w) 是否落在集合中约需要 $2^{4.68}\times 2\times 2^4\times 4=2^{11.68}$ 次 S 盒运算, 这部分计算量相对于密文一轮解密可以忽略不计, 所以恢复

$k_1^{(0)}, k_1^{(2)}, k_1^{(8)}, k_1^{(10)}$ 的时间复杂度为 $2^{21.68}$ 次一轮解密运算. 在恢复块 $k_1^{(1)}, k_1^{(3)}, k_1^{(9)}, k_1^{(11)}$ 时, 仍选择第 2、4 列差分为 0 的密文对进行一轮解密, 计算量可以与恢复 $k_1^{(0)}, k_1^{(2)}, k_1^{(8)}, k_1^{(10)}$ 时共享. 所以恢复 k_1 全部 64 比特密钥, 时间复杂度为 $2^{22.68}$ 次一轮解密, 相当于 $2^{19.68}$ 次 8 轮 PRINCE 加密. 在获得正确的 k_1 和 $\hat{k}$ 后, 利用它们之间的关系可以直接恢复出原始的 128 比特密钥. 综上所述, 恢复 8 轮 PRINCE 的全部密钥需要的数据复杂度为 $2^{61.89}$, 时间复杂度为 $2^{19.68}$, 存储复杂度为 $2^{15.21}$, $D \times T$ 复杂度为 $2^{81.57}$.

作为比较, 将文献 [8] 中 A. Canteaut 等给出的 10 轮攻击方案减少到 8 轮, 恢复 128 比特密钥中的 32 比特的过程如图7所示. 由于 A. Canteaut 等给出的区分器最长为 6 轮, 概率为 $2^{-53.36}$, 所以攻击 8 轮 PRINCE 需要首尾同时猜测密钥且最少需要猜测 8 个密钥块. 根据 6 轮区分器的概率, 数据复杂度至少需要 $2^{53.36} \times 4 \times 2 = 2^{56.36}$ 个选择明文, 时间复杂度至少 $2^{32} \times (2^{56.36} \times 2^{-48}) \times 8 = 2^{43.36}$ 次 S 盒运算, 相当于 $2^{36.36}$ 次 8 轮 PRINCE 加密. 恢复全部 128 比特密钥的数据复杂度至少为 $2^{58.36}$, 时间复杂度至少为 $2^{38.36}$. 相较之下, 本文在时间复杂度上具有明显优势.

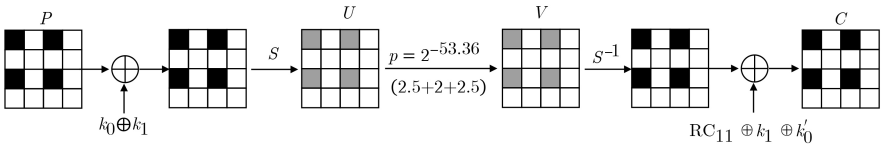

图 7 应用文献 [8] 区分器攻击 8 轮 PRINCE

Figure 7 Attack on 8-round PRINCE using distinguisher given in Ref. [8]

6 结束语

本文利用改进的多重差分技术给出了目前轮数最长的 7 轮 PRINCE 区分器, 将其与随机置换区分需要数据复杂度约为 $2^{57.89}$ 个选择明文, 计算复杂度约为 $2^{56.89}$ 次密文异或. 利用此区分器我们给出了 8 轮 PRINCE 的密钥恢复攻击, 数据复杂度为 $2^{61.89}$ 个选择明文, 时间复杂度为 $2^{19.68}$ 次 8 轮加密, 存储复杂度为 $2^{15.21}$ 个 16 比特计数器. 相比目前已知的 8 轮 PRINCE 密钥恢复攻击的结果, 包括将 A. Canteaut 等给出的 10 轮攻击方案减少到 8 轮, 本文的时间复杂度和 $D \times T$ 复杂度都是最低的. 能否将本文的攻击技术扩展到更高轮数将是我们后续研究的方向.

参考文献

[1] BOGDANOV A, KNUDSEN L R, LEANDER G, et al. PRESENT: An ultra-lightweight block cipher[C]. In: Cryptographic Hardware and Embedded Systems—CHES 2007. Springer Berlin Heidelberg, 2007: 450–466. [DOI: 10.1007/978-3-540-74735-2_31]

[2] BEAULIEU R, SHORS D, SMITH J, et al. The Simon and Speck families of lightweight block ciphers[C]. In: Proceedings of the 52nd Annual Design Automation Conference. ACM, 2015: 175. [DOI: 10.1145/2744769.2747946]

[3] BORGHOFF J, CANTEAUT A, GÜNEYSU T, et al. PRINCE—A low-latency block cipher for pervasive computing applications[C]. In: Advances in Cryptology—ASIACRYPT 2012. Springer Berlin Heidelberg, 2012: 208–225. [DOI: 10.1007/978-3-642-34961-4_14]

[4] KILIAN J, ROGAWAY P. How to protect DES against exhaustive key search (an analysis of DESX)[J]. Journal of Cryptology, 2001, 14(1): 17–35. [DOI: 10.1007/s001450010015]

[5] SOLEIMANY H, BLONDEAU C, YU X, et al. Reflection cryptanalysis of PRINCE-like ciphers[C]. In: Fast Software Encryption—FSE 2013. Springer Berlin Heidelberg, 2013: 71–91. [DOI: 10.1007/978-3-662-43933-3_5]

[6] JEAN J, NIKOLIC I, PEYRIN T, et al. Security analysis of PRINCE[C]. In: Fast Software Encryption—FSE 2013. Springer Berlin Heidelberg, 2013: 92–111. [DOI: 10.1007/978-3-662-43933-3_6]

[7] LI L B, JIA K T, WANG X Y. Improved meet-in-the-middle attacks on AES-192 and PRINCE[J]. IACR Cryptology ePrint Archive, 2013: 2013/573. http://eprint.iacr.org/2013/573.pdf

[8] CANTEAUT A, FUHR T, GILBERT H, et al. Multiple differential cryptanalysis of round-reduced PRINCE[C]. In: Fast Software Encryption—FSE 2014. Springer Berlin Heidelberg, 2014: 591–610. [DOI: 10.1007/978-3-662-46706-0_30]

[9] ZHAO G Y, SUN B, LI C, et al. Truncated differential cryptanalysis of PRINCE[J]. Security and Communication Networks, 2015, 8: 2875–2887. [DOI: 10.1002/sec.1213]
[10] DERBEZ P, PERRIN L. Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE[C]. In: Fast Software Encryption—FSE 2015. Springer Berlin Heidelberg, 2015: 190–216. [DOI: 10.1007/978-3-662-48116-5_10]
[11] MORAWIECKI P. Practical attacks on the round-reduced PRINCE[J]. IACR Cryptology ePrint Archive, 2015: 2015/245. http://eprint.iacr.org/2015/245.pdf
[12] POSTEUCA R, NEGARA G. Integral cryptanalysis of round-reduced PRINCE cipher[J]. Proceedings of the Romanian Academy Series A—Mathematics, Physics, Technical Sciences, Information Science, 2015, 16: 265–269.
[13] RASOOLZADEH S, RADDUM H. Faster key recovery attack on round-reduced PRINCE[C]. In: Lightweight Cryptography for Security and Privacy—LightSec 2016. Springer Cham, 2017: 3–17. [DOI: 10.1007/978-3-319-55714-4_1]
[14] GRASSI L, RECHBERGER C. Practical low data-complexity subspace-trail cryptanalysis of round-reduced PRINCE[C]. In: Progress in Cryptology—INDOCRYPT 2016. Springer Cham, 2016: 322–342. [DOI: 10.1007/978-3-319-49890-4_18]
[15] BIHAM E, SHAMIR A. Differential cryptanalysis of DES-like cryptosystems[J]. Journal of Cryptology, 1991, 2(3): 3–72. [DOI: 10.1007/BF00630563]

作者信息

段春晖 (1995–), 黑龙江哈尔滨人, 硕士. 主要研究领域为对称密码的设计与分析.
duan_chunhui@126.com

谭林 (1983–), 湖北天门人, 博士, 副教授. 主要研究领域为对称密码的设计与分析.
tanlin100@163.com

戚文峰 (1963–), 浙江宁波人, 博士, 教授. 主要研究领域为对称密码算法的安全性分析.
qiwenfeng@263.com

附录: PRINCE 算法 S 盒的差分分布表

输入差分	输出差分															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	0	2	0	2	0	4	2	0	2	0	0	0	0
2	0	2	0	4	0	0	0	2	2	0	0	0	0	4	2	0
3	0	0	0	0	0	2	2	0	2	2	2	2	2	0	0	2
4	0	2	2	4	2	2	0	0	2	0	2	0	0	0	0	0
5	0	0	2	2	0	2	0	2	0	2	0	2	2	2	0	0
6	0	0	2	2	0	2	2	0	0	2	0	2	0	0	4	0
7	0	0	2	0	0	0	2	0	2	0	4	0	0	2	2	2
8	0	0	2	0	4	2	0	0	2	2	0	2	0	2	0	0
9	0	0	2	2	0	0	0	0	0	2	2	0	4	2	0	2
A	0	0	0	2	2	4	0	4	2	0	0	0	0	0	0	2
B	0	2	0	0	4	0	0	2	0	0	0	2	2	0	2	2
C	0	4	0	0	0	2	2	0	0	0	2	2	2	0	2	0
D	0	2	0	0	0	0	0	2	0	4	2	0	0	2	2	2
E	0	0	2	0	0	0	4	2	0	0	0	2	2	2	0	2
F	0	0	2	0	2	0	2	2	0	0	2	0	2	0	2	2